Action Nominals in the Grammar Matrix

Keren Ruditsky

A thesis
submitted in partial fulfillment of the
requirements for the degree of

Master of Science

University of Washington

2024

Committee:
Emily M. Bender
Shane Steinert-Threlkeld

Program Authorized to Offer Degree:
Department of Linguistics

University of Washington

Abstract

Action Nominals in the Grammar Matrix

Keren Ruditsky

Chair of the Supervisory Committee:
Emily M. Bender
Department of Linguistics

This thesis describes the addition of a library for action nominal constructions (ANCs) to the LinGO Grammar Matrix customization system. Actions nominals are nominalized verbs which refer to an action or process and are often used cross-linguistically to mark clausal complements and adverbial clauses. They occupy an intermediate state between nouns and verbs, having the external distribution of a noun phrase, but often still retaining certain verbal properties. In this thesis, I build on the existing analysis for nominalized clauses in the Grammar Matrix, but shift away from an approach where the dual nominal and verbal characteristics of action nominals are explained based on what level in the tree nominalization occurs to one that relies primarily on lexical rules. This change is motivated by a desire to expand the typological range of nominalization patterns the Matrix can handle while also more closely reflecting the hybrid syntactic nature of action nominals. I present an HPSG analysis of action nominals and the implementation of that analysis within the Grammar Matrix. I develop the library using a combination of pseudo and illustrative languages (English [eng], Hixkaryana [hix], Russian [rus], Korean [kor]) and then test on small testsuites from five held-out languages (Wayana [way], Maltese [mit], Dutch [nld], Lango [laj], Finnish [fin]. The system achieved on average 95.2% coverage and 7.0% over-generation on the held-out data.

TABLE OF CONTENTS

ACKNOWLEDGMENTS

Thank you to the DELPH-IN members who provided feedback on the analysis and helped me spot problems as well as areas for improvement. Special thank you to Dan Flickinger, Ann Copestake, Guy Emerson, and Berthold Crysmann for both asking and answering questions as well as pointing me towards useful resources. Another large thank you to my reader Shane Steinert-Threlkeld for providing helpful and insightful comments which allowed me to see things from a new perspective. I am also incredibly grateful to my advisor Emily M. Bender who has helped me throughout the entire process in ways both big and small. She guided me in shaping the analysis and pulled me out of many a rabbit hole. I am thankful for her patience and detailed comments and questions long before my work and writing took on a coherent shape.

Chapter 1
INTRODUCTION

The aim of this thesis is to expand the existing nominalized clauses library (Howell et al., 2018) in the Grammar Matrix customization system to more comprehensively account for action nominal constructions. The Grammar Matrix is a starter-kit designed to aid in the rapid development of implemented HPSGs (Head-Driven Phrase Structure Grammars) (Bender et al., 2002). It acts as a common starting ground that grammar developers can enrich with language-specific information. The customization system is a web user-interface that expedites this second step by allowing a developer to provide a grammar specification, including typological and lexical information through the use of a questionnaire (Bender et al., 2010). The answers to the questionnaire are subsequently used to generate language-specific analyses written in a machine-readable format (TDL or type description language (Copestake, 2002)) for a variety of phenomena. The python code which outputs these analyses and the accompanying questionnaire questions are referred to as libraries (Bender et al., 2010). Thus, the existing nominalized clauses library, for example, allows users to input information about nominalized clauses in a given language and then outputs files containing an HPSG analysis written in TDL. In this thesis, I describe the design and implementation of a library for action nominal constructions including creating a cross-linguistic HPSG analysis, adding the appropriate questions to the online questionnaire, and writing the TDL and python code which will output the analysis.

The definition that I use for action nominals is based on the one first given by Comrie (1976) and also adopted by Koptjevskaja-Tamm (1993) when designing a typology of action nominal constructions. For the purposes of the library, an action nominal is defined as a nominalized verb with the meaning of an action or a process which can either take nominal

morphology or take as arguments the same adpositions as non-derived nouns (Comrie, 1976, p. 178). An action nominal construction (ANC) is simply a phrase headed by an action nominal (Koptjevskaja-Tamm, 1993).

Stepping back from action nominals specifically, the term nominalization refers to the process of turning a word (generally a verb or an adjective) into a noun. A variety of different nominalizations exist in the world's languages with many falling into the category of argument nominalizations. These are nominalizations where, in contrast to action nominals, the derived noun refers to one of the arguments of the original lexical item (Comrie and Thompson, 1985). One example includes agent nouns such as *singer* (the one singing) in English [eng].

Alongside the difference in meaning, action nominals derived from verbs are further distinguished from argument nominalizations of verbs in that despite both having the external distribution of a noun, only action nominals retain certain verbal properties (Comrie and Thompson, 1985). This can include taking verbal morphology or marking the arguments of the derived noun in a typically verbal manner. Action nominals also retain the ability to mark all the core participants of the underlying verb. For intransitive verbs this is the agent (the one doing the action) and for transitive verbs it is both the agent and the patient (the one the action is done to). An example (1)[1] is shown below from Meadow Mari [mme].

(1) *möj-ön* *pis'ma-m* *voz-öm-em*
I-GEN letter-ACC write-NMLZ-1SG.POSS
'My writing of the letter.' [mme] (Koptjevskaja-Tamm, 2013)

The nominalized verb in (1) *voz-öm-em* 'write' has two arguments which correspond to the subject *möj* 'I' and the object *pis'ma* 'letter' of the verb. Notably, the object is

[1]Abbreviations used in this thesis: ABS = absolutive, ACC = accusative, CG = common gender, COLL = collective, COMP = complementizer, COMPL = completive, DAT = dative, DVL = devaluative, DENOMLZR = denominalizer, DIST = distant, ERG = ergative, FEM = feminine, GEM = genitive, HAB = habitual, HON = honorific, IMM = immediate, INDIV = individual, INSTR = instrumental, MASC = masculine, NEUT = neuter, NMLZ = nominalizer, NOM = nominative, NPST = nonpast, OBLAGT = oblique agent, PFV = perfective, PRS = present, PL = plural, POSS = possessive, PST = past, Q = question-marker, REC = recent, SBJ = subject, SE = sentence-ending, SG = singular. 1 = first person, 3 = 3rd person.

marked with accusative case just as the object of a verb would be in a typical finite clause (Koptjevskaja-Tamm, 2013).

To further define the library's scope, only actions nominals which can express all the core arguments of the original verb within an ANC will be considered. This does not mean that all these arguments must always be overt, but there must be a way to express the argument if desired. For example, certain action nominals in Hixkaryana [hix] can only express the patient-like argument of the verb (Derbyshire, 1985). The agent-like argument is null and interpreted as the same as the subject of a superordinate clause. Nominals with this restriction are created through the addition of the suffix *-txahke* which indicates 'desire involving effort'.

(2) *r-otaha-txahke*
1.POSS-hit-NMLZ.DESIRE.INVOLVING.EFFORT
'Trying to hit me.' [hix] (Derbyshire, 1985: 46)

(3) **ti-wya* *r-otaha-txahke*
3.REFLEXIVE-by 1.POSS-hit-NMLZ.DESIRE.INVOLVING.EFFORT
'Him trying to hit me.' [hix] (Derbyshire, 1985: 46)

As seen by comparing (2) and (3), the action nominal *r-otaha-txahke* 'hit' is only grammatical as long as the agent-like argument *ti-wya* '3.reflexive-by' is left out of the ANC. However, the patient-like argument can still be expressed through the prefix *r-* '1POSS' on the action nominal. Since there is no possibility to mark the agent within the ANC itself, these action nominals are outside of the library's coverage.

Also out of scope are nominalized copulas. Copulas are treated as distinct from other verbs in the Grammar Matrix and have numerous additional complexities. Thus, while action nominals derived from copulas are possible (*my being happy*), the library as described in this thesis will not account for them.

Another important note is that while my work will focus solely on action nominals, nominalizations are often ambiguous between several different readings. For example, an English

[eng] nominalization such as *agreement* can have both an action reading (*his agreement to the contract surprised me*), and a more concrete interpretation (*The agreement was signed last night*) (Koptjevskaja-Tamm, 1993). Argument nominalizations are not yet implemented in the Grammar Matrix, but future work on them should ensure that their semantic representation is distinct from that of action nominals in order to capture this ambiguity.

As to the value of modeling action nominals, cross-linguistically, action nominals often serve as a way to mark clausal complements and adverbial modifiers (Koptjevskaja-Tamm, 2006). The existing nominalized clauses library was designed to allow nominalization to serve as one strategy to signal such clauses (Howell et al., 2018). Improvements to the library will then serve to broaden the range of languages the customization system can account for. Additionally, since action nominals have aspects of nominal and verbal syntax they serve as a good test of how different analyses in the Grammar Matrix are interacting with one another.

As is, the nominalized clauses library can handle any nominalized clause which is either identical to finite clauses in the language or where any deviations from finite clauses are marked by case changes. Thus, the current library is able to account for the example in (1) where the only difference between this clause and a regular finite clause is that the agent-like argument takes genitive as opposed to nominative case (Koptjevskaja-Tamm, 2013). However, the library has no clear way to mark nominalized clauses in languages that do not use case, but in which nominalized clauses still show significant differences from finite clauses.

An example from Hixkaryana [hix], a language without case, but which makes extensive use of nominalization to signal subordinate clauses, is shown in (4) (Derbyshire, 1985). Finite clauses in Hixkaryana [hix] (5) have OVS word order with prefixes on the verb marking the person of the subject and/or object and suffixes marking elements such as tense and aspect. Nominalized clauses have both a distinct word order as well as a different set of affixes on the nominalized verb.

(4) *txetxa* *wawo* *wewe* *y-omokoto-ni-ri* *w-enye-no*
forrest in tree 3.OVERT-fall-NMLZ-POSS 1-see-IMM.PST.INDIV
'I saw a tree falling in the forest.' [hix] (Derbyshire, 1985: 40)

(5) *toto* *y-ahoshi-ye* *kamara*
man 3>3-get-DIST.PST.COMPL jaguar
'The jaguar grabbed the man.' [hix] (Derbyshire, 1985: 32)

Specifically, the nominalized verb *omokoto-ni* 'fall' in (4) takes a third person prefix which agrees in person with the immediately preceding noun (*wewe* 'tree') and a possessive suffix. *wewe* 'tree' in this clause serves as the semantic agent. Unlike in finite clauses where the agent-like argument follows the verb, *wewe* 'tree' appears before the nominalized verb. As shown in (6), this is the same word order and affixes used in possessive noun phrases. The prefix on the non-derived noun (*owa* 'chest') agrees in person with the semantic possessor (*waraka* 'waraka') which precedes the possessed noun.

(6) *waraka* *y-owa-ni*
waraka 3.OVERT-chest-POSS
'Waraka's chest.' [hix] (Derbyshire, 1985: 200)

The overall syntax of nominalized clauses is thus more similar to noun phrases in Hixkaryana [hix] than to verb phrases with one of the arguments of the nominalized verb treated in the same way as a syntactic possessor. The current nominalized clauses library cannot easily model this fact since the most common syntactic analysis of possessors in the Grammar Matrix is as specifiers (Nielsen, 2018). Nominalized verbs, however, in line with other verbs, are treated as having subjects and subsequently blocked from using any part of the existing specifier possessor analysis. Due to this, it would likely be necessary to create a duplicate set of prefixes and suffixes that are specific to action nominals. This approach is repetitive and does not take advantage of the already existing analyses that are present in the adnominal possession library. While the semantics between (4) and (6) may differ, it would be ideal to develop an analysis that reflects the common features of their syntax.

The problem that this thesis attempts to solve is to develop an analysis of action nominals that accounts for nominalized clauses in languages without case and that more strongly models the interaction between nominal and verbal syntax in nominalized clauses. Chapter 2 provides background concerning the syntactic and semantic formalisms used by the Grammar Matrix. This chapter also goes into more detail about the Grammar Matrix and customization system itself including libraries such as the adnominal possession library which are particularly important for action nominals. Chapter 3 goes over the typological literature concerning ANCs as well as previously proposed syntactic analyses. This provides set-up for the range of phenomena the library covers as well as motivation for the final selected syntactic analysis. Chapter 4 describes the analysis of action nominal constructions used by the library and chapter 5 deals with its implementation. The latter involves expanding the questionnaire as well as converting the analysis from chapter 4 into TDL and writing the python code to translate answers to questionnaire questions into particular analyses. Chapter 6 discusses the evaluation of the library which involves looking at its coverage and over-generation on a set of pseudo, illustrative and held-out languages.

Chapter 2
BACKGROUND

This chapter provides necessary background for the syntactic and semantic formalisms used by the Grammar Matrix and customization system. The grammars produced by the Grammar Matrix are HPSG (Head-Driven Phrase Structure Grammar) grammars with Minimal Recursion Semantics (MRS) used as a semantic representation. Section 2.1 provides an overview of HPSG (Pollard and Sag, 1994) and section 2.2 of MRS (Copestake et al., 2005). Section 2.3 follows up with a more in-depth discussion of the Grammar Matrix and customization system itself, touching upon the implementation of several syntactic and semantic phenomena which have particular importance for the analysis of action nominal constructions to be discussed in later chapters.

2.1 Head-Driven Phrase Structure Grammar

HPSG makes use of a sign-based architecture where all lexical items, lexical rules, and phrase structure rules are analyzed as signs. A sign is a pairing of meaning (semantics) and form (syntax, phonology, morphology) which is represented in HPSG through typed feature structures. A feature structure is a set of feature-value pairs with the specific *type* of a feature structure determining which features are appropriate for it. Features themselves can only take values of particular types. The feature-values pairs specified for a type then serve as constraints as to what properties a given type can exhibit, with the full set of types in a language serving as a prediction of what linguistic entities exit in that language. The value of a feature may be an atomic value or another feature structure.

Another core idea in HPSG is that all types are organized in a type hierarchy with subtypes inheriting all the constraints of their supertypes. This makes it easy to both

express generalizations across many types in a language (for example setting a constraint which applies to all verbs), while also making it possible to create constraints specific to one or a small number of lexical items.

In terms of how HPSG determines what counts as grammatical in a particular language, grammatical phrases are modeled as trees with a variety of lexical rules, phrase structure rules, and various general principles serving as constraints on the set of valid trees in a language (Pollard et al., 2003). As with all signs, phrase structure and lexical rules are represented as feature structures with rules containing features corresponding to both the mother and daughter(s) of the rule. Lexical rules are differentiated from phrase structure rules in being restricted to the bottom of the tree. All lexical rules must then occur prior to the phrase structure rules. More will be said about how this is specifically implemented within the Grammar Matrix in section 2.3.

As suggested by the name, headedness plays a large role in HPSG with the head daughter in all headed phrase structure rules sharing its categorical information (stored under the HEAD feature) with the overall phrase. This is significant as categorical information is used to determine the overall distribution of a lexical item. One particularly significant HEAD value is the part of speech of the lexical item. For example, *noun* or *verb*. The VAL(valence) features of the head daughter also dictate which arguments the head selects for (selectional information), based on the constraints specified for a specific headed rule. In this way, the head daughter of a headed rule determines both the distribution and the selectional behavior of the entire phrase.

As an example, a feature structure and a headed phrase structure rule are shown below as AVMs (attribute-value matrices) in (1) and (2) respectively. (1) and (2) only show features that are relevant for the explanation that follows. Additionally, all feature names and paths used are specific to the Grammar Matrix.

(1)
$$\begin{bmatrix} \textit{word} \\ \text{SYNSEM} & \begin{bmatrix} \text{LOCAL} & \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{verb} \\ \text{FORM} \quad \text{fin} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SUBJ} \quad \langle \boxed{1} \rangle \\ \text{SPEC} \quad \langle\rangle \\ \text{SPR} \quad \langle\rangle \\ \text{COMPS} \quad \langle \boxed{2} \rangle \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix} \\ \text{ARG-ST} & \langle \boxed{1}, \boxed{2} \rangle \end{bmatrix}$$

(2)
$$\begin{bmatrix} \textit{basic-head-subj-phrase} \\ \text{SYNSEM} & \begin{bmatrix} \text{LOCAL} \begin{bmatrix} \text{CAT} \begin{bmatrix} \text{VAL} \begin{bmatrix} \text{SUBJ} \quad \langle\rangle \end{bmatrix} \end{bmatrix} \end{bmatrix} \\ \text{HEAD-DTR} & \begin{bmatrix} \text{SYNSEM} \begin{bmatrix} \text{LOCAL} \begin{bmatrix} \text{CAT} \begin{bmatrix} \text{VAL} \begin{bmatrix} \text{SUBJ} \quad \langle \boxed{1} \rangle \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix} \\ \text{NON-HEAD-DTR} & \begin{bmatrix} \text{SYNSEM} \quad \boxed{1} \end{bmatrix} \end{bmatrix}$$

The AVM in (1) is of type *word* and contains the feature SYNSEM. This feature contains both syntactic (SYN) and semantic (SEM) information. The CAT feature is specific to syntactic information, with a different feature CONT (not shown in (1)) responsible for all semantic information. The HEAD and VAL features within CAT specify categorical and selectional information respectively. As mentioned above, the HEAD feature is of particular importance as a separate constraint requires the HEAD values of the mother and head daughter to be identical in all headed rules. In addition to the HEAD feature providing information about part of speech, it also takes the FORM feature. This is a feature that is used to differentiate between different types of verbal distributions such as verbs that can appear in finite versus non-finite positions. As will be discussed later on, at least within the Grammar Matrix, the FORM feature has also been extended to complemantizers allowing different complementizers to be selected based on their orthographic form (Zamaraeva et al., 2019). For example, the English complementizer *that* could have the FORM value *that*. A verb could then select for a complementizer phrase with [FORM *that*] to ensure that no other

complementizers would be selected by that verb.

Moving on to the VAL feature, it demonstrates that there are three kinds of arguments that a word can take: SUBJ (subject), SPR (specifier), and COMPS (complement). SUBJ and SPR are distinguished by the fact that SUBJ is the external argument of a verb, while SPR is the external argument of a noun. SPEC is essentially the inverse operation to SPR, allowing a specifier to have control over the properties of the word it appears with. (1) then provides an example of a *verb* that takes a subject and a single complement. Boxed numbers are used to indicate structure sharing. Thus, the verb's subject is identical to the first element on its ARG-ST list (list of arguments) and the complement to its second element.

The rule in (2) is an example of a simplified head-subj(ect) rule.[1] This rule constrains the HEAD-DTR (head daughter) of a head-subj phrase to have a SUBJ value equal to the SYNSEM of the NON-HEAD-DTR (non-head daughter), and for the overall phrase to have an empty SUBJ value. The rule then demonstrates how lexical items can discharge SUBJ values. The rule is underspecified in the sense that it does not specify any value for features such as the HEAD feature. This allows the head-subj phrase rule to combine (unifiy) with types containing any HEAD value. For example, if the type in (1) were to unify with the HEAD-DTR value in (2), the resulting category would be a verb phrase, since the HEAD value of the HEAD-DTR would be shared with the mother. However, the general constraint dictating how a lexical item should combine with a subject holds for lexical items of all HEAD values. In this way phrase structure rules (as well as lexical rules and other similar mechanisms) serve to constrain the set of phrases and thus trees that are valid in a given language.

As discussed in Malouf (1998), the fact that all linguistic entities in HPSG are defined based on what features they can take, makes the formalism ideal for describing mixed categories such as action nominals. A mixed category is just a type with a combination of

[1]Unless the order is explicitly given whenever the name of a specific headed rule (head-subject (head-subj), head-specifier (head-spec), head-complement (head-comp)) is mentioned it is assumed to refer to both head-initial and head-final versions of the rule. Thus, a mention of the head-subj rule removed from the context of a specific language refers to both the head-subj rule and the subj-head rule.

features that appear in other lexical types. For example, in a given language, a type could inherit verbal VAL features but nominal HEAD features. An adequate cross-linguistic account of action nominals then requires ensuring that for a given language only valid mixed category types (valid combinations of nominal and verbal features) can occur, and all other combinations are deemed ungrammatical.

2.2 Minimal Recursion Semantics

Minimal Recursion Semantics (MRS) is a computational semantics framework that can be implemented using typed feature structure formalisms such as HPSG (Copestake et al., 2005). The goal of MRS is to create a flat semantic representation that still allows for the underspecification of scope. Each underspecified MRS then points to a set of fully resolved semantic representations with constraints put in place to ensure only valid representations are possible. The basic unit in an MRS is an EP (elementary predication) which consists of a handle, a relation and lists of both ordinary and scopal arguments. The majority of lexemes in a language correspond to exactly one EP. Example EPs for the lexemes *dog* and *sleep* are shown below in (3) and (4) respectively:

(3) $\begin{bmatrix} \text{LBL} & \textit{handle} \\ \text{PRED} & \textbf{\_dog\_n\_rel} \\ \text{ARG0} & \textit{ref-ind} \end{bmatrix}$

(4) $\begin{bmatrix} \text{LBL} & \textit{handle} \\ \text{PRED} & \textbf{\_sleep\_v\_rel} \\ \text{ARG0} & \textit{event} \\ \text{ARG1} & \textit{ref-ind} \end{bmatrix}$

In the Grammar Matrix, the PRED feature is used to store the EP's relation which is just a string containing the orthographic form of the lexeme as well as part of speech tags (n (noun), v (verb) ...) used as coarse-grained sense labels. Argument positions are stored in general ARG features. ARG0 serves as an index for the lexeme itself, while all other ARG values represent positions that can be filled by other elements during semantic composition.

Thus, the ARG1 feature in (4) represents the one doing the *sleeping* action. As can be seen by comparing (3) and (4), different lexical items take different ARG0 types with ordinary nouns having indices of type *ref-ind* and ordinary verbs of type *event*. These different types have consequences in terms of how they interact with quantifiers as will be discussed below.

Returning to the example EPs, the LBL feature is used to store the EP's handle. The fact that each EP has a handle allows for a fully flat representation as if an argument position needs to be identified with an EP, it can be identified with the EP's handle and not the EP itself. There is thus no need for EP embedding within MRSs. The idea behind MRS is that as words combine in the syntax their EPs also combine in the semantics with argument positions in particular EPs being linked to elements that can fill them. For example, in a sentence such as *dogs sleep*, the ARG1 of the EP in (4) would be conindexed with the ARG0 of the EP in (3). Both of the above EPs are examples of non-scopal EPs which take no scopal arguments. Generally, non-scopal argument positions are identified with *ref-ind* indices (ARG0 values) and scopal argument positions are identified with *handle* indices (LBL values). An example of an EP that takes scopal arguments, namely the generalized quantifier *every*, is shown below in (5).

(5)
$$\begin{bmatrix} \text{LBL} & \textit{handle} \\ \text{PRED} & \textbf{\_every\_q\_rel} \\ \text{ARG0} & \textit{ref-ind} \\ \text{RSTR} & \textit{handle} \\ \text{BODY} & \textit{handle} \end{bmatrix}$$

All generalized quantifiers take three arguments: a bound variable (ARG0), and two scopal arguments (RSTR(restriction) and BODY). Scopal arguments differ from regular arguments in that a sentence can be underspecified for scope. Copestake et al. (2005) provide the following sentence as an example:

(6) Every dog chases some white cat

This sentence has two possible readings, one where *every* has wide scope (7) and one where the generalized quantifier *some* has wide scope (9). In the below example, (as in Copestake

et al. (2005)), an EP's handle appears immediately before it separated by a colon. I give the first order logic (FOL) equivalents of (7) and (9) in (8) and (10) respectively.

(7) h1: every(x,h3,h5), h3:dog(x), h7: white(y), h7:cat(y), h5: some(y,h7, h4), h4: chase(x,y)

(8) $\forall(x)(dog(x) \Rightarrow \exists(y)(cat(y) \wedge white(y) \wedge chase(x,y)))$

(9) h1: every(x,h3,h4), h3:dog(x), h7: white(y), h7:cat(y), h5: some(y,h7, h1), h4:chase(x,y)

(10) $\exists(y)(cat(y) \wedge white(y) \wedge \forall(x)(dog(x) \Rightarrow chase(x,y)))$

As can be seen above, the only difference between the two readings is the BODY values for the quantifiers. As a consequence, MRS allows the BODY value of a quantifier to be completely unconstrained. However, constraints do need to be placed on the restriction to ensure that no invalid linkings are possible. This is because the restriction position can either be directly linked to an EP handle or one or more quantifiers can come between the restriction position and the EP handle that will fill it. To account for this, each generalized quantifier introduces a qeq (equality modulo quantifiers) constraint. A qeq constraint ensures that a scopal argument is either identified directly with a particular EP handle or that the EP handle in question is identified with the BODY position of the last in a chain of one or more quantifiers that come between the original quantifier and the EP handle.

An MRS is then made up of the EPs of all semantically contentful items involved, the identities between the arguments of EPs, and any constraints required to ensure valid linkings. Specifically, in the Grammar Matrix, the type *mrs* constains three features: HOOK, RELS, and HCONS. RELS contains a list of EPs and HCONs a list of qeq constraints. HOOK takes a *hook* feature structure value which contains features that need to be exposed to the rest of the grammar for semantic composition. This includes LTOP (the top-most handle that is not the handle of a quantifier) and INDEX (a distinguished non-handle variable). An example MRS based on the example provided in Copestake et al. (2005) is shown

below. Certain details are left out compared to Copestake et al. (2005) to focus attention on the concepts that will be most relevant for the analysis in this thesis.

(11) Every dog sleeps

$$\begin{bmatrix} \textit{mrs} & \\ \text{HOOK} & \begin{bmatrix} \textit{hook} & \\ \text{LTOP} & \boxed{9} \\ \text{INDEX} & \boxed{1} \end{bmatrix} \\ \text{RELS} & \left\langle \begin{bmatrix} \text{PRED} & \textbf{\_every\_q\_rel} \\ \text{LBL} & \boxed{2} \\ \text{ARG0} & \boxed{3} \\ \text{RSTR} & \boxed{4} \\ \text{BODY} & \textit{handle} \end{bmatrix}, \begin{bmatrix} \text{PRED} & \textbf{\_dog\_n\_rel} \\ \text{LBL} & \boxed{6} \\ \text{ARG0} & \boxed{3} \end{bmatrix}, \begin{bmatrix} \text{PRED} & \textbf{\_sleep\_v\_rel} \\ \text{LBL} & \boxed{9} \\ \text{ARG0} & \boxed{1} \\ \text{ARG1} & \boxed{3} \end{bmatrix} \right\rangle \\ \text{HCONS} & \left\langle \begin{bmatrix} \textit{qeq} & \\ \text{HARG} & \boxed{4} \\ \text{LARG} & \boxed{6} \end{bmatrix} \right\rangle \end{bmatrix}$$

In (11), the value of RELS is a list containing the EPs of each of the three words in the sentence. The HCONS feature contains the qeq constraint contributed by the generalized quantifier *every*. Finally, HOOK contains the values of the features LTOP and INDEX. The LTOP of the *mrs* is identified with the top-most non-quantifier LBL value (the LBL of the EP for *sleep*), while the INDEX value of the *mrs* is identified with the ARG0 value of the *sleep* EP. In most cases the value of INDEX will be identified with the ARG0 of the main relation introduced by the syntactic head daughter (Flickinger et al., 2003). In this example, this *mrs* might be the result of the noun phrase *every dog* combining with the verb *sleep* which acts as the syntactic head of the sentence. Thus, the INDEX of the overall *mrs* is the ARG0 of the relation introduced by the head daughter.

In general there are three semantic principles governing how the components of an MRS interact with the syntax: (1) the RELS value of the mother of a rule is the appends of the RELS values of its daughter(s), (2) the HCONS of the mother of a rule is the appends of the HCONS values of its daughters(s), and (3) the HOOK value of the mother is identified with the HOOK value of the semantic head daughter (Flickinger et al., 2003). As seen in

the above example involving *sleep* combining with *every dog*, the semantic head of a phrase is often the same as its syntactic head. Within the Grammar Matrix this is encoded through the type *head-compositional* shown below in (12) (Flickinger et al., 2003):

(12)
$$\begin{bmatrix} \textit{head-compositional} & \\ \text{C-CONT} & \begin{bmatrix}\text{HOOK} & \boxed{1}\end{bmatrix} \\ \text{HEAD-DTR} & \begin{bmatrix}\text{SYNSEM.LOCAL.CONT.HOOK} & \boxed{1}\end{bmatrix} \end{bmatrix}$$

This type constrains the HOOK value of the head daughter to be the same as the C-CONT.HOOK value where C-CONT represents the semantic content of the rule itself. A separate constraint on all rules identifies the C-CONT.HOOK value with that of the mother of the rule (Flickinger et al., 2003). In this way the HOOK value of the overall rule is the same as the HOOK value of its head daughter. Several rules inherit from *head-compositional* including the head-comp(lement) and head-subj rules. However, there are also rules where the non-head daughter is treated as the semantic head. In these cases, the rule does not inherit from *head-compositional*, but instead from a separate constraint. One such rule is the head-spec(ifier) rule as the Grammar Matrix treats specifiers as semantic heads (Flickinger et al., 2003).

One final note regarding MRS, is that an MRS for a full sentence is well-formed only if all *ref-ind* indices are bound (Flickinger et al., 2003). This requires all *ref-ind* indices introduced by nouns to be paired with a quantifier. This quantifier can be a generalized quantifier such as *every* or a quantifier associated with a determiner such as *the* in English [eng]. In cases where a noun appears without an overt quantifier, the Grammar Matrix has nouns go through a unary rule called the *bare-np-phrase* rule which introduces an EP with the **_exist_q_rel** relation. This added EP then acts as a quantifier for the noun. *event* indices have no such constraint and do not require explicit quantifiers. This is an important difference in how *ref-ind* versus *event* indices are treated and thus in how the semantics of nouns and verbs are handled. Another area where the distinction between *ref-ind* versus *event* is significant is in adjectival modification, as the Grammar Matrix constrains adjectives to only modify lexical items with a *ref-ind* INDEX value. Given their intermediate nature, an account of

action nominals needs to settle on what INDEX value to use for action nominals (*event* or *ref-ind*) and ensure that the resulting MRS is well-formed given the semantic composition principles of MRS.

2.3 The Grammar Matrix and Customization System

The Grammar Matrix and its customization system make use of both the above discussed formalisms to allow grammar engineers to download formal precision grammars suitable for both parsing and generation. As defined in Zamaraeva et al. (2022) a precision grammar is a computational model of a language which can be used to capture the grammatical predictions made by the model. Thus, a precision grammar only parses sentences seen as grammatical and nothing else. In this way, precision grammars are useful ways to test linguistic hypothesises. However, building such precision grammars is time-consuming.

The Grammar Matrix and its customization system serve as a way to expedite the process of grammar development by allowing the reuse of analyses and grammar code (Bender et al., 2010). The Grammar Matrix is a series of files consisting of core types that are used by all resulting computational grammars. Stored in these files are: (1) types defining the basic feature geometry, (2) types required for MRS semantic composition (for example a type such as *head-compositional* mentioned above), (3) types defining general properties of different classes of rules (unary versus binary rules, head-inital versus head-final rules...), and (4) types defining basic constructions such as head-subj, head-comp, head-spec, and so on (Bender et al., 2002). This core grammar is designed to be maximally general and cannot parse anything on its own. Instead the core grammar is used in conjunction with the customization system which collects language-specific information.

The customization system includes a web-based typological questionnaire which stores a user's answers to a variety of linguistic questions in a text file called a choices file. The questionnaire consists of numerous web-pages pertaining to questions about different linguistic phenomena. For example, the Lexicon page allows users to input information about specific lexical items. Python code makes use of the information in the choices file to add language-

specific types to a series of files that can then be downloaded by the user. The combination of specific questionnaire questions and subsequent types used to handle a particular linguistic phenomena is referred to as a library.

The set of files comprising the final downloadable grammar includes: the core matrix types (`matrix.tdl`), various language-specific types that inherit from the matrix types (stored in a file bearing the name of the language with the .tdl extension), files instantiating specific lexical entries (`lexicon.tdl`) and both lexical (`irules.tdl`, `lrules.tdl`) and phrase structure rules (`rules.tdl`). All these files use the TDL formalism (Type-Description Language) which is described in (Copestake, 2002). TDL is the joint-reference formalism used by the international Deep Linguistic Processing with HPSG Initiative (DELPH-IN) Consortium. The DELPH-IN site[2] provides numerous open-source tools for grammar development including software for both paring and generation. Making use of TDL, the Grammar Matrix is fully compatible with all DELPH-IN resources (Bender et al., 2010).

As the Grammar Matrix has expanded, more and more new libraries have been added to the customization system. The analyses in each of the libraries have the potential to interact and conflict with each other. To ensure that a new analysis does not interfere with an older one, each new library is run through a series of regression tests. The regression tests consist of three parts: (1) a choices file which is used to create a grammar, (2) a testsuite containing both grammatical and ungrammatical strings, and (3) the gold results of using the grammar to parse the testsuite which includes both the syntactic parses and semantic representations of each grammatical testsuite string (Bender et al., 2010). Running a regression test involves using the choices file to create a new grammar which is then used to parse the testsuite. The results of the newly produced grammar on the testsuite are then compared to the expected gold results. Any addition to the Grammar Matrix must be able to pass these regression tests. Additionally, after the development of a library, the testsuites and grammars used during development and evaluation are added to the regression testsuite. Many of these

[2] `https://github.com/delph-in/docs/wiki/`

testsuites include pseudo languages meant to test specific combinations of choices, but they also include examples from natural languages as well. Thus, one step in the evaluation of the library for action nominal constructions will be to ensure that it successfully passes all the existing regression tests, of which there are currently 563.

On the note of interacting analyses, it is important to discuss the details of several libraries which are of major importance to action nominal constructions. As will be discussed in the next chapter, action nominal constructions are parasitic using head and dependent marking strategies seen in finite clauses and possessive noun phrases as opposed to any unique syntactic means (Koptjevskaja-Tamm, 2013). To this end, the next two subsections give brief overviews of the adnominal possession (§2.3.1) and morphotactics (§2.3.2) libraries since a basic understanding of how these phenomena are analyzed outside of action nominal constructions is necessary to understand how they are integrated into the analysis that will be presented in this thesis. Finally, given that the action nominal construction library is building on top of the normalized clauses library, the last section (§2.3.3) in this chapter describes this library in detail to establish what the existing library can already cover.

2.3.1 Adnomominal Possession

As mentioned above, action nominal constructions often use the same syntactic mechanisms as those found in possessive noun phrases. In the Grammar Matrix, both the syntax and semantics of possessive noun phrases is handled by the adnominal possession library (Nielsen, 2018). The library offers both a specifier and a modifier analysis of possessors. However, only the specifier analysis will be discussed here as no strong evidence has yet been found which suggests that the modifier analysis is relevant for action nominals.

The semantics of possession are achieved through the addition of the **poss_rel** relation to the RELS list of the MRS representation of the possessve noun phrase. An **_exist_q_rel** relation is also added to the RELS list to serve as a quantifier for the possessum. An example of the RELS list for the phrase *the dog's cat* is shown below in (13):

(13)
$$\begin{bmatrix} \textit{mrs} \\ \text{RELS} \left\langle \begin{bmatrix} \text{PRED} & \textbf{def\_q\_rel} \\ \text{LBL} & \boxed{1} \\ \text{ARG0} & \boxed{2} \\ \text{RSTR} & \boxed{3} \\ \text{BODY} & \textit{handle} \end{bmatrix}, \begin{bmatrix} \text{PRED} & \textbf{\_dog\_n\_rel} \\ \text{LBL} & \boxed{4} \\ \text{ARG0} & \boxed{2} \end{bmatrix}, \begin{bmatrix} \text{PRED} & \textbf{poss\_rel} \\ \text{LBL} & \boxed{5} \\ \text{ARG0} & \textit{event} \\ \text{ARG1} & \boxed{6} \\ \text{ARG2} & \boxed{2} \end{bmatrix}, \right. \\ \left. \begin{bmatrix} \text{PRED} & \textbf{exist\_q\_rel} \\ \text{LBL} & \boxed{8} \\ \text{ARG0} & \boxed{6} \\ \text{RSTR} & \boxed{7} \\ \text{BODY} & \textit{handle} \end{bmatrix}, \begin{bmatrix} \text{PRED} & \textbf{cat\_n\_rel} \\ \text{LBL} & \boxed{5} \\ \text{ARG0} & \boxed{6} \end{bmatrix}, \right\rangle \end{bmatrix}$$

poss_rel takes two ordinary arguments corresponding to the possessor and the possessum. In (13) the ARG1 of **poss_rel** is identified with the ARG0 of the possessum (*cat*) and the ARG2 of **poss_rel** is identified with the ARG0 of the possessor (*dog*). Additionally, **poss_rel** shares the same LBL value as the LBL of the possessum since they share the same position in the scope tree. Depending on the marking strategy, **poss_rel** is either introduced by a lexical rule, a unary phrase structure rule, a binary phrase structure rule, or a lexical entry.

The library allows for zero, single, and double marking with marking defined as either possessive affixes or separate words that can appear with the possessum or possessor. In the case of zero-marking a special phrase structure rule (*poss-phrase*) combines the possessor and possessum and introduces possessive semantics. In all other cases the possessor and possessum are combined through the regular head-spec rule.

Both the possessum and possessor can be marked with a possessive affix or word. When the possessor takes a possessive word, the word is analyzed as an adposition which takes the possessor as a complement. Alternatively, when the possessum is marked with a possessive word, the possessive word is analyzed as a noun which take the possessum as a complement. Whenever the possessor is an independent word and either the possessor or possessum is marked with an affix or a separate word, possessive semantics are introduced through a unary rule which takes the possessor as a daughter. The mother of this unary rule is what acts as the specifier for the possessum in the head-spec rule. The above mechanisms account

for common noun possessors, however the library also offers an analysis for special possessor pronouns. As above, the possessor pronouns can be realized as either independent words or as affixes. Independent possessor pronouns are given their own lexical entry and serve as daughters to the same unary phrase rule mentioned above. When the pronoun is an affix, possessive semantics are introduced through the lexical rule associated with the affix. The library also introduces several new features to account for differences in word order between determiners and nouns and possessors and nouns as well as to handle agreement (in person, number, or gender) between the possessor and possessum.

Given the richness of the analysis already present in the adnominal possession library and how often marking in action nominals is identical to that of adnominal possessives, it would be ideal to make as much use of the existing machinery in the adnominal possession library as possible. At the same time, it might be desirable to distinguish between the two with regards to elements such as their semantics. Likewise, it is important not to over-generate and allow action nominals to enter into ungrammatical possessive constructions.

2.3.2 Morphotactics

Another library important to discuss in the context of action nominal constructions is the morphotactics library, as action nominals often make use of a mixture of both nominal and verbal morphology. The morphotactics library was originally designed by O'Hara (2008) and then expanded by Goodman (2013). The following description of the library comes from Goodman (2013). The customization system allows users to create position classes. A position class (pc) is a collection of lexical rule types (lrts) that all occur in a certain part of a word. Lrts are associated with different morphemes and, alongside the pcs they are a part of, can be optional or mandatory. Users can specify which lrts occur in each pc and which other pcs, lexical types, and lrts can serve as input to a given pc. As mentioned briefly in the HPSG section (§2.1), lexical rules are differentiated from phrase structure rules in always occurring lower in the tree. The morphotactics library enforces this notion by requiring that the input to all phrase structure rules be inflectionally satisfied. Goodman (2013) defines

inflectional satisfaction as a legal application of lexical rules. This means that all required morphemes are present and that all co-occurrence restrictions between different morphemes are met.

A feature called INFLECTED is used to track whether a lexical item has obtained inflectional satisfaction. This feature takes a value of type *inflected* which can consist of various flags. A flag is created for each lexical rule that either must occur or has a co-occurrence restriction. Flags can take three values ($-$, $+$, *na*) with all flags starting off with the initial value *na-or*$--$. This value can unify with a value of *na* or a value of $-$. A flag is in a satisfactory state when it can unify with the value *na-or-*$+$. The lexical rule associated with a flag changes its value while all other flags are passed up from rule daughter to rule mother unchanged. A flag value of $+$ indicates that the associated lexical rule has occurred, a value of $-$ indicates that the lexical rule still needs to occur, and a value of *na* indicates that the lexical rule has not occurred and there is no requirement for it to occur. Any flag with a value of $+$ or *na* is considered to be in a satisfactory state. The use of these flag values allows the system to determine which lexical rules can and must follow one another and which ones can never co-occur. Since action nominals can potentially take both verbal and nominal morphology, one part of the analysis will require ensuring that inflectional satisfaction is still occurring properly. This entails both forbidding action nominals from taking ungrammatical verbal morphology, while allowing them to take all necessary or optional nominal morphology.

2.3.3 Nominalized Clauses

Finally, it is essential to discuss the current functioning of the nominalized clauses library (Howell et al., 2018) to understand the expansions I make to accommodate a larger variety of action nominals. In addition, I keep many parts of the library the same and integrate them into the new analysis I describe later on in this thesis. The current nominalized clauses library is designed to handle affixal nominalization when used as a marking strategy in complement clauses and adverbial modifiers. The library accounts for both case-frame

changes in nominalized clauses and the option to allow nominalized clauses to be modified by either adjectives, adverbs or both with certain restrictions that will be discussed below.

The nominalized clauses subpage of the customization system questionnaire gives users the freedom to create any number of nominalization strategies by answering a few questions regarding the strategy. Affixes can then be defined on the morphology page which take one of the defined nominalization strategies as a feature value. A user can also specify that the nominalized verb resulting from the addition of this affix should mark its arguments with different cases compared to the standard case-marking pattern for verbs.

For each defined nominalization strategy the user is asked to provide a name (which will be used as a value for a nominalization feature), to state at what level nominalization occurs (low, mid or high) and whether the nominalization is purely syntactic or should also be reflected in the semantics. The library affords different options based on which level of nominalization is selected. Low nominalization allows case changes on both the subject and object, mid nominalization only on the subject, and high nominalization forbids case changes on either the subject or object. With repect to adjectival/adverbial modification, low nominalization only allows adjectival modification, high nominalization only allows adverb modification, and mid nominalization allows both. The ability to keep the semantics free of the effect of nominalization is reserved exclusively for high nominalization.

Moving on to the implementation of the library, nominalization is defined by the change in HEAD value from *verb* to *noun*. Different nominalization levels correspond to the point in the tree when this change occurs. Low refers to nominalization at the V (verb) level, mid to nominalization at the VP (verb phrase) level, and high to nominalization at the S (sentence) level. This effectively switches the word or phrase from having the external distribution of a verbal category to that of a nominal one.

Regardless of nominalization level, a nominalization lrt (an lrt which takes a nominalization strategy as a feature value) changes the value of a newly introduced boolean feature NMZ from – to + indicating that the verb has been morphologically nominalized. This effectively separates the actual point of nominalization (the change in HEAD value which occurs either

low, mid, or high) from the point of affixation which always occurs low. All non-derived nouns and verbs are constrained to be [NMZ −], and thus the NMZ feature serves as a way of differentiating between non-derived nouns ([NMZ −]) and nominalized verbs ([NMZ +]) as well as non-nominalized verbs ([NMZ −]) from verbs which have a nominalization affix ([NMZ +]) but are still HEAD *verb*.

The semantics of nominalization is achieved through the addition of the relation **nominalized_rel** to the RELS list of the verb. **nominalized_rel** is shown below in (14):

(14)
$$\begin{bmatrix} \text{PRED} & \textbf{nominalized\_rel} \\ \text{LBL} & \textit{handle} \\ \text{ARG0} & \textit{ref-ind} \\ \text{ARG1} & \textit{handle} \end{bmatrix}$$

Both the LBL value and the ARG0 value of nominalized_rel are identified with the LTOP and INDEX values of the mother of the rule where the relation is added. The ARG1 value is connected through a qeq constraint with the LTOP value of the original verb. This effectively wraps a nominal predication around the EP provided by the verb. The fact that ARG0 is of type *ref-ind* and not *event* also allows the nominalized verb to be bound by a quantifier and to be modified by adjectives.

In the case of low nominalization, the nominalization lrt does all three tasks of changing the value of NMZ to +, changing the HEAD value of the lexical item to *noun*, and introducing **nominalized_rel** to the RELS list of the verb. Instead of simply copying up the COMPS and SUBJ values between the mother and daughter, the nominalization identifies just the INDEX values of both the COMPS and SUBJ features. This allows the CASE and HEAD values of the both the SUBJ and COMPS of the original verb to be replaced with new values based on the user's input. The ability to change the HEAD value of an argument is important as certain cases may be associated with specific HEAD values such as if a language has case-marking adpositions.

In mid nominalization, the nominailiation lrt just changes the NMZ feature to have a + value and makes it possible to change the CASE and HEAD value of the subject, but not

the object. A unary phrase structure rule which takes a [NMZ +] VP as a daughter is then responsible for changing the HEAD value of the nominalized verb from *verb* to *noun* and introducing **nominalized_rel** to the RELS list. Introducing nominal semantics higher up in the tree allows the nominalized verb to be modified by both adverbs (prior to the phrase structure rule) and adjectives (after the phrase structure rule).

Finally, high nominalization works in a similar way to mid nominalization except that the nominailization lrt copies up the SUBJ and COMPS values between the mother and daughter meaning there can be no case-change in high nominalization. The unary high nominalization phrase structure rule then takes a [NMZ +] S as a daughter, changes the HEAD value to *noun* and adds **nominalized_rel** to the RELS list. For only high nominalization there is also an option to leave **nominalized_rel** out of the semantics which is achieved by a separate phrase structure rule which performs the same function as the typical high nominalization phrase structure rule, but does not introduce **nominalized_rel**. This results in a nominalized clause with the same semantics as non-nominalized subordinate clauses.

For both mid and low nominalization, special non-event head-subj rules are also added to the grammar. This is necessary since mid and low nominalization both add **nominalized_rel** to the semantics prior to the combination of the nominalized verb with its subject. The INDEX value of the overall nominalized phrase is then *ref-ind*. This prevents the nominalized verb from using the typical head-subj rules which require the head-daughter to be of INDEX *event*.[3] The non-event head-subj rules then perform exactly the same function as the regular head-subj rules just for lexical items with a *ref-ind* index.

Thus, in the current library, nominalization is defined by a change in HEAD value from *verb* to *noun* with the ability to alter the case-marking of arguments and choose whether the nominal verb is modified by adjectives or adverbs dependent on at which level in the parse tree nominalization occurs.

---

[3]Specifically, the head-subj rules inherit a constraint from the type *declarative-clause* which specifies that a declarative clause must have an SF value of *prop-or-question* (proposition or question). SF is a feature exclusive to *event* indices.

2.4 Summary

This chapter has provided an overview of the syntactic (Head-Driven Phrase Structure Grammar) and semantic (Minimal Recursion Semantics) frameworks underlying the Grammar Matrix and thus the analysis of action nominals that I present later in this thesis. In the chapter, I also went into more detail about the Grammar Matrix and its customization system including specifics about libraries of particular importance to action nominal constructions. The next chapter will provide an overview of the typology of action nominal constructions as well as discuss relevant prior syntactic analyses with a focus on those within the HPSG framework.

Chapter 3
LITERATURE REVIEW

This chapter is divided into an overview of the typology of action nominal constructions (ANCs) and a brief discussion of prior proposed syntactic analyses. The first half offers a preview of the range of phenomena that I attempt to cover in the new action nominal construction library. The second half goes over several problems found with approaches similar to the one in the current nominalized clauses library and outlines alternative analyses that will allow for the analysis of additional nominalization strategies not yet handled by the current library. Prior to the discussion of ANCs in different languages, section 3.1 introduces important terminology that will be used when discussing ANCs throughout the remainder of this thesis. Section 3.2 begins by detailing the ANC typology in Koptjevskaja-Tamm (1993) which serves as a framework for the analysis presented in this thesis. The following subsections provide information about different properties of ANCs that hold across the categories defined in the typology and which the analysis aims to handle. The bulk of the material to be discussed in this section comes from overviews given in Comrie (1976), Comrie and Thompson (1985), and Koptjevskaja-Tamm (1993). Section 3.3 ends by discussing several syntactic analyses that have been proposed for ANCs including both theoretical and implemented approaches.

3.1 Terminology

This section introduces a few terms that will be used consistently throughout the remainder of the thesis. As described in chapter 1, an action nominal is defined as a nominalized verb denoting an action or process which can take nominal morphology and/or take adposi-

tions exclusive to nouns (Comrie, 1976).[1] An ANC is a phrase headed by an action nominal with the dependents of the action nominal denoting the participants of the event expressed by the verb from which the action nominal is derived. For example, there is a direct parallel between the finite clause in (1) and the ANC in (2)

(1) *the enemy destroy-ed the city*
the enemy destroy-PST the city
'The enemy destroyed the city.' [eng]

(2) *the enemy 's destroy-tion of the city*
the enemy POSS destroy-NMLZ of the city
'The enemy's destruction of the city.' [eng]

In both cases, the entity performing the destruction (the agent) and the entity being destroyed (the patient) are clear. In the context of ANCs, I will refer to the agent-like argument as the A and the patient-like argument as the P. I also use A to refer to the single argument of intransitive action nominals. Both A and P refer only to semantic roles and not syntactic positions. Thus, the A of an ANC is not necessarily the subject of the action nominal. However, generally, the A will have the same semantic role as the subject of the finite clause, and the P will have the same semantic role as the direct object of the finite clause. Conversely, as shown in (2), ANCs often mark arguments with the same strategies used to mark possessors in possessive noun phrases. Compare (2) to the noun phrase in (3) where both *'s* and the preposition *of* can be used to mark adnominal possessors in English:

(3) *the enemy 's map of the city*
the enemy POSS map of the city
'The enemy's map of the city.' [eng]

To this end, any argument taking possessive marking in an ANC will be referred to as a Poss regardless of whether it is a semantic possessor or not. Taking possessive marking

[1]Throughout this thesis action nominals and nominalized verbs will be used interchangeably although in general nominalized verbs can extend to a much larger category of items (argument nominalizations) then just action nominals.

refers not only to having possessive morphology, but also assimilating to the word order seen in possessive noun phrases. Thus, if possessors in possessive noun phrases occur before the head noun, then the Poss argument in an ANC would also occur prior to the head action nominal. In line with these conventions, the A in (2) is the *the enemy*, the P is the *the city*, and both arguments are Poss arguments since they both take different types of possessive marking in English.[2]

3.2 Typology of Action Nominal Constructions

One primary cross-linguistic generalization regarding ANCs is that they are parasitic, using marking strategies found in either finite clauses or non-derived noun phrases, but by and large not making use of any new or unique marking strategies (Koptjevskaja-Tamm, 2013). This adoption of existing marking strategies causes ANCs to differ both across languages and within the same language as to whether they more closely resemble finite clauses or non-derived noun phrases. Based on the properties of ANCs in a sample of 70 languages, Koptjevskaja-Tamm (1993) breaks ANCs into four main categories which range from looking fully verbal to looking fully nominal. The classification is primarily based on the dependent marking of both the A and the P, with head marking used as an additional criterion when necessary. Dependent marking strategies include case-marking or the use of a specific adposition (Koptjevskaja-Tamm, 1993).

It generally holds that all non-core arguments (arguments that are neither the A nor the P) of the verb will be marked the same in finite clauses and ANCs (Comrie and Thompson, 1985). In addition to the four main categories, there are also a few additional minor ANC types that will not be discussed here due to either being very rare, or relying on phenomena such as relative clauses that are not yet present in the Grammar Matrix customization system.

---

[2]I include the examples from English [eng] to demonstrate how the terminology is used, but chapter 4 provides more discussion about whether both arguments in (2) should actually be treated as Poss arguments. While *'s* meets the criteria for a possessive marker in ANCs, the *of* preposition is more debatable.

The four main types are sentential (SENT), Possessive-Accusative (POSS-ACC), Ergative-Possessive (ERG-POSS) and Nominal (NOMINAL). The following subsections will discuss each category in detail.

3.2.1 Sentential

The SENT type describes ANCs where all dependents are marked in the same way as they would be with the non-nominalized verb. For example, Godoberi [god] finite clauses require transitive subjects to take ergative case while intransitive subjects and direct objects take absolutive case. As (4) shows, even in ANCs, this same pattern of case-marking is observed with *aHmadi* 'Ahmad' (the A) taking ergative case and *rec'i* 'bread' (the P) taking absolutive case (Koptjevskaja-Tamm, 2013).

(4) (SENT)
aHmadi-di *maHamadi-łi* *rec'i* *ik-ir*
Ahmad-ERG Mahamad-DAT bread.ABS give-NMLZ
'Ahmad's giving bread to Mahamad.' [god] (Kazenin, 1996: 260)

(4) also showcases the indirect object *maHamadi* 'Mahamad' taking dative case which marks it as the recipient of the action. This is the same case as would be used in a finite clause. As mentioned above, this is expected of all non-core arguments across all the ANCs categories and not unique to the SENT type.

3.2.2 POSS-ACC

In the POSS-ACC type, the P retains sentential marking, while the A is marked as a Poss. As example (5) from Meadow Mari [mme] shows, the nominalized verb *voz-öm* 'write' takes a possessive suffix and the A *möj* 'I' is marked with genitive case, which is also used to mark possessors in non-derived noun phrases (Koptjevskaja-Tamm (2013)). Meanwhile, the P *pis'ma* 'letter' takes accusative case which is the same case used to mark direct objects in finite clauses (Koptjevskaja-Tamm, 2013).

(5) (POSS-ACC)
möj-ön *pis'ma-m* *voz-öm-em*
I-GEN letter-ACC write-NMLZ-1SG.POSS
'My writing of the letter.' [mme] Koptjevskaja-Tamm (2013)

3.2.3 ERG-POSS

In the ERG-POSS type, both intransitive As and Ps are marked as Poss arguments, while transitive As take an oblique form of marking. In Russian [rus] (6), the P *sonat* 'sonata' takes genitive case as would the possessor in a non-derived noun phrase (Koptjevskaja-Tamm, 2013). In contrast, the A *pianist* 'pianist' takes instrumental case which can also be used to mark agents in passive sentences (Koptjevskaja-Tamm, 2013).

(6) (ERG-POSS)
ispoln-enij-e *sonat-y* *pianist-om*
Perform-NMLZ-NOM sonata-GEN pianist-INSTR
'The performance of the sonata by the pianist.'[3] [rus] Koptjevskaja-Tamm (2013)

3.2.4 Nominal

Finally, in the NOMINAL type all arguments have markings that are typical of non-derived noun phrases. This either means that both the A and P are marked as a Poss, or that the A is marked as a Poss while the P takes an oblique form of nominal marking (Koptjevskaja-Tamm, 1993). The English [eng] example in (7) serves as an example. The A *enemy's* is marked as Poss while the P *of the city* could arguably be analyzed as either a Poss or an oblique form of marking by the nominal preposition *of*.

(7) (NOMINAL)
the *enemy 's* *destroy-tion* *of the city*
the enemy POSS destroy-NMLZ of the city
'The enemy's destruction of the city.' [eng]

[3]As in Russian [rus], across many languages, the A in ERG-POSS ANCs takes the same marking as the agent of a passive verb, however this is not a requirement of the ERG-POSS category (Koptjevskaja-Tamm, 1993). The A just needs to take an oblique form of marking (something distinct from how it would be marked in finite clauses, while also not taking Poss marking).

3.2.5 Nominal versus Verbal Properties

While the typology in Koptjevskaja-Tamm (1993) concentrates on dependent marking there is also the question of what morphological marking the action nominal head itself can take. Here morphological marking means that a certain property is expressed by a grammatical morpheme in the ANC (Comrie and Thompson, 1985). Comrie and Thompson (1985) step through a number of verbal (tense, aspect, voice, transitivity, negation) and nominal (case, number, definiteness) properties discussing which of these are commonly found in ANCs. They find that morphological expression of tense, aspect, voice, and transitivity are often lost in ANCs, but there are exceptions in different languages for all four features. In contrast, negation is usually possible to express in ANCs, although languages vary in whether they use verbal means of negation, nominal means of negation, or a mechanism distinct to ANCs.[4] With respect to the nominal properties, it is generally true that if non-dervied noun phrases showcase case and definiteness then so do ANCs. However, number is often excluded from ANCs. This is attributed to the fact that action nominals fall into the same class as abstract nouns, which also do not take number (Comrie and Thompson, 1985).

Looking at these observations about nominal and verbal properties in the context of the four types in the classification system proposed by Koptjevskaja-Tamm (1993) there appears to be a correlation between ANC types and the presence or absence of verbal properties. For example, ANCs of the SENT type are more likely to retain certain verbal characterisitics such as tense distinctions (Koptjevskaja-Tamm, 1993). However, the mapping is not fully one-to-one and two languages which have ANCs of the same type can still differ in whether a verbal property such as aspect is loss or retained in ANCs (Koptjevskaja-Tamm, 2015). Thus, while it is necessary for the library introduced here to allow action nominals to express nominal properties such as case, while also losing certain verbal properties such as tense, this process needs to be flexible and customizable for different languages.

---

[4]The interaction of negation with ANCs is considered out of scope for this thesis.

3.2.6 Optional Arguments

One striking difference between ANCs and finite clauses is the increased tendency for the nominalized verb to drop arguments even if they are mandatory for the underlying verb. Koptjevskaja-Tamm (1993) notes that it is quite common for ANCs to lack one or more arguments, and that ANCs with all arguments present are relatively rare. Comrie (1976) uses English [eng] as an example to point out that in English [eng] ANCs, such as the *hunter's shooting of the lion*, both arguments *the hunter's* and *of the lion* are optional (*the shooting*),[5] despite the verb *shoot* requiring both an overt subject and object. To add to this, Comrie and Thompson (1985) claim that cross-linguistically it is always possible to leave the A of an action nominal unexpressed.[6] Thus, overall, action nominals seem to have more flexible constraints on overt complements compared to the verbs from which they are derived. Since this is such a common characteristic of action nominals, this is a contrast between action nominals and their underlying verbs that the library should be able to handle.

3.2.7 Single Argument

As mentioned in the previous subsection, ANCs very frequently drop one or more arguments. ANCs taking on a single argument often show a distinct pattern compared to the ANC with all arguments present (Koptjevskaja-Tamm, 2003). For example, as shown in (6) in Russian [rus] when both the A and P appear with an action nominal, the A takes instrumental case and the P takes genitive case. In contrast, if only one argument appears it always takes genitive case and is ambiguous between an A or P (Comrie, 1976). This Russian [rus] example demonstrates what Koptjevskaja-Tamm (2003) describes as a cross-linguistic tendency for ANCs with a single argument to assimilate to the syntax of non-derived noun phrases. In other words if an ANC contains only a single argument, that argument will

[5]In English, the action nominal still needs to appear with an article.

[6]A different form of the action nominal may need to be used when the A is expressed versus when it is not. For example, in Tagalog [tgl], nominalizing the aspectless form of a verb results in an abstract noun that cannot appear alongside an A (Comrie and Thompson, 1985).

often be marked as a Poss regardless of how it is marked when the ANC takes additional arguments.

3.2.8 Word order

Since many languages have the same word order across finite clauses and non-derived noun phrases, this same word order is also observed in ANCs. However, Koptjevskaja-Tamm (1993) notes that in their sample of 70 languages, all languages with different head-dependent orders in finite clauses versus noun phrases, had ANCs which followed the nominal pattern. One example of this is seen in Hixkaryana [hix] which has OVS word order in finite clauses, but where the preferred word order in intransitive ANCs is SV (A action.nominal) (Derbyshire, 1985). In Hixkaryana [hix], possessors in non-derived noun phrases always immediately precede their heads. Since As in intransitive ANCs are marked as a Poss, they follow this same pattern and appear before the action nominal head. Thus, instead of relying on the word order in finite clauses where subjects occur after their heads, ANCs instead use the same dependent-head word order as in possessive noun phrases. To handle such languages an analysis of ANCs must be able to account for cases where head-dependent order is different for finite clauses versus non-derived noun phrases.

3.2.9 Adverb/Adjectival Modification and Coordination

Two remaining dimensions along which different languages tend to vary is in whether action nominals can be modified by adverbial or adjectival modifiers and whether or not they can coordinate with non-derived noun phrases (Koptjevskaja-Tamm, 1993). In different ANCs either adjectival or adverbial modifiers are more acceptable, with certain languages allowing for both (Comrie and Thompson, 1985). However, Koptjevskaja-Tamm (1993) notes that making the determination about whether a particular ANC uses either adjectival or adverbial modification can be quite complicated. Many languages simply do not have this distinction to begin with while others have very particular requirements, with certain lexical items appearing as adverbs in ANCs while others are only acceptable as adjectives.

Languages also differ in whether ANCs can coordinate with non-derived noun phrases or whether this is impossible (Koptjevskaja-Tamm, 1993).

From the perspective of the typology in Koptjevskaja-Tamm (1993) there does seem to be a tendency for more verbal ANC types to include more verbal characteristics (adverbial modifiers, inability to coordinate with non-derived noun phrases) however this rule is not always followed and there is considerable variation across different languages (Koptjevskaja-Tamm, 1993). For example, Koptjevskaja-Tamm (1993) notes that there are languages with Nominal ANCs that allow adverbial modifiers. This is unexpected given the idea that Nominal ANCs are closer to nominal syntax and would be expected to always take adjectival modifiers. Therefore, these two properties (modification and coordination) appear to be another place where flexibility is required to allow for the patterns seen in different languages.

3.2.10 Summary

The above has provided an overview of the typology of ANCs proposed by Koptjevskaja-Tamm (1993) as well as gone over various additional properties of ANCs including to what extent they contain nominal versus verbal properties, their tendency to allow optional arguments, and special marking patterns seen for stand-alone arguments. Also discussed were several dimensions along which ANCs vary across languages, such as whether there are word order differences between ANCs and finite clauses, whether they can be modified by adjectival or adverbial modifiers, and whether they can be coordinated with non-derived noun phrases.

The four main types proposed by Koptjevskaja-Tamm (1993) serve as a good starting point for the most prominent kinds of ANCs found in the world's languages and an ideal account of ANCs should at least be able to account for these four major categories. At the same time, the other properties discussed such as retention or loss of verbal morphology and adjectival/adverbial modification do not map neatly onto any one of the four ANC types. It is then necessary to provide a means to model these four categories while also allowing

different languages to select the more variable properties that either do or not apply to them. Now that I have outlined the properties of ANCs in different languages, in the next section, I describe several syntactic analyses that have been proposed to account for their mixed (both nominal and verbal) behavior.

3.3 Syntactic Analyses

3.3.1 Syntactic versus Lexicalist approaches

Nominalizations have a long history in the linguistic literature with a significant amount of debate centering around whether the mixed nominal and verbal properties of nominalizations stem from the lexicon or the syntax (Alexiadou, 2010). Lexicalist approaches claim that nominalizations contain no verbal syntax and that all verbal properties come from lexical processes. In contrast, syntactic approaches argue that the verbal properties exhibited by nominalizations (such as adverbial modification or sentential dependent marking) are the result of nominalized clauses containing verbal syntax which at minimum includes a verb phrase.

As described by Malouf (1998) in a review of approaches modeling English [eng] action nominals (referred to by Malouf (1998) as verbal gerunds), syntax-based analyses tend to revolve around a notion of categorical change-over. That is while the exact details of proposals differ, they all consist of a verbal head dominated by a VP which is at some point dominated by an NP. This embedding of a VP within an NP is claimed to explain the more verbal properties of action nominals based on how high in the tree the change-over occurs. The prototypical structure that Malouf (1998) describes is shown in figure 3.1.

3.3.2 Strengths of syntactic change-over approaches

The main advantages of change-over approaches are: (1) their ability to account for the verbal characteristics of action nominals and (2) the predictions they make regarding which nominalization types are observed in the world's languages. To the latter point, as observed by Croft (1991), action nominals do not seem to allow any random combination of nominal

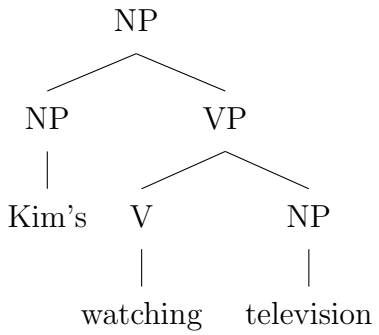


Figure 3.1: Basic structure found in change-over approaches (Malouf, 1998: 87)

and verbal features, but instead appear to retain verbal features in a certain order. The specific generalization given by Croft (1991) is that any verb form which takes tense-aspect-mood marking will mark its arguments in the same way as a finite verb. Additionally, any verb form which marks its subject in the manner seen in finite clauses will also mark its object using the typical verbal pattern. Certain feature combinations such as an action nominal which marks its P as a Poss (the nominal pattern), but marks its A using the verbal pattern are then predicted to be impossible or at least very rare. Change-over approaches offer an explanation for this generalization through what Malouf (1998) terms the phrasal coherence hypothesis. That is, change-over approaches predict that there will be a single point in the tree where the change over between noun and verb occurs. Everything below the point of change-over is predicted to be a coherent verb phrase and everything above a coherent noun phrase. The only valid normalization types are then those that respect this clean division between the nominal and verbal parts of the ANC.

By this definition, the nominalized clauses library (Howell et al., 2018) provides an analysis somewhat in the middle between the lexicalist and syntactic paradigms. On the one hand, high and mid nominalization use phrase structure rules to change the HEAD value of the verb at different points in the tree, with high nominalization allowing more verbal properties (only adverbial modification is permitted) due to containing a full clause (an S), while mid

nominalization allows more mixed properties (both adjectival and adverbial modification are allowed) due to only containing a VP. This falls in line with the idea that the level of categorical change-over is responsible for the combined nominal and verbal characteristics of the overall phrase. However, in low nominalization all changes to the properties of the original verb are achieved through a lexical rule without the need for any phrase structure rules. Thus, in low nominalization, the nominalized clause does not contain any verbal syntax with any verbal properties retained by the nominalized verb stemming from lexical processes. The nominalized clauses library then offers both a syntactic and lexicalist option with certain limitations such as the fact that adjectival/adverbial modification can only be controlled by the level of nominalization and cannot be made fully lexical. This approach was chosen for the nominalized clauses library as it integrates well with other libraries in the Grammar Matrix and is able to account for both differences in adjectival/adverbial modification and case-frame changes seen in different languages (Howell et al., 2018).

3.3.3 Problems posed by syntactic change-over approaches

Despite its intuitive appeal, Malouf (1998) discusses a number of challenges with the change-over approach including that at least certain formulations of it violate the endocentricity principle. Endoncentriy refers to the notion that every phrase should have a distinguished head (Malouf, 1998). In HPSG the head is further specified to be the element with roughly the same distribution as the phrase (Malouf, 1998). The approach in the nominalized clauses library violates this principle since the head daughter of a nominalized clause does not share the same distribution as the overall nominalized clause in mid or high nominalization (where the head daughter intially has a *verb* HEAD value). This problem is acknowledged by Howell et al. (2018) and accepted as a compromise to better suit the computational goals of the Matrix.

Another issue brought up by Malouf (1998) is that there is no clear way to block adjectival modification in what the nominalized clauses library calls mid nominalization. An example of an ungrammatical English phrase that would not be blocked is shown in (8). The action

nominal in this example is *watching* and the overall ANC is an instance of mid nominalization since the P (*movies*) is marked in a verbal manner while the A (*girl*) is marked as a Poss. Being mid nominalization, the change-over is predicted to occur at the VP level meaning everything below *watching movies* is expected to be verbal and everything above nominal. This should theoretically allow for adjectival modification above the cross-over point. While such a phrase might be grammatical in other languages, the English example demonstrates that the choice of adjectival/adverbial modification needs to be more flexible than the change-over approach allows.

(8) **the girl 's avid watch-ing movies*
the girl POSS avid watch-NMLZ movies
'The girl's avid watching movies.' [eng]

Malouf (1998) further points out problems with both of the attested strengths of change-over approaches discussed above. He argues that there is insufficient evidence to claim that action nominals contain verbal syntax since their verbal properties, such as adverbial modification or ability to take accusative noun phrases (at least in English), are not exclusive to verb phrases. For example, adverbs modifiy not just verbs but also adjectives and adpositions. Malouf (1998) also discusses numerous cases where the phrasal coherence hypothesis fails to make correct predictions. One example with particular relevance to the nominalized clauses library is the claim in Malouf (1998) that change-over approaches cannot effectively handle VSO languages.

As stated above, the phrasal coherence hypothesis predicts that there is a clear division between the nominal and verbal parts of an ANC and thus a structure such as the one shown in figure 3.2 should be impossible.

This creates a problem when considering examples such as the following in (9) from Modern Standard Arabic [arb]:

(9) *ʔaqlaqa-nii ntiqaad-u zayd-in ᶜamr-an*
annoyed-me criticizing-NOM zayd-GEN amr-ACC
'Zayd's criticizing Amr annoyed me.' [arb] (Fehri, 1993: 223f)

(184)

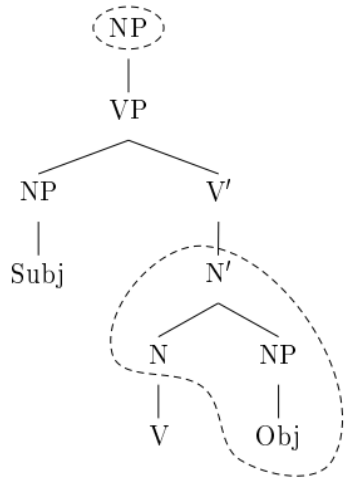


Figure 3.2: Example of a forbidden structure under change-over approaches containing an ANC where the nominal section is split into two parts (shown by the dotted lines) (Malouf, 1998: 127)

In (9) the nominalized verb *ntiqaad* 'criticizing' first combines with the A *zayd* 'Zayd' which is marked in the same way as a possessor in a non-derived phrase and then combines with the P *ᶜamr* 'Amr' which is marked in the typical way for direct objects in finite clauses (Malouf, 1998). Under a change-over approach, this would involve a structure where the verb first changes-over to a noun to combine with *zayd* 'Zayd' and then has to change back to a verb to combine with *ᶜamr* 'Amr'. This is exactly the type of structure change-over approaches claim are impossible.[7]

The nominalized clauses library offers a potential solution by analyzing (9) as an instance of low nominalization with a case change on just the A. However, this seems to go against the central idea of change-over approaches that what determines verbal versus nominal char-

[7]Malouf (1998) does acknowledge that there are formulations of change-over approaches that can handle the ANCs in Modern Standard Arabic [arb], but that they must crucially rely on head-movement. Since head-movment in not a device used in HPSG, an analysis which cannot work without it is not a possibility for the analysis in this thesis.

acteristics is the height in the tree where the change-over occurs. Additionally, the current implementation of low nominalization in the nominalized clauses library would not be able to account for the fact that action nominals in Modern Standard Arabic [arb] can be modified by adverbs, since low nominalization only allows for adjectival modification.

Thus, while certain criticisms such as violating endocentricy or lacking sufficient proof of verbal structure are more theoretical, two practical places change-over approaches, and by extension the nominalized clauses library, struggle is lack of flexibility with adjectival/adverbial modification and accounting for the mixing of nominal and verbal structure in VSO languages.

3.3.4 Lexicalist approaches

As a lexicalist alternative to the change-over approach, Malouf (1998) suggests treating English action nominals as having a new HEAD value (*gerund*) distinct from either *noun* or *verb*. *gerund* is a subtype of both *noun* and another newly introduced type *relational*. Subtypes of *noun* all have the external distribution of a noun phrase while *relational* encompasses all elements that can be modified by adverbs including verbs, adjectives, and, in English, action nominals. Thus, English action nominals are lexical items which have the external distribution of a noun phrase and can be modified by adverbs. In the proposal by Malouf (1998) a lexical rule acts on the verb and changes its HEAD value from *verb* to *gerund* and also makes any changes to the valence properties of the verb such as identifying the SUBJ of the rule daughter with the SPR of the mother.[8] Thus, instead of a category change at some point during syntactic derivation, action nominals represent lexical items

[8]Malouf (1998) goes on to discuss that this lexical rule proposal is perhaps too unconstrained in that there is nothing to prevent any imaginable combination of nominal and verbal features from occurring. This is in contrast to change-over approaches which do make clear predictions about which nominalization types are possible. To remedy this, Malouf (1998) proposes an equivalent analysis to the lexical rule approach using constraints on lexical types. This second proposal is much more restricted. However, it makes use of several mechanisms such as default inheritance which are not yet implemented in the Grammar Matrix. Given that the main goal of the Grammar Matrix is not to predict which linguistic combinations are possible, but to support grammar developers in testing their own hypotheses, the discussion here focuses on the lexical rule analysis which can be straight-forwardly implemented in the Grammar Matrix.

that inherit both verbal and nominal properties. This affords more flexibility with regards to adjectival/adverbial modification and VSO languages since verbal and nominal properties are not dependent on tree height, but on which properties the gerund has inherited.

Hudson (2003) builds off the approach in Malouf (1998) in the context of word grammar (WG) which also uses multiple inheritance, but unlike HPSG has no phrase structure and relies solely on dependency links between words. Hudson (2003) argues that what nouns have in common is their external distribution (ability to function as subjects, objects...) and what verbs have in common is their internal distribution (the types of dependents they take). Furthermore, there is no internal distribution common to all nouns (there is no dependent taken by all nouns) and likewise no external distribution common to all verbs (this is instead determined by inflectional class). This then allows action nominals to be subtypes of both *verb* and *noun* since the properties being inherited will never clash. Action nominals are nouns since they have the external distribution of a noun and verbs since they have the internal distribution of a verb. This means that action nominals can take their semantics and dependents from the verbal properties they inherit while their distribution is determined by their inherited nominal properties.

The approach in Hudson (2003) diverges slightly from that in Malouf (1998) as Hudson (2003) proposes that action nominals are subtypes of nouns and non-finite verbs and not a type such as *relational*. This argument comes from the fact that adverbs do not exclusively modify adjectives, verbs, and action nominals but rather modify everything except for nouns and thus there is no need to create a separate type to handle adverbial modification. Instead, the fact that action nominals can be modified by adverbs stems from the fact that verbs can be modified by adverbs. It should be noted that while this is true for English, the type of hierarchy proposed in Hudson (2003) may be insufficient for languages that allow adjectival modification of action nominals. The notion of having an action nominal inherit the appropriate nominal and verbal properties through lexical processes then seems to offer a promising alternative to solve some of the problems surrounding lack of flexibility encountered with change-over approaches.

3.3.5 Analyses in other DELPH-IN Grammars

The final portion of this chapter discusses two implementations of nominalization in different DELPH-IN Grammars. Being DELPH-IN Grammars, they use the same syntactic (HPSG) and semantic (MRS) formalisms as the Grammar Matrix. They thus provide a good example of language-specific analyses of nominalization that can be generalized to allow for a cross-linguistic handling of action nominals in the Grammax Matrix. Section 3.3.6 discusses the handling of actions nominals in the Korean Resouce Grammar (KRG) (Kim and Yang, 2004), while section 3.3.7 discusses the corresponding analysis in the Hausa Grammar (HaG) (Crysmann, 2012).

3.3.6 Korean Resource Grammar

Kim and Yang (2004) provide an analysis of sentential nominalizations in Korean [kor]. The nominalizations described in the paper correspond to SENT ANCs, having the external distribution of a noun phrase, but the internal distribution of a verb phrase. Thus, Korean [kor] nominalizations can appear in all typical nominal positions (subject, object, and object of an adposition) and can take nominal affixes such as case. However, both the arguments of the ANC take the same case-marking (nominative on the A and accusative on the P) as would the subject and object in a finite clause. Such nominalizations also exhibit other verbal properties such as the ability to be modified by adverbs.

To model these facts, Kim and Yang (2004) define a new *lex-st* type called *v-ger* which inherits from both nominal and verbal subtypes. *lex-st* types form the basic components of the lexicon and are created through lexical processes such as lexical rules (Kim and Yang, 2004). The specific types *v-ger* inherits from ensure that *v-ger* has the argument structure of a verb while also being able to enter into nominal lexical rules and thus take nominal morphology. Multiple inheritance then offers a straight-forward solution to the mixed properties seen in Korean [kor] nominalizations.

The type corresponding to an action nominal (*v-ger*) retains a [HEAD *verb*] value. To handle the nominal distrubition of action nominals, the authors employ two binary features

(NOMINAL and VERBAL), both of which *v-ger* is positive for. Not much justification is provided for these two features other then mentioning that a different kind of lexical item (verbal nouns used in light verb constructions) are likewise [NOMINAL +, VERBAL +], but HEAD *noun* instead of *verb*. Since it is unclear how specific this juxtaposition of light verb constructions and ANCs is to Korean [kor], it appears more consistent with the rest of the Grammar Matrix to treat action nominals as having HEAD *noun* given that they have the external distribution of noun phrases. In general, however, the approach in Kim and Yang (2004) is in line with the English [eng] analyses in Malouf (1998) and Hudson (2003), with the constraints (verbal argument-structure and nominal morphology) inherited by the mixed category lexical item specific to the characteristics of Korean action nominals.

3.3.7 Hausa Grammar

In Huasa [hau], action nominals (called verbal nouns by Crysmann (2011)) are used in the continuative aspect.[9] Action nominals are split into two classes: weak and strong. The following discussion focuses solely on strong action nominals.

Strong action nominals are distinct from regular verbs in taking a genitive linker suffix when an overt P is present and taking bound object affixes from the genitive as opposed to the accusative set (Crysmann, 2011). The genitive linker suffix has a number of functions in Hausa [hau] including marking the possessed noun in possessive noun phrase. Crysmann (2011) argues that the genitive linker serves to mark noun-complement structures and thus that both possessive noun phrases and action nominals and their Ps are cases of a noun taking a complement. This makes the genitive linker the nominal equivalent to existing verbal morphology which serves to mark *in situ* direct objects (Crysmann, 2012).

To account for the behavior of strong action nominals, Crysmann (2011) proposes a lexical

[9]Koptjevskaja-Tamm (1993) classifies the ANCs in Hausa as belonging to one of the minor ANC types (the Relative type). In the Relative type, the P of a transitive (or the A of an intransitive) action nominal appears as a typical nominal dependent, while the A of a transitive action nominal can only be expressed within a relative clause. Relative clauses are not yet implemented in the Grammar Matrix making this ANC type beyond the scope of this thesis. However, the discussion in Crysmann (2011) is included here as it focuses on how the P (and not the A) of transitive action nominals is marked.

rule which changes the HEAD value of an lexical item from *verb* to *noun*, but preserves all other syntactic and semantic constraints of the underlying verb. The genitive linker is introduced by a lexical rule which takes a [HEAD *noun*] daughter and can thus accept both regular nouns and any action nominals which have been though the newly introduced HEAD changing rule. Additional inflectional rules likewise introduce genitive or accusative bound pronouns based on the HEAD value of the lexical item. Thus, the approach in Crysmann (2011) relies principally on a lexical rule which changes the HEAD value of the underlying verb. The resulting action nominal is then able to serve as a daughter to various other lexical rules which require lexical items with a [HEAD *noun*] value.

3.4 Summary

The above section discussed various issues with syntactic change-over approaches to ANCs and what problems these pose for the current nominalized clause library. Namely, change-over approaches and thus the nominalized clauses library can not easily handle action nominals in VSO languages and do not afford sufficient flexibility with respect to adjectival/adverbial modification. I then presented several proposed lexicalist approaches including two implemented in other DELPH-IN grammars. These approaches had in common the idea that action nominals make use of constraints from both nominal and verbal types with different authors varying in whether they treated action nominals as having a distinct HEAD type (*gerund*) or having a HEAD value of *noun* or *verb*. In most approaches the HEAD value of the action nominal allowed it to take on the external distribution of a noun phrase. Additionally, at least two approaches ((Malouf, 1998), (Crysmann, 2011)) used lexical rules to change the HEAD value of a regular verb to the HEAD value designated for action nominals. These lexical rules then also changed certain properties of the verb to allow it to engage in typically nominal syntax (such as taking an argument marked as a Poss).

As a whole, all these approaches made use of lexical processes to ensure that action nominals have the correct combination of categorical and selectional properties expected in a given language. The use of these lexicalist approaches in several DELPH-IN grammars

and the limitations of the current nominalized clauses library then suggest that it may be worthwhile to shift the analysis in the nominalized clauses library from a hybrid change-over/lexicalist approach to a fully lexicalist approach. The next chapter will discuss the analysis of ANCs used to develop the new ANC library building off of the syntactic analyses discussed in this chapter and trying to account for as much of the typological breath as possible.

Chapter 4
ANALYSIS

This chapter provides an analysis of action nominal constructions (ANCs) which covers the phenomena discussed in the previous chapter. In what follows, to avoid confusion, I refer to the analysis of nominalized clauses presented in Howell et al. (2018) as the existing or previous/prior analysis, and my own analysis of action nominal constructions as either my analysis, the new analysis or the proposed analysis.

The analysis I outline in this chapter is specific to morphological nominalization meaning that in all cases there is a morpheme (though it may be a null morpheme) which attaches to the verb and nominalizes it. All other forms of nominalization are not yet covered by either my analysis or the previous one. While still building upon the existing nominalized clauses library, the present analysis moves away from a mixed change-over/lexicalist approach to one that relies as much as possible on lexical processes to account for the behavior of action nominals. Specifically, the lexical rule type (lrt) associated with a nominalization affix does the majority of the work of nominalization while existing mechanisms in the Grammar Matrix are used to account for elements such as word order differences in ANCs or their ability to mark arguments with possessive strategies. This expands the prior library's coverage by allowing for action nominals with a more diverse set of arguments including action nominals which can take syntactic possessors and determiners.

I define the full process of nominalization as consisting of three parts. These are the changes that are necessary for the action nominal to be treated by the rest of the grammar as a noun. The three components are: (1) syntactic properties, including the change in HEAD value from *verb* to *noun* and changes (if present) to the action nominal's valence features relative to the underlying verb, (2) morphological properties (whether the action nominal takes typically verbal affixes or typically nominal ones), and (3) the introduction

of nominal semantics. Under the proposed analysis, (1) and (2) are always performed by lexical processes whereas (3) is achieved by either lexical or syntactic processes depending on whether the nominalized verb has a subject or a specifier.

The chapter is structured as follows. Section 4.1 begins by discussing elements left unchanged from the previous nomanialized clauses library. The remaining sections then dive into extensions and changes to the prior analysis. Section 4.2 introduces the set of types inherited by nominalization lrts of different ANC types which account for the differences in dependent marking seen in the ANC typology proposed by Koptjevskaja-Tamm (1993). Section 4.3 describes the morphological behavior of action nominals and section 4.4 discusses the semantic representations available to ANCs when they take syntactic possessors. Section 4.5 covers additional properties of ANCs and is split into the following subsections: word order in ANCs (§4.5.1), optional arguments (§4.5.2), the single-possessor analysis (§4.5.3), adjectival versus adverbial modification (§4.5.4), and coordination (§4.5.5).

4.1 Commonalities to the Existing Nomainalized Clauses Library

One component left entirely unchanged from the existing analysis in the nominalized clauses library is the representation of nominal semantics (Howell et al., 2018). Thus, as discussed in chapter 3, the semantics of nominalization is captured through adding the relation **nominalized_rel** to the RELS list of the nominalized verb. This relation is shown in (1). As before, the LBL and *ref-ind* ARG0 of **nominalized_rel** are identified with the LTOP and INDEX of the nominalized verb. As mentioned in section 2.3.3, having a *ref-ind* INDEX value allows the nominalized verb to be modified by adjectives and requires that the INDEX be bound by a quantifier as with non-derived nouns. Certain argument positions (subjects of all verbs, complements of adpositions, and some verbal complements) are also restricted to categories with *ref-ind* INDEX values.

A *qeq* constraint is added to the HCONS list of the nominalized verb. This constraint links the ARG1 value of **nominalized_rel** to the LTOP of the original verb (the rule daughter) and provides a connection between the nominal semantics and the underlying verbal

semantics. The introduction of **nominalized_rel** then allows the action nominal to participate in certain syntactic processes that are generally reserved for nouns. Whenever I mention adding nominal semantics, I am referring to introducing this set of constraints to the nominalized verb.

(1)
$$\begin{bmatrix} \text{C-CONT} & \begin{bmatrix} \text{RELS} & \left\langle \begin{bmatrix} \text{PRED} & \text{nominalized\_rel} \\ \text{LBL} & \boxed{3} \\ \text{ARG0} & \boxed{2}\textit{ref-ind} \\ \text{ARG1} & \boxed{4} \end{bmatrix} \right\rangle \\ \text{HCONS} & \left\langle \begin{bmatrix} \textit{qeq} \\ \text{HARG} & \boxed{4} \\ \text{LARG} & \boxed{5} \end{bmatrix} \right\rangle \\ \text{HOOK} & \left\langle \begin{bmatrix} \text{INDEX} & \boxed{2} \\ \text{LTOP} & \boxed{3} \end{bmatrix} \right\rangle \end{bmatrix} \\ \text{DTR.SYNSEM.LOCAL.CONT.HOOK} & \begin{bmatrix} \text{LTOP} & \boxed{5} \end{bmatrix} \end{bmatrix}$$

Another element that is retained from the existing library is the fact that for all nominalization affixes in a language there is a corresponding nominalization lrt which acts to mark a verb as nominalized. I discuss the specifics of the constraints placed on this lrt across the different ANC types in the next section.

4.2 Normalization Lexical Rule Types

As in the previous library, I allow users to define different nominalization strategies and tie them to particular lrts. These lrts (nominalization lrts) take a verb as a daughter and produce an action nominal. The main difference is that instead of defining nominalization strategies based on the height in the tree when nominalization occurs, I instead base them around the different ANC types proposed by Koptjevskaja-Tamm (1993). Each ANC type results in an action nominal with a different valence features. By specifying that an lrt uses a nominalization strategy of a particular ANC type, it constrains the arguments that the resulting action nominal can take.

Nominalization lrts serve three primary functions: (1) changing the HEAD value from *verb* to *noun*, (2) setting the NMZ feature on the action nominal to [NMZ +], and (3) making any changes to the valence features of the action nominal relative to the underlying verb.

The HEAD change is necessary to ensure that action nominals have a nominal versus a verbal distribution. This is also why I decided to give action nominals a [HEAD *noun*] value and not a new HEAD value such as *gerund*, since introducing a new HEAD value would require making numerous edits to the rest of the Grammar Matrix to ensure that [HEAD *noun*] and [HEAD *gerund*] lexical items can appear in the same syntactic positions.

As in the previous nominalized clauses library (Howell et al., 2018), the NMZ feature is used to mark a lexical item as nominalized. Thus, all[1] lexical items initially have a [NMZ −] value, with a [NMZ +] value signifying that a lexical item has gone through a nominalization lrt.[2] Actions nominals are then distinguished from verbs through having a [HEAD *noun*] value and from non-derived nouns by being [NMZ +] .

These first two functions of nominalization lrts are the same across all nominalization strategies, however, for the third function, the changes made to an action nominal's valence features relative to the underlying verb vary based on the strategy's ANC type. For example, certain action nominals take subjects while others require specifiers. The nominalization lrt which produces the action nominal with a specifier must then have different valence constraints from the one that produces the action nominal with a subject.

To accommodate this variability in action nominal's valence features, I propose the hierarchy of types shown in figures 4.1[3] and 4.2.[4]

---

[1]The one exception to this is adpositions which instead take the same NMZ value as their complements.

[2]It would also be possible to leave the NMZ feature initially underspecified. However. as will be explained below, I often use the NMZ feature to control which rules can be used exclusively by nominalized verbs ([NMZ +]) versus another part of speech (nouns or verbs (both [NMZ −])). In particular, the analysis of possessive semantics in ANCs requires nominalized verbs to be distinguished from non-derived nouns. If nouns are underspecified for the NMZ feature, the constraints placed on these rules become ineffective.

[3]In all hierarchies, the final *-lex-rule* in all but *anc-lex-rule* is omitted for space.

[4]Only *trans-erg-poss* and *det-trans-erg-poss* are shown in figure 4.2 due to space constraints, but all the rules ending in *_st* have two subtypes following the same pattern with one rule having the same name minus the ending *_st* and the other having the same name plus the prefix *det-* *_st* is short for supertype.

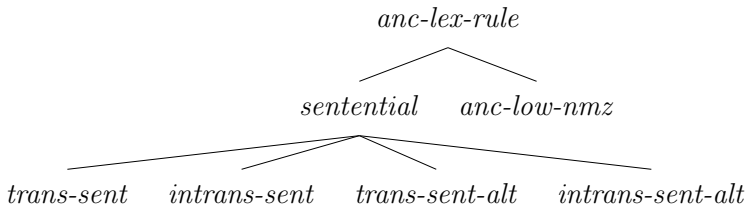


Figure 4.1: ANC type hierarchy

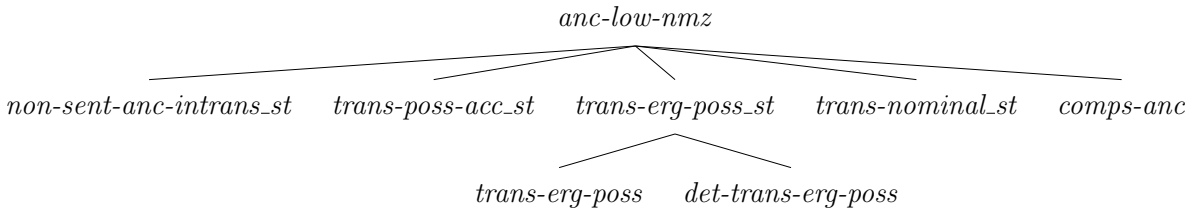


Figure 4.2: *anc-low-nmz-lex-rule* type hierarchy

No single nominalization lrt and thus action nominal makes use of all of these types. Instead a nominalization lrt inherits from a subset of these types determined by its nominalization strategy's ANC type. For example, and as I will explain in more detail later on, if a language has SENT action nominals (action nominals which share the same valence features as the underlying verb), all nominalization lrts which use a SENT nominalization strategy will inherit from the types *anc-lex-rule*, *sentential-lex-rule*, and depending on whether the rule acts on transitive or intransitive verbs, one of either *trans-sent-lex-rule* or *intrans-sent-lex-rule*. If this language has no other kinds of ANC, then only these specific types are added to the grammar.

Putting this in the context of the customization system, a user would define a lrt on the morphology page and mark it with a particular defined nominalization strategy. Based on the ANC type of the nominalization strategy, the customization system will include the relevant parts of this hierarchy in the customized grammar, and the lrt will inherit from

comps-anc has a distinct set of subtypes which are shown in figure 4.4

the corresponding leaf type in the hierarchy. The nominalization lrt will then have all the constraints specified by the types it inherits from as well as any constraints (such as case or other feature constraints on the action nominal's arguments) added by the user when creating the nominalization lrt.

The different types in the hierarchy allow for six different kinds of action nominal (SENT, ALT-SENT, POSS-ACC, ERG-POSS, NOMINAL, ALL-COMPS), each of which has a type for intransitive and transitive verbs. Four of these (SENT, POSS-ACC, ERG-POSS, NOMINAL) directly account for the main types proposed by Koptjevskaja-Tamm (1993). I added the remaining two (SENT-ALT, ALL-COMPS) for implementation purposes and I will motivate them below.

Walking through the type hiereachy, the supertype *anc-lex-rule* at the top contains constraints shared by all nominalization lrts. As mentioned above, all nominalization lrts both change the HEAD value of the rule daughter (the underlying verb) from *verb*[5] to *noun* and set the NMZ value of the mother (the action nominal) to +. This set of constraints is shown below in (2).[6]

(2)
$$\begin{bmatrix} \textit{anc-lex-rule} & \\ \text{SYNSEM.LOCAL.CAT.HEAD} & \begin{bmatrix} \textit{noun} & \\ \text{NMZ} & + \end{bmatrix} \\ \text{DTR.SYNSEM.LOCAL.CAT.HEAD} & \textit{verb} \end{bmatrix}$$

The first split in the tree in figure 4.1 models the difference between nominalized verbs which take a subject (and inherit from *sentential-lex-rule*) and those that take a specifier (and inherit from *anc-low-nmz-lex-rule*). Any action nominal which can take either a Poss argument or a determiner[7] needs a specifier (and no subject), while action nominals which

[5]The [HEAD *verb*] constraint on the DTR of *anc-lex-rule* is likely not necessary since verbs are already the only possible input to a nominalization position class defined on the Morphology page.

[6]Here and in all remaining feature structures, unless explicitly stated otherwise only constraints that differ between the daughter and mother are shown. All features not discussed are either passed up unchanged from mother to daughter, are left underspecified, or are present only to reduce ambiguity.

[7]I define the terms Poss argument and determiner with respect to the library in section 4.2.2

more closely preserve the valence features of the underlying verb such as SENT action nominals require subjects.

This distinction between action nominals with and without subjects raises a question as to where nominal semantics should be introduced. The current proposal strives to make the process of nominalization as lexical as possible. It is then desirable for the nominalization lrt to handle not only syntactic changes but semantic ones as well, including the introduction of **nominalized_rel**. However, this is not possible for nominalized verbs with subjects given the constraints on the head-subj rules within the Grammar Matrix. As mentioned in section 2.3.3, the head-subj rules in the Grammar Matrix require the head daughter to be of INDEX *event*. If an action nominal is to take a subject and serve as the head daughter in a head-subj rule, then it cannot have a *ref-ind* INDEX, which means that the nominalization lrt which created the subject-taking action nominal cannot introduce nominal semantics. On the other hand, action nominals with specifiers do not need to use the head-subj rules and can thus have nominal semantics introduced directly by the nominalization lrt.

Nominalized verbs which take subjects as opposed to specifiers are then further distinguished by whether or not nominal semantics can be introduced by the nominalization lrt. *sentential-lex-rule* and its subtypes will be discussed in the following subsection (§4.2.1) while *anc-low-nmz-lex-rule* and its subtypes will be discussed in the next (§4.2.2).

4.2.1 Nominalized verbs which take subjects

The constraints introduced by *sentential-lex-rule* are shown below in (3).

(3)
$$\begin{bmatrix} \textit{sentential-lex-rule} \\ \text{C-CONT} \begin{bmatrix} \text{RELS} & \langle\ \rangle \\ \text{HCONS} & \langle\ \rangle \end{bmatrix} \end{bmatrix}$$

The empty list values for both C-CONT.RELS and C-CONT.HCONS signifies that no semantic information is added by the lrt. This models the fact that action nominals with subjects do not receive nominal semantics from their corresponding nominalization lrts.

Returning to the hierarchy in figure 4.1, *sentential-lex-rule* has four subtypes (*trans-sent-lex-rule*, *intrans-sent-lex-rule*, *trans-sent-alt-lex-rule* and *intrans-sent-alt-lex-rule*). These subtypes represent two different kinds of action nominal (SENT and ALT-SENT) both of which have transitive (*trans*) and intransitive versions (*intrans*). To preserve space, all discussion and all MRSs shown will be about the transitive versions of these subtypes. The intransitive versions are identical to their transitive counterparts aside from having an empty COMPS list on the mother and daughter of the rule.[8]

SENT action nominals are those that mark the agent-like (A) and patient-like (P) arguments in the same manner as the underlying verb. Thus, as shown in (4), transitive SENT nominalization lrts inherit from *trans-sent-lex-rule* which passes up the rule daughter's (the verb's) SUBJ and COMPS value to the action nominal. This ensures that the subject and complement of the action nominal are marked in the same way as the original verb.

(4)
$$\begin{bmatrix} \textit{trans-sent-lex-rule} & \\ \text{SYNSEM.LOCAL.CAT.VAL} & \begin{bmatrix} \text{SUBJ} & \boxed{1} \\ \text{COMPS} & \boxed{2} \end{bmatrix} \\ \text{DTR.SYNSEM.LOCAL.CAT.VAL} & \begin{bmatrix} \text{SUBJ} & \boxed{1} \\ \text{COMPS} & \boxed{2} \end{bmatrix} \end{bmatrix}$$

As I mention above, in addition to the four main ANC types proposed by Koptjevskaja-Tamm (1993), I also add two additional ANC types for the purposes of implementation. I allow the option for an ALT-SENT action nominal where the P is marked in the same way as the P argument of the underlying verb, but the A can take alternative marking. If the A is marked as a Poss, one of specifier action nominals, to be discussed in the next section, should be used instead. This ANC type is motivated by languages that mark subjects in subordinate clauses in a manner distinct from finite main clauses. One example from English

[8]I include different types for SENT intransitive and transitive action nominals, despite both just copying up the COMPS list, to allow for nominalization strategies that work exclusively on either intransitive or transitive verbs. One example where this is necessary is Italian [ita], where transitive infinitives follow a SENT nominalization pattern, but intransitive infinitives can only mark the A with a possessive strategy (Koptjevskaja-Tamm, 2015). In this case, it is necessary to add a transitive SENT type, but not the corresponding intransitive SENT type.

[eng] is shown below in (4.2.1) where the ANC is in brackets.

(ALT-SENT)
He *was* *shocked* *by* *[him* *destroy-ing* *the* *evidence]*
He[NOM] was shocked by [him[ACC] destroy-NMLZ the evidence]
'He was shocked by him destroying the evidence.' [eng]

This example does not fit well into either the POSS-ACC, ERG-POSS, or NOMINAL types since none of the arguments of the ANC are marked as a Poss. The P (*the evidence*) is marked in the same way as in finite main clauses, and the A is marked with accusative case. While, in English [eng], nominative case is used to mark subjects in finite clauses, accusative case can be used to mark subjects in non-finite subordinate clauses such as in (5) where the non-finite subordinate clause is in brackets.

(5) *I* *was* *waiting* *for* *[him* *to* *destroy* *the* *evidence]*
I[NOM] was waiting for [him[ACC] to destroy the evidence]
'I was waiting for him to destroy the evidence.' [eng]

This suggests that the accusative A (*him*) in (4.2.1) should also be treated as a syntactic subject. Since ALT-SENT action nominals use a different subject marking compared to SENT action nominals, ALT-SENT nominalization lrts inherit from *trans-sent-alt-lex-rule* which is shown in (6).

(6)
$$\begin{bmatrix} \textit{trans-sent-alt-lex-rule} & \\ \text{SYNSEM.LOCAL.CAT.VAL} & \begin{bmatrix} \text{SUBJ} & \left\langle \left[\text{LOCAL.CONT.HOOK.INDEX} \quad \boxed{1} \right] \right\rangle \\ \text{COMPS} & \boxed{2} \end{bmatrix} \\ \text{DTR.SYNSEM.LOCAL.CAT.VAL} & \begin{bmatrix} \text{SUBJ} & \left\langle \left[\text{LOCAL.CONT.HOOK.INDEX} \quad \boxed{1} \right] \right\rangle \\ \text{COMPS} & \boxed{2} \end{bmatrix} \end{bmatrix}$$

As with *trans-sent-lex-rule*, *trans-sent-alt-lex-rule* also identifies the COMPS value across the mother and daughter of the rule, but only identifies the INDEX value of the SUBJ instead of the whole SUBJ value. This ensures that the syntactic subject of the ANC will receive the

same semantic interpretation as the underlying verb's subject (it still has the INDEX value of the underlying verb's subject), but also allows other processes in the grammar including other lexical rules to add additional constraints. For example, the SUBJ of the nominalized verb can take a different case value from the underlying verb. This is the same approach used by the previous library in both mid and low nominalization to allow for case-frame changes and marking by case-marking adpositions (Howell et al., 2018).

This completes the work done by the nominalization lrts for SENT and ALT-SENT action nominals. However, since both action nominals inherit from *sentential-lex-rule*, they still have *event* semantics. As in the previous library, I allow for two different semantic options for only SENT and ALT-SENT action nominals. The first is to leave **nominalized_rel** completely out of the semantics. In that case, nothing more needs to be done. The second option is to introduce nominal semantics at some point after the application of the nominalization lrt. This must occur after the action nominal takes a subject (forms a sentence (S) constituent) for the reasons I give above. This is the same scenario as for high nominalization in the prior nominalized clauses library where the introduction of nominal semantics occurred at the S level. Due to this, I use a slightly modified version of the high-nominalization phrase structure rule (*high-nominalized-clause-phrase*), shown in (7), to introduce nominal semantics to SENT and ALT-SENT action nominals.

(7)
$$\left[\begin{array}{l}\textit{high-nominalized-clause-phrase}\\ \text{SYNSEM.LOCAL.CAT}\left[\begin{array}{ll}\text{HEAD} & \boxed{6}\begin{bmatrix}\textit{noun}\\ \text{NMZ} & +\end{bmatrix}\\ \text{VAL} & \left[\text{SPR}\ \left\langle\left[\text{OPT}\ +\right]\right\rangle\right]\end{array}\right]\\ \text{C-CONT}\left[\begin{array}{ll}\text{RELS} & \left\langle\begin{bmatrix}\text{PRED} & \text{nominalized\_rel}\\ \text{LBL} & \boxed{3}\\ \text{ARG0} & \boxed{2}\textit{ref-ind}\\ \text{ARG1} & \boxed{4}\end{bmatrix}\right\rangle\\ \text{HCONS} & \left\langle\begin{bmatrix}\textit{qeq}\\ \text{HARG} & \boxed{4}\\ \text{LARG} & \boxed{5}\end{bmatrix}\right\rangle\\ \text{HOOK} & \begin{bmatrix}\text{INDEX} & \boxed{2}\\ \text{LTOP} & \boxed{3}\end{bmatrix}\end{array}\right]\\ \text{ARGS}\left\langle\left[\text{SYNSEM.LOCAL}\left[\begin{array}{ll}\text{CAT} & \left[\begin{array}{ll}\text{HEAD} & \boxed{6}\\ \text{VAL} & \begin{bmatrix}\text{SPR} & \langle\rangle\\ \text{COMPS} & \langle\rangle\\ \text{SUBJ} & \langle\rangle\\ \text{SPEC} & \langle\rangle\end{bmatrix}\end{array}\right]\\ \text{CONT.HOOK} & \begin{bmatrix}\text{INDEX} & \textit{event}\\ \text{LTOP} & \boxed{5}\end{bmatrix}\end{array}\right]\right]\right\rangle\end{array}\right]$$

This rule is almost unchanged from the original taking a lexical item of INDEX *event* with fully saturated valance features and adding the constraints associated with nominal semantics discussed in section 4.1. As in the original rule, setting the SPR on the mother to be [OPT +] forces the result of this phrase structure rule to go through the *bare-np-rule* which adds the quantifier required by all lexical items with a *ref-ind* INDEX value. The only minor change I make is that instead of having *high-nominalized-clause-phrase* change the HEAD value of the rule daughter from *verb* to *noun*, the head value is shared among mother and daughter. Thus, both the rule daughter and mother are head *noun*. This reflects the fact that in the new analysis, the change in HEAD value is always performed by a noniminalization lrt.

This analysis makes certain predictions about the behavior of action nominals which lack nominal semantics. For one, all non-*ref-ind* categories are blocked from certain argument positions in the Grammar Matrix including being the subject of a verb, the complement of an adposition, and certain verbal complements. It is then expected that action nominals analyzed as lacking nominal semantics should not appear in these environments. It is likewise predicted that action nominals with *event* semantics will never be modified by adjectives, since adjectives in the Grammar Matrix only modify elements that have a *ref-ind* INDEX. The decision to either include or exclude nominal semantics should then be made with these predictions in mind.

If a language has both SENT and ALT-SENT action nominals it is also necessary to be consistent with the choices of semantic representation. That is either both should include **nominalized_rel** in the semantics or both should exclude it. This is because various types outside of the nominalized clauses library which handle the different semantic options will conflict if both smantic options are added to the grammar simultaneously. This concludes the analysis of nominalization types which take subjects, with the next subsection tackling the remaining types which take specifiers.

4.2.2 Nominalized verbs which take specifiers

Moving on to action nominals which can take specifiers, specifiers in the Grammar Matrix can either be possessors (Poss arguments) or determiners (for languages which have them). Since there are action nominals which can take both kinds of specifier, I briefly define what I mean by both a Poss and a determiner before moving on to describe the types that specifier action nominals can inherit from.

As discussed in section 2.3.1, the adnominal possession library (Nielsen, 2018) allows for either a specifier or modifier analysis of possessors, but only the specifier analysis will be used in the proposed nominalized clauses library. Under the specifier analysis, an independent syntactic possessor (excluding possessor pronoun affixes on the possessum) is either the specifier of a noun or the non-head daughter of the binary rule *poss-phrase*. In the second

case, the head daughter (the possessum) of *poss-phrase* is constrained to have a non-empty specifier. Thus, if any lexical item is to have an independent syntactic possessor, it must have a non-empty specifier value regardless of whether the possessor is a specifier or the non-head daughter of *poss-phrase*. Whenever I use the term Poss, I am referring to either one of these two positions that an independent syntactic possessor can take or to a possessor pronoun affix on the possessum.

In addition to syntactic possessors, determiners (words such as *the* and *a* in English [eng]) also serve as specifiers to nouns. Whenever I use the term determiner, I am referring only to this set of nonpossessive specifiers. Unless further constraints are added, action nominals which can take specifiers can take either a Poss or a determiner.

Finally, with respect to both kinds of specifier, it is important to note that unlike verbal subjects or complements, a user can not specify the marking of a specifier for a nominalization lrt on the morphology page. All determiners must be defined as independent lexical items on the Lexicon page, and all Poss markings should be defined as different possessive specifier strategies on the Adnominal Possession page. I further allow users to specify which possessive strategies can be used in ANCs. The types to be discussed below do the work of allowing action nominals to take any specifier (both possessive and nonpossessive), but the analysis also assumes that possessive strategies and/or determiners have been independently defined. The complements of specifier action nominals can still be constrained by specifying constraints on the object of a nominalization lrt as with any other lrt.

The constraints on *anc-low-nmz-rule* inherited by all action nominals which take specifiers are given below in (8).

(8)
$$\begin{bmatrix} \textit{anc-low-nmz-lex-rule} & \\ \text{SYNSEM.LOCAL.CAT.VAL.SUBJ} & \langle\rangle \\ \text{C-CONT} & \begin{bmatrix} \text{RELS.LIST} & \left\langle \begin{bmatrix} \text{PRED} & \text{nominalized\_rel} \\ \text{LBL} & \boxed{3} \\ \text{ARG0} & \boxed{2}\textit{ref-ind} \\ \text{ARG1} & \boxed{4} \end{bmatrix} \right\rangle \\ \text{HCONS.LIST} & \left\langle \begin{bmatrix} \textit{qeq} \\ \text{HARG} \quad \boxed{4} \\ \text{LARG} \quad \boxed{5} \end{bmatrix} \right\rangle \\ \text{HOOK} & \begin{bmatrix} \text{XARG} & \boxed{1} \\ \text{INDEX} & \boxed{2} \\ \text{LTOP} & \boxed{3} \end{bmatrix} \end{bmatrix} \\ \text{DTR.SYNSEM.LOCAL.CONT.HOOK} & \begin{bmatrix} \text{XARG} & \boxed{1} \\ \text{LTOP} & \boxed{5} \end{bmatrix} \end{bmatrix}$$

In direct opposition to *sentential-lex-rule*, this type adds the constraints related to **nominalized_rel** and constrains the SUBJ value on the mother of the lexical rule to be empty. This type is inherited by all the remaining action nominal types meaning that all the action nominal types (POSS-ACC/ERG-POSS/NOMINAL/ALL-COMPS) to be discussed below must include **nominalized_rel** in their semantic representations. This creates a strict semantic divide between subtypes of *sentential-lex-rule* and *anc-low-nmz-lex-rule* as the latter has a *ref-ind* INDEX from the moment of nominalization and has no option to exclude nominal semantics.

Figure 4.2 shows an abbreviated version of the set of subtypes which inherit from *anc-low-nmz-lex-rule*. Ignoring *comps-anc* for now, all the other subtypes add constraints which model the behavior of the remaining three main types (POSS-ACC/ERG-POSS/NOMINAL) in Koptjevskaja-Tamm's ANC typology (Koptjevskaja-Tamm, 1993). I begin by discussing the types necessary to handle the intransitive versions of all three ANC types before then going over what is needed to cover the transitive versions of each.

Intransitive POSS-ACC/ERG-POSS/NOMINAL action nominals all mark the A as a Poss. This means that the same type (*non-sent-anc-intrans_supertype*) can be inherited by

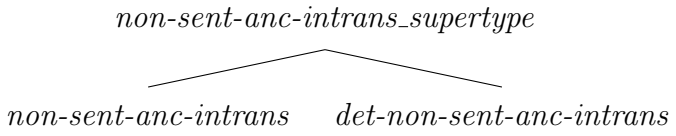


Figure 4.3: non-sent-anc-intrans type hierarchy

intransitive nominalization lrts of all three ANC types. An example of an intransitive ANC from English [eng] is shown in (9).

(9) *the actor 's sing-ing*
the actor POSS sing-NMLZ
'The (actor's) singing.' [eng]

In sentences like (9) where the Poss *actor's* is overt, it is interpreted as the agent of the action nominal *singing*. However, as (10) shows, it is also possible for the action nominal to take just a determiner.

(10) *the sing-ing*
the sing-NMLZ
'The singing.' [eng]

In this case, there is no overt agent. Since it is not correct to interpret *the* as the agent of *singing*, two distinct types are required: one where the action nominal's specifier is a Poss and is interpreted as the semantic agent (*non-sent-anc-intrans*) as in (9) and one where the action nominal's specifier is a determiner (*det-non-sent-anc-intrans*) as in (10). These two types share a common supertype (*non-sent-anc-intrans_supertype*) which results in the hierarchy shown in figure 4.3.

The types *non-sent-anc-intrans_supertype* and *non-sent-anc-intrans* are added to all grammars which have intransitive POSS-ACC, ERG-POSS, or NOMINAL action nominals. *det-non-sent-anc-intrans* is only added to a grammar if a language allows intransitive POSS-ACC, ERG-POSS, or NOMINAL action nominals to take determiners as with English [eng]

in (10). Thus, I analyze (9) and (10) as consisting of different action nominals resulting from two different nominalization lrts. Both nominalization lrts inherit from *anc-lex-rule*, *anc-low-nmz-lex-rule*, and *non-sent-anc-intrans_supertype*, but they differ in that the nominalization lrt in (9) inherits from *non-sent-anc-intrans* and the nominalization lrt in (10) inherits from *det-non-sent-anc-intrans*.

The supertype *non-sent-anc-intrans_supertype* (11) contains the constraints that are shared between the action nominals in (9) and (10). This mainly boils down to requiring an intransitive verb rule daughter (a daughter with an empty COMPS list).

(11)
$$\begin{bmatrix} \textit{non-sent-anc-intrans-lex-rule\_supertype} & \\ \text{SYNSEM.LOCAL.CAT.VAL.COMPS} & \boxed{2} \\ \text{DTR.SYNSEM.LOCAL.CAT.VAL.COMPS} & \boxed{2}\langle\ \rangle \end{bmatrix}$$

The constraints on *non-sent-anc-intrans-lex-rule* (12) identify the INDEX of the verb's SUBJ with the action nominal's SPR ensuring that the specifier is interpreted as the same semantic argument as the verb's subject would have been. I use a feature value from the adnominal possession library ([POSSESSOR possessive]) to require the specifier to be a syntactic possessor (Nielsen, 2018).

(12)
$$\begin{bmatrix} \textit{non-sent-anc-intrans-lex-rule} & \\ \text{SYNSEM.LOCAL.CAT.VAL.SPR} & \left\langle \begin{bmatrix} \text{LOCAL} & \begin{bmatrix} \text{CONT.HOOK.INDEX} & \boxed{1} \\ \text{CAT.HEAD.POSSESSOR} & \text{possessive} \end{bmatrix} \end{bmatrix} \right\rangle \\ \text{DTR.SYNSEM.LOCAL.CAT.VAL.SUBJ} & \left\langle \begin{bmatrix} \text{LOCAL.CONT.HOOK.INDEX} & \boxed{1} \end{bmatrix} \right\rangle \end{bmatrix}$$

The constraints on *det-non-sent-anc-intrans-lex-rule* are the exact inverse (13) as here I require the specifier to be [POSSESSOR nonpossessive]. I also use the OPT feature from the argument optionality library (Saleem, 2010) to mark the specifier as mandatory. This helps avoid ambiguity as the version of the action nominal with a determiner will never appear without its determiner in contrast to the option with a Poss which can drop its specifier argument.

(13) $\begin{bmatrix} \textit{det-non-sent-anc-intrans-lex-rule} & \\ \text{SYNSEM.LOCAL.CAT.VAL.SPR} & \left\langle \begin{bmatrix} \text{OPT} & - \\ \text{LOCAL.CAT.HEAD.POSSESSOR} & \text{nonpossessive} \end{bmatrix} \right\rangle \end{bmatrix}$

There are types analogous to *det-non-sent-anc-intrans-lex-rule* for transitive POSS-ACC/ERG-POSS/NOMINAL action nominals (*det-trans-poss-acc-lex-rule-rule*, *det-trans-erg-poss-lex-rule*, *det-trans-nominal-lex-rule*), however I will not mention them below as they all contain the exact same constraints as in (13) just with different supertypes. This is the reason why all the transitive POSS-ACC/ERG-POSS/NOMINAL types have supertypes ending in *_supertype* since they all have a Poss and determiner variant.

I now switch to the analysis of transitive POSS-ACC/ERG-POSS/NOMINAL action nominals. All of the types to be described below inherit from *anc-low-nmz-lex-rule*. Based on whether a language has POSS-ACC, ERG-POSS, or NOMINAL ANCs, I then add the necessary additional types to model that particular ANC type. In the following description, whenever I mention the specifier of a POSS-ACC, ERG-POSS, or NOMINAL ANC, I am referring only to a Poss (not a determiner).

Starting with the POSS-ACC type, as discussed in section 3.2.2, transitive POSS-ACC ANCs are characterized by marking the A as a Poss and the P in the same way as the underlying verb. This is modeled by the types shown in (14) and (15) which collectively copy up the entire COMPS value of the rule daughter and identify the indices of the daughter's SUBJ and the mother's SPR. This allows the action nominal's specifier to fill the same semantic role as the underlying verb's subject.

(14) $\begin{bmatrix} \textit{trans-poss-acc-lex-rule\_supertype} & \\ \text{SYNSEM.LOCAL.CAT.VAL} & \left[\text{COMPS} \quad \boxed{2}\right] \\ \text{DTR.SYNSEM.LOCAL.CAT.VAL} & \left[\text{COMPS} \quad \boxed{2}\right] \end{bmatrix}$

(15) $\begin{bmatrix} \textit{trans-poss-acc-lex-rule} & \\ \text{SYNSEM.LOCAL.CAT.VAL.SPR} & \left\langle \left[\text{LOCAL} \begin{bmatrix} \text{CONT.HOOK.INDEX} & \boxed{1} \\ \text{CAT.HEAD.POSSESSOR} & \text{possessive} \end{bmatrix}\right] \right\rangle \\ \text{DTR.SYNSEM.LOCAL.CAT.VAL.SUBJ} & \left\langle \left[\text{LOCAL.CONT.HOOK.INDEX} \quad \boxed{1}\right] \right\rangle \end{bmatrix}$

In ERG-POSS ANCs, the P is marked as a Poss, and the A as an oblique argument. I add the following two types *trans-erg-poss-lex-rule_supertype*[9] (16) and *trans-erg-poss-lex-rule* (17) which identify the INDEX of the daughter's SUBJ with the mother's COMPS INDEX and the INDEX of the daughter's COMPS with the mother's SPR INDEX. This ensures that the nominalized verb's complement will be interpreted in the same way as the underlying verb's subject, and its possessive specifier in the same way as the underlying verb's complement.

(16)
$$\begin{bmatrix} \textit{trans-erg-poss-lex-rule\_supertype} & \\ \text{SYNSEM.LOCAL.CAT.VAL} & \begin{bmatrix} \text{COMPS} & \left\langle \left[\text{LOCAL.CONT.HOOK.INDEX} \quad \boxed{1} \right] \right\rangle \end{bmatrix} \\ \text{DTR.SYNSEM.LOCAL.CAT.VAL} & \begin{bmatrix} \text{SUBJ} & \left\langle \left[\text{LOCAL.CONT.HOOK.INDEX} \quad \boxed{1} \right] \right\rangle \\ \text{COMPS} & \langle\ [\ ]\ \rangle \end{bmatrix} \end{bmatrix}$$

(17)
$$\begin{bmatrix} \textit{trans-erg-poss-lex-rule} & \\ \text{SYNSEM.LOCAL.CAT.VAL.SPR} & \left\langle \begin{bmatrix} \text{LOCAL} & \begin{bmatrix} \text{CONT.HOOK.INDEX} & \boxed{2} \\ \text{CAT.HEAD.POSSESSOR} & \text{possessive} \end{bmatrix} \end{bmatrix} \right\rangle \\ \text{DTR.SYNSEM.LOCAL.CAT.VAL.COMPS} & \left\langle \left[\text{LOCAL.CONT.HOOK.INDEX} \quad \boxed{2} \right] \right\rangle \end{bmatrix}$$

Since only indices are shared between the valence arguments of the verb daughter and the action nominal mother, the action nominal is free to mark its arguments in a way distinct from the verb. The specifier (P argument) can only take possessive marking, but the complement (A argument) can take additional case or adpositional constraints. This then allows the A to take an oblique marking.

In the NOMINAL nominalization type, the A is marked as a Poss while the P is either also marked as a Poss or else takes oblique marking. The situation where the P takes oblique marking is handled by the constraints in (18) and (19) as the INDEX values between the daughter's COMPS and the mother's COMPS as well as the daughter's SUBJ and the mother's SPR are identified.

[9] *trans-erg-poss-lex-rule_supertype* constrains the rule daughter to have a non-empty COMPS value to ensure that only transitive verbs can use this rule. This is necessary for the type *det-trans-erg-poss-lex-rule* which unlike *trans-erg-poss-lex-rule* does not add any additional constraints to the rule daughter.

(18)
$$\begin{bmatrix} \textit{trans-nominal-lex-rule\_supertype} & \\ \text{SYNSEM.LOCAL.CAT.VAL} & \left[\text{COMPS} \left\langle \left[\text{LOCAL.CONT.HOOK.INDEX} \ \boxed{2}\right]\right\rangle\right] \\ \text{DTR.SYNSEM.LOCAL.CAT.VAL} & \left[\text{COMPS} \left\langle \left[\text{LOCAL.CONT.HOOK.INDEX} \ \boxed{2}\right]\right\rangle\right] \end{bmatrix}$$

(19)
$$\begin{bmatrix} \textit{trans-nominal-lex-rule} & \\ \text{SYNSEM.LOCAL.CAT.VAL.SPR} & \left\langle \left[\text{LOCAL} \begin{bmatrix} \text{CONT.HOOK.INDEX} & \boxed{1} \\ \text{CAT.HEAD.POSSESSOR} & \text{possessive} \end{bmatrix}\right]\right\rangle \\ \text{DTR.SYNSEM.LOCAL.CAT.VAL.SUBJ} & \left\langle \left[\text{LOCAL.CONT.HOOK.INDEX} \ \boxed{1}\right]\right\rangle \end{bmatrix}$$

The action nominal's complement and specifier then receive the same semantic interpretation as would the underlying verb's complement and subject, respectively. Further constraints can be added to the nominalized verb's COMPS value to indicate the oblique marking, whereas the nominalized verb's SPR will be marked with whichever possessive strategy is defined for the grammar.

However, it is not possible, under my proposal, to have a nominalized verb where both arguments are marked as a Poss. This is because the adnominal possession library (Nielsen, 2018) only allows for a lexical item to take a single possessor. This causes a problem for examples such as (20) in English [eng] since both *the enemy's* and *of the city* could be analyzed as syntactic possessors.

(20) *the enemy 's destroy-tion of the city*
the enemy POSS destroy-NMLZ of the city
'The enemy's destruction of the city.' [eng]

Since only one argument can use the analysis from the adnominal possession library, I choose to always treat the A argument as the Poss. This means that if a user specifies that action nominals can use a particular possessive strategy, that strategy will be applied to only the A of a NOMINAL action nominal and not the P. I analyze the P as the complement of the action nominal. The marking of the P must then be specified on the Morphology page as it would be for any other verbal complement. For a sentence such as (20), this means that

the enemy's is marked by a defined possessive strategy, while *of the city* is treated simply as a prepositional complement with no possessive syntax or semantics. While it is possible to define a possessive strategy where the possessor is marked by the adposition *of*, such a strategy could not be used to mark the P argument in (20), since *the enemy's* is already serving as the syntactic possessor of the action nominal.

To aid the user, whenever a NOMINAL nominalization strategy is defined on the Nominalized Clauses page, I instruct the user to add any constraints to the P argument on the Morphology page, and that any defined possessive strategies will only serve to mark the A and not the P argument. My analysis does make the prediction that all marking on the P will either involve case or adpositional marking and thus can be handled on the Morphology page. In the case of English [eng], the P could be marked by requiring the action nominal's complement to appear with the preposition *of*. In chapter 5, I discuss the implementation of such prepositional phrase complements.

The above types capture the core properties of the four main ANC types detailed by Koptjevskaja-Tamm (1993). However, in addition to ALT-SENT described above, I also add one more type (ALL-COMPS) to cover English [eng] examples such as the one in (21).

(21) (ALL-COMPS)
He was shocked by [the dismiss-al of the secretary by the chairman]
He[NOM] was shocked by [the dismiss-NMLZ of the secretary by the chairman]
'He was shocked by the dismissal of the secretary by the chairman.' [eng]

The *secretary* acts as the P in (21) while *chairman* serves as the A. Both are marked by distinct prepositions (*of* and *by*). Koptjevskaja-Tamm (1993) classifies examples such as these as ERG-POSS, given that the preposition *of* can act as a possessive marker. The P is then marked as a Poss while the A takes oblique marking. However, the ANC in (21) can not be accommodated by the constraints in the transitive ERG-POSS type *trans-erg-poss-lex-rule*, as the nominalized verb *dismissal* has its specifier position taken by the determiner

the. This means that it cannot also simultaneously have a possessive specifier.

This might serve as evidence towards using the modifier analysis of possessors provided by the adnominal possession library as the modifier analysis allows possessors to appear with determiners (Nielsen, 2018). The phrase *of the secretary* would then be treated as a modifier of the action nominal. However, there does not seem to be sufficient motivation to treat such arguments as possessive modifiers, at least in English [eng]. For one, given that the preposition *of* is ambiguous, it is not fully clear that it is actually acting as a possessive marker in (21). Additionally, as just described, in the NOMINAL ANC type the P argument is always treated as a nonpossessive complement regardless of whether it potentially takes the same marking as a syntactic possessor. This is because the action nominal already has a specifier possessor (the A argument) and under the analysis in Nielsen (2018) a lexical item cannot take two possessors simultaneously. It would then be more consistent to always treat non-specifier Poss arguments as complements. For these reasons, a better alternative seems to be to treat *of the secretery* as a prepositional complement with no syntactic or semantic possessive marking.

To account for a sentence such as (21), I introduce the ALL-COMPS ANC type which allows for nominalized verbs that take two complements corresponding to the A and the P, as well as a nonpossessive specifier (a determiner), if present in the language. ALL-COMPS action nomimals are unique in being able to take a determiner, but not a Poss. As with the other types, I include separate types for intransitive and transitive ALL-COMPS ANCs. For transitive ANCs, I also allow the choice between whether the first complement of the action nominal is the A or the P argument. (21) provides an example where the P argument would be analyzed as the action nominal's first complement and the A argument as the second complement.

Across both intransitive and transitive varients, ALL-COMPS ANCs must forbid possessive specifiers. I model this through the constraints on the common supertype *comps-anc-lex-rule* shown in (22).[10]

[10]The nonpossessive constraint shown in (22) is only added if a language has defined possessive strategies.

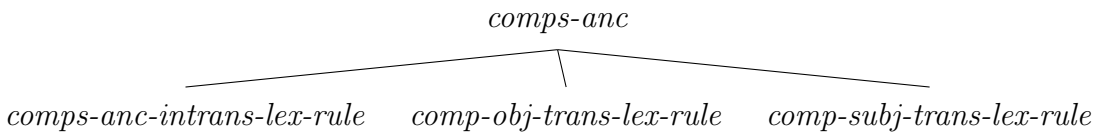


Figure 4.4: ALL-COMPS type hierarchy

(22) $$\begin{bmatrix} \textit{comps-anc-lex-rule} \\ \text{SYNSEM.LOCAL.CAT.VAL.SPR} \left\langle \left[\text{LOCAL.CAT.HEAD.POSSESSOR} \quad \text{nonpossessive} \right] \right\rangle \end{bmatrix}$$

The full hierarchy of ALL-COMPS types is shown in figure 4.4.

Intransitive ALL-COMPS action nominals take a single complement where that complement is interpreted as the A. The nominalization lrts which produce these action nominals inherit from *comps-anc-intrans-lex-rule* (23) which identifies the INDEX of the rule daughter's SUBJ with the INDEX value of the sole element on the mother's COMPS list. The only complement of the nominalized verb is therefore interpreted in the same way as the underlying verb's subject would have been.

(23) $$\begin{bmatrix} \textit{comps-anc-intrans-lex-rule} \\ \text{SYNSEM.LOCAL.CAT.VAL.COMPS} \left\langle \left[\text{LOCAL.CONT.HOOK.INDEX} \quad \boxed{1} \right] \right\rangle \\ \text{DTR.SYNSEM.LOCAL.CAT.VAL} \begin{bmatrix} \text{SUBJ} & \left\langle \left[\text{LOCAL.CONT.HOOK.INDEX} \quad \boxed{1} \right] \right\rangle \\ \text{COMPS} & \langle\rangle \end{bmatrix} \end{bmatrix}$$

Transitive ALL-COMPS action nominals whose first complement is the P inherit from *comps-obj-trans-lex-rule*. (24) identifies the INDEX values of the daughter's SUBJ with

---

If no possessive strategies are defined then the only specifier an action nominal can take is a determiner. Additionally, the nonpossessive constraint constraint on *comps-anc-lex-rule* is enough to prevent an ALL-COMPS action nominal from taking a possessive specifier, but it does not block it from combining with a possessor through a juxtaposition possessive strategy. A juxtaposition strategy combines two unmarked nouns with the binary rule *poss-phrase* as opposed to a head-spec rule. To ensure that this constraint also blocks *poss-phrase*, the specifier of the head daughter of *poss-phrase* is constrained to have a [POSSESSOR possessive] value.

the mother's second complement (the A) and the daughter's single COMPS item with the mother's first complement (the P).

(24)
$$\begin{bmatrix} \textit{comp-obj-trans-lex-rule} & \\ \text{SYNSEM.LOCAL.CAT.VAL} & \begin{bmatrix} \text{COMPS} & \left\langle \begin{bmatrix} \text{LOCAL.CONT.HOOK.INDEX} & \boxed{2} \end{bmatrix}, \right. \\ & \left. \begin{bmatrix} \text{LOCAL.CONT.HOOK.INDEX} & \boxed{1} \end{bmatrix} \right\rangle \end{bmatrix} \\ \text{DTR.SYNSEM.LOCAL.CAT.VAL} & \begin{bmatrix} \text{SUBJ} & \left\langle \begin{bmatrix} \text{LOCAL.CONT.HOOK.INDEX} & \boxed{1} \end{bmatrix} \right\rangle \\ \text{COMPS} & \left\langle \begin{bmatrix} \text{LOCAL.CONT.HOOK.INDEX} & \boxed{2} \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix}$$

Transitive ALL-COMPS action nominals whose first complement is the A inherit from *comps-subj-trans-lex-rule* which behaves in an equivalent fashion to *comps-obj-trans-lex-rule*, but with the INDEX of the first complement on the mother's COMPS list being identified with the daughter's SUBJ INDEX and the INDEX of the second complement with the daughter's single COMPS item (25).

(25)
$$\begin{bmatrix} \textit{comp-subj-trans-lex-rule} & \\ \text{SYNSEM.LOCAL.CAT.VAL} & \begin{bmatrix} \text{COMPS} & \left\langle \begin{bmatrix} \text{LOCAL.CONT.HOOK.INDEX} & \boxed{1} \end{bmatrix}, \right. \\ & \left. \begin{bmatrix} \text{LOCAL.CONT.HOOK.INDEX} & \boxed{2} \end{bmatrix} \right\rangle \end{bmatrix} \\ \text{DTR.SYNSEM.LOCAL.CAT.VAL} & \begin{bmatrix} \text{SUBJ} & \left\langle \begin{bmatrix} \text{LOCAL.CONT.HOOK.INDEX} & \boxed{1} \end{bmatrix} \right\rangle \\ \text{COMPS} & \left\langle \begin{bmatrix} \text{LOCAL.CONT.HOOK.INDEX} & \boxed{2} \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix}$$

These two lexical rules then allow transitive ALL-COMPS action nominals to either have the first complement interpreted in the same manner as the underlying verb's subject (*comp-subj-trans-lex-rule*) and the second in the same manner as the underlying verb's complement or vice-versa (*comps-obj-trans-lex-rule*). Since the P argument appears closer to the action nominal than the A argument in (21), the action nominal would inherit from *comp-obj-trans-lex-rule*.

The above covers the full set of types that are needed to cover the six different ANC types handled by the library. Each nominalization lrt in a language inherits from one of the

leaf subtypes depending both on the lrt's ANC type (SENT/ALT-SENT/POSS-ACC/ERG-POSS/NOMINAL) and whether the lrt applies to intransitive or transitive verbs. A grammar for a language is expected to only need a small subset of all the available types. As will discussed in chapter 6, among the languages that I worked with across both development and evaluation, most languages only required 2 or 4 of the 17 possible leaf subtypes, with English [eng] requiring the most at 8.

For all subtypes that only identify the INDEX values (and not the whole values) between the VAL features of the mother and daughter it is also possible to specify additional constraints on the action nominal's VAL features to capture differences in marking between the underlying verb and the nominalized verb. In total, nominalization lrts are then responsible for changing the HEAD value of the verb, setting the NMZ feature to +, potentially adding nominal semantics, and adjusting the valence features of the action nominal depending on its ANC type. One remaining task for nominalization lrts is ensuring the correct morphotactics of the nominalized verb, which will be discussed next.

4.3 Morphology

As discussed in section 3.2.5, across languages action nominals can take a mix of both typically verbal affixes such as tense as well as typically nominal ones such as case. I make the assumption that all verbal affixes must precede the nominalization affix and all nominal ones must follow it. Thus, while nominalized verbs can take both nominal and verbal affixes a nominalized verb cannot take a verbal affix after it has been nominalized or a nominal one prior to being nominalized. Nominalized here refers to going through a nominalization lrt.

Section 2.3.2 explains that, in the Grammar Matrix, morphological requirements are handled through the use of flag features. Nominalization lrts are unique in that no flags from the verbal rule daughter are passed up to the action nominal mother. Thus, if all verbs in a language are required to go through a tense position class (have a tense-flag), once a verb goes through a nominalization lrt, this flag is lost. In contrast, if a nominalization lrt serves as input to a mandatory nominal position class, the corresponding inflectional flag is

added directly to the nominalization lrt. Any verb that goes through a nominalization lrt then loses its verbal morphological requirements, but gains any necessary nominal ones. An example of a nominalization lrt (*nmz-lex-rule*) in a language with mandatory tense on verbs and mandatory case on nouns and action nominals is shown in (26).

(26) $\begin{bmatrix} \textit{nmz-lex-rule} & \\ \text{INFLECTED.CASE-FLAG} & - \\ \text{DTR.INFLECTED.TENSE-FLAG} & - \end{bmatrix}$

The rule daughter (the verb) in (26) has a mandatory tense-flag ([TENSE-FLAG −]) which is not copied over to the mother of the rule (the action nominal). The rule mother does however have a mandatory case-flag ([CASE-FLAG −]). Action nominals resulting from this nominalization lrt then have no obligation and in fact cannot[11] receive tense as is required by verbs, but do now need to go through a case position class to satisfy the action nominal's case requirement.

4.4 Possessive Semantics

After going through a nominalization lrt, as described in the previous two sections, the resulting action nominal is free to make use of any mechanisms in the grammar that it unifies with. In particular, action nominals with possessive specifiers can use the full specifier possessor analysis provided by the adnominal possession library. However, this also means that any action nominal which takes a Poss will contain possessive semantics. Since it may not always be desirable to interpret ANCs as possessive, I offer three different semantic options: (1) the semantics in ANCs is the same as in possessive noun phrases (noun-only), (2) the semantics in ANCs is the same as in finite clauses (verb-only), and (3) the semantics in ANCs has elements of both possessive noun phrases and finite clauses (both).

Noun-only preserves full possessive semantics with the argument marked as a Poss interpreted as the possessor of the nominalized verb and not as one of the arguments (agent or

[11]This is not directly tied to the loss of the tense-flag, but rather due to other factors such as the ordering of position classes or the fact that tense is a feature exclusive to *event* indices.

patient) of the underlying verb itself. The verb-only option leaves out possessive semantics entirely and interprets the argument marked as a Poss as solely a semantic argument of the verb. Finally, the both option offers the ability to interpret the argument marked as a Poss as both the possessor and one of the semantic arguments of the verb. Examples of all three types are shown below in (27), (28), and (29) for the ANC *the cat's sleeping*.

(27) Noun-only semantic representation

$$\begin{bmatrix} \textit{mrs} \\ \text{HOOK} & \begin{bmatrix} \text{LTOP} & \boxed{h6} \\ \text{INDEX} & \boxed{x2} \end{bmatrix} \\ \text{RELS} & \left\langle \begin{bmatrix} \text{PRED} & \textbf{\_exist\_q\_rel} \\ \text{LBL} & \boxed{h3} \\ \text{ARG0} & \boxed{x2} \\ \text{RSTR} & \boxed{h5} \\ \text{BODY} & \boxed{h4} \end{bmatrix}, \begin{bmatrix} \text{PRED} & \textbf{poss\_rel} \\ \text{LBL} & \boxed{h6} \\ \text{ARG0} & \boxed{e7} \\ \text{ARG1} & \boxed{x2} \\ \text{ARG2} & \boxed{x8} \end{bmatrix}, \begin{bmatrix} \text{PRED} & \textbf{\_def\_q\_rel} \\ \text{LBL} & \boxed{h9} \\ \text{ARG0} & \boxed{x8} \\ \text{RSTR} & \boxed{h11} \\ \text{BODY} & \boxed{h10} \end{bmatrix}, \right. \\ & \left. \begin{bmatrix} \text{PRED} & \textbf{\_cat\_n\_rel} \\ \text{LBL} & \boxed{h12} \\ \text{ARG0} & \boxed{x8} \end{bmatrix}, \begin{bmatrix} \text{PRED} & \textbf{\_sleep\_v\_rel} \\ \text{LBL} & \boxed{h13} \\ \text{ARG0} & \boxed{e15} \\ \text{ARG1} & \boxed{x14} \end{bmatrix}, \begin{bmatrix} \text{PRED} & \textbf{\_nominalized\_rel} \\ \text{LBL} & \boxed{h6} \\ \text{ARG0} & \boxed{x2} \\ \text{ARG1} & \boxed{h16} \end{bmatrix} \right\rangle \\ \text{HCONS} & \left\langle \begin{bmatrix} \textit{qeq} \\ \text{HARG} & \boxed{h5} \\ \text{LARG} & \boxed{h6} \end{bmatrix}, \begin{bmatrix} \textit{qeq} \\ \text{HARG} & \boxed{h11} \\ \text{LARG} & \boxed{h12} \end{bmatrix}, \begin{bmatrix} \textit{qeq} \\ \text{HARG} & \boxed{h16} \\ \text{LARG} & \boxed{h13} \end{bmatrix} \right\rangle \end{bmatrix}$$

(28) Verb-only semantic representation

$$\begin{bmatrix} mrs \\ \text{HOOK} & \begin{bmatrix} \text{LTOP} & \boxed{h13} \\ \text{INDEX} & \boxed{x2} \end{bmatrix} \\ \text{RELS} & \left\langle \begin{bmatrix} \text{PRED} & \textbf{\_exist\_q\_rel} \\ \text{LBL} & \boxed{h3} \\ \text{ARG0} & \boxed{x2} \\ \text{RSTR} & \boxed{h5} \\ \text{BODY} & \boxed{h4} \end{bmatrix}, \begin{bmatrix} \text{PRED} & \textbf{\_def\_q\_rel} \\ \text{LBL} & \boxed{h6} \\ \text{ARG0} & \boxed{x8} \\ \text{RSTR} & \boxed{h9} \\ \text{BODY} & \boxed{h7} \end{bmatrix}, \right. \\ & \left. \begin{bmatrix} \text{PRED} & \textbf{\_cat\_n\_rel} \\ \text{LBL} & \boxed{h10} \\ \text{ARG0} & \boxed{x8} \end{bmatrix}, \begin{bmatrix} \text{PRED} & \textbf{\_sleep\_v\_rel} \\ \text{LBL} & \boxed{h11} \\ \text{ARG0} & \boxed{e12} \\ \text{ARG1} & \boxed{x8} \end{bmatrix}, \begin{bmatrix} \text{PRED} & \textbf{\_nominalized\_rel} \\ \text{LBL} & \boxed{h13} \\ \text{ARG0} & \boxed{x2} \\ \text{ARG1} & \boxed{h14} \end{bmatrix} \right\rangle \\ \text{HCONS} & \left\langle \begin{bmatrix} qeq \\ \text{HARG} & \boxed{h5} \\ \text{LARG} & \boxed{h13} \end{bmatrix}, \begin{bmatrix} qeq \\ \text{HARG} & \boxed{h9} \\ \text{LARG} & \boxed{h10} \end{bmatrix}, \begin{bmatrix} qeq \\ \text{HARG} & \boxed{h14} \\ \text{LARG} & \boxed{h11} \end{bmatrix} \right\rangle \end{bmatrix}$$

(29) Both semantic representation

$$\begin{bmatrix} mrs \\ \text{HOOK} & \begin{bmatrix} \text{LTOP} & \boxed{h6} \\ \text{INDEX} & \boxed{x2} \end{bmatrix} \\ \text{RELS} & \left\langle \begin{bmatrix} \text{PRED} & \textbf{\_exist\_q\_rel} \\ \text{LBL} & \boxed{h3} \\ \text{ARG0} & \boxed{x2} \\ \text{RSTR} & \boxed{h5} \\ \text{BODY} & \boxed{h4} \end{bmatrix}, \begin{bmatrix} \text{PRED} & \textbf{poss\_rel} \\ \text{LBL} & \boxed{h6} \\ \text{ARG0} & \boxed{e7} \\ \text{ARG1} & \boxed{x2} \\ \text{ARG2} & \boxed{x8} \end{bmatrix}, \begin{bmatrix} \text{PRED} & \textbf{\_def\_q\_rel} \\ \text{LBL} & \boxed{h9} \\ \text{ARG0} & \boxed{x8} \\ \text{RSTR} & \boxed{h11} \\ \text{BODY} & \boxed{h10} \end{bmatrix}, \right. \\ & \left. \begin{bmatrix} \text{PRED} & \textbf{\_cat\_n\_rel} \\ \text{LBL} & \boxed{h12} \\ \text{ARG0} & \boxed{x8} \end{bmatrix}, \begin{bmatrix} \text{PRED} & \textbf{\_sleep\_v\_rel} \\ \text{LBL} & \boxed{h13} \\ \text{ARG0} & \boxed{e14} \\ \text{ARG1} & \boxed{x8} \end{bmatrix}, \begin{bmatrix} \text{PRED} & \textbf{\_nominalized\_rel} \\ \text{LBL} & \boxed{h6} \\ \text{ARG0} & \boxed{x2} \\ \text{ARG1} & \boxed{h15} \end{bmatrix} \right\rangle \\ \text{HCONS} & \left\langle \begin{bmatrix} qeq \\ \text{HARG} & \boxed{h5} \\ \text{LARG} & \boxed{h6} \end{bmatrix}, \begin{bmatrix} qeq \\ \text{HARG} & \boxed{h11} \\ \text{LARG} & \boxed{h12} \end{bmatrix}, \begin{bmatrix} qeq \\ \text{HARG} & \boxed{h15} \\ \text{LARG} & \boxed{h13} \end{bmatrix} \right\rangle \end{bmatrix}$$

All three MRSs have relations for *cat* and *sleep* as well as an associated quantifier for *cat* (**_def_q_rel**). Additionally, since the phrase is nominalized **nominalized_rel** is present along with an accompanying quantifier (**_exist_q_rel**). The three *qeq* relations are tied to the

two quantifiers and **nominalized_rel** and the overall INDEX and LTOP of the phrase are the same as ARG0 and LBL of **nominalized_rel** respectively. The differences between the three lie in the relationship between the *cat* and *sleep* relations as well as whether **poss_rel** is included and how it connects to the other relations present.

In (27) (noun-only), **poss_rel** is present with ARG1 (the possessum) co-indexed with ARG0 of **nominalized_rel** and ARG2 (the possessor) co-indexed with ARG0 of **_cat_n_rel**, but not with ARG1 of **_sleep_v_rel**. This indicates that *the cat* is the possessor of the nominalized verb, but not the agent of *sleeping*. In (28) (verb-only), **poss_rel** is absent from the RELS list while the ARG1 value of **_sleep_v_rel** is co-indexed with ARG0 of **_cat_n_rel** indicating that *the cat* is the one doing the *sleeping*. This semantic representation is the same as it would be for a finite clause such as *the cat sleeps* aside from the inclusion of **nominalized_rel** and the associated **_exist_q_rel**. Finally, (29) (both) includes **poss_rel** in the semantics and co-identifies ARG1 of **_cat_n_rel** with both ARG1 of **_sleep_v_rel** and ARG2 of **poss_rel**. ARG1 of **poss_rel** is identified with ARG0 of **nominalized_rel** as in the noun-only semantic variant. In this option, *the cat* is both the possessor of the nominalized verb and the agent of the verb. The purpose of offering these three semantic options is to allow the user analytic freedom with respect to the semantic representation.

To allow nominalized verbs to use the syntax associated with possession, but to potentially use a different semantic representation, I create subtypes of all the rules that introduce possessive semantics. One subtype constrains the argument acting as the possessum to be [NMZ +] and the other [NMZ −]. The [NMZ −] subtypes are used by non-derived nouns and retain the semantic analysis provided by the adnominal possession library, while the [NMZ +] subtypes are specific to nominalized verbs and can make use of one of the three semantic options discussed above based on how the grammar is customized. All syntactic constraints remain in existing supertypes, while semantic constraints are pulled out and added by the new [NMZ +] and [NMZ −] subtypes. Thus, non-derived nouns and action nominals display the same syntax when taking possessors, but potentially distinct semantics. The customization system adds these [NMZ +] and [NMZ −] subtypes to grammars based

on which possessive strategies can be used in ANCs. That is, the customization system will only create [NMZ +] and [NMZ −] subtypes of rules which can be used by both non-derived nouns and action nominals. If a possessive strategy is used only by non-derived nouns, [NMZ +] subtypes of its rules are not added.

Depending on the specific possessive strategy, possessive semantics can be introduced by several different rules in the adnominal possession library (Nielsen, 2018). In what follows, I only discuss one of these rules *poss-phrase*, as an example, but the explanation, aside from minor differences, also applies to the other rules that introduce possessive semantics.

poss-phrase is used whenever possession is marked through juxtaposition (two unmarked nouns). If ANCs can use a particular juxtaposition strategy then I add [NMZ −] (*noun-poss-phrase*) and [NMZ +] (*anc-poss-phrase*) subtypes of *poss-phrase* to the grammar for that strategy.[12] These two subtypes are responsible for adding all the semantic constraints for non-derived nouns and action nominals, respectively. The specific constraints introduced by *anc-poss-phrase* differ based on what ANC semantic option a user selects.

The majority of semantic constraints introduced by both subtypes are taken unchanged from the analysis in the adnominal possession library (Nielsen, 2018). Most significantly both *noun-poss-phrase* and *anc-poss-phrase* still introduce an **_exist_q_rel** and accompanying *qeq* constraint to act as a quantifier for the possessum (the nominalized verb for ANCs). The only difference among the three ANC semantic options involves the co-indexing behavior of the INDEX value of the argument acting as a Poss. The constraints for the three semantic options are given for *anc-poss-phrase* in (30) (noun-only), (31) (verb-only), and (32) (both). I only show constraints that differ across the three semantic variants.

(30) has the exact same constraints[13] as those provided by the adnominal possession

[12]In practice, the adnominal possession library itself creates subtypes of *poss-phrase* for each defined possessive strategy. Thus, the type *poss-phrase-1* is a subtype of *poss-phrase* which is used in the first user-defined juxtaposition strategy. My [NMZ +] and [NMZ −] subtypes are actually subtypes of these strategy specific types (*noun-poss-phrase-1* and *anc-poss-phrase-1*). However, I use *noun-poss-phrase* and *anc-poss-phrase* in the prose and in the example feature structures for the sake of clarity.

[13]The one exception to this is that for one of the rules that introduce possessive semantics (*poss-unary-phrase*), one constraint present in the [NMZ −] rule is removed for the [NMZ +] version. The placement of

library (with the exception of [NMZ +] on the HEAD-DTR), identifying the ARG2 (the possessor) of **poss_rel** with the INDEX of the argument serving as the Poss (the NON-HEAD-DTR). (31) (verb-only) lacks **poss_rel** and instead identifies the INDEX of the Poss argument (the NON-HEAD-DTR) with the INDEX of the nominalized verb's (the HEAD-DTR) specifier. Since the POSS-ACC, ERG-POSS, and NOMINAL nominalization lrts all identify the INDEX of the nominalized verb's specifier with either the SUBJ or COMPS INDEX of the underlying verb, this ensures the non-head daughter of *poss-phrase* will be interpreted as the relevant semantic argument of the verb. (32) (both) combines the constraints added in (30) and (31) so that the NON-HEAD-DTR of *poss-phrase* is interpreted as both the ARG2 (the possessor) of **poss_rel** and as a semantic argument of the underlying verb.

(30) Noun-only constraints

$$\left[\begin{array}{l} \textit{anc-poss-phrase} \\ \text{HEAD-DTR.SYNSEM.LOCAL.CAT.HEAD.NMZ} \quad + \\ \text{NON-HEAD-DTR.SYNSEM.LOCAL.CONT.HOOK.INDEX} \quad \boxed{5} \\ \text{C-CONT} \left[\text{RELS} \left\langle \left[\begin{array}{ll} \text{PRED} & \textbf{poss\_rel} \\ \text{LBL} & \boxed{2} \\ \text{ARG1} & \boxed{3} \\ \text{ARG2} & \boxed{5} \end{array}\right] \right\rangle\right] \end{array}\right]$$

(31) Verb-only constraints

$$\left[\begin{array}{l} \textit{anc-poss-phrase} \\ \text{HEAD-DTR.SYNSEM.LOCAL.CAT} \left[\begin{array}{ll} \text{HEAD.NMZ} & + \\ \text{VAL.SPR} & \left\langle \left[\text{LOCAL.CONT.HOOK.INDEX} \quad \boxed{5}\right]\right\rangle \end{array}\right] \\ \text{NON-HEAD-DTR} \left[\text{SYNSEM.LOCAL.CONT.HOOK.INDEX} \quad \boxed{5}\right] \end{array}\right]$$

---

this constraint on *poss-unary-phrase* causes problems for nominalized verbs and thus a similar constraint is contributed by a different type *head-compositional*. I will discuss the role of *head-compositional* in the analysis in the next subsection.

(32) Both constraints

$$\begin{bmatrix} \textit{anc-poss-phrase} \\ \text{HEAD-DTR.SYNSEM.LOCAL.CAT} \begin{bmatrix} \text{HEAD.NMZ} & + \\ \text{VAL.SPR} & \left\langle \left[\text{LOCAL.CONT.HOOK.INDEX} \quad \boxed{5} \right] \right\rangle \end{bmatrix} \\ \text{NON-HEAD-DTR} \left[\text{SYNSEM.LOCAL.CONT.HOOK.INDEX} \, \boxed{5} \right] \\ \text{C-CONT.RELS} \left\langle \begin{bmatrix} \text{PRED} & \textbf{poss\_rel} \\ \text{LBL} & \boxed{2} \\ \text{ARG1} & \boxed{3} \\ \text{ARG2} & \boxed{5} \end{bmatrix} \right\rangle \end{bmatrix}$$

4.4.1 Head-Spec-phrase

As mentioned in section 4.2.2, certain action nominals can take both a Poss (*the artist's singing*) and a determiner (*the singing*) with the semantics of the specifier necessarily differing between the two. Determiners in the Grammar Matrix are combined with their head nouns through a head-spec rule. There are also several possessive strategies where the possessor and possessum are likewise joined through a head-spec rule. As section 2.2 describes, the head-spec rules differ from other rules such as the head-subj or head-comp rules in treating the non-head daughter (and not the head daughter) as the semantic head. This accounts for some semantic differences between a noun taking a determiner and a verb taking a subject. However, under the verb-only/both semantic ANC options, the semantics between the action nominal and its specifier are more similar to that between a verb and its subject, than a noun and its determiner. Specifically, the action nominal's specifier is identified with one of the semantic arguments of the verb.

Due to this, in all cases where an action nominal can combine with a Poss using *head-spec-phrase*, I create a subtype (*anc-head-spec-phrase*) of *head-spec-phrase* which, in line with the head-subj and head-comp rules, treats the head daughter as the semantic head. This is done by having *anc-head-spec-phrase* inherit from one of the core Grammar Matrix types *head-compositional* which I discussed in section 2.2. *anc-head-spec-phrase* is restricted to be [NMZ +] so that only action nominals can use this version of *head-spec-phrase*. I also

require the non-head daughter to be [POSSESSOR possessive], which prevents determiners from participating in this rule. The constraints for *anc-head-spec-phrase* are shown in (33).

(33) $\begin{bmatrix} \textit{anc-head-spec-phrase} & \\ \text{SYNSEM.LOCAL.CAT.HEAD.NMZ} & + \\ \text{NON-HEAD-DTR.SYNSEM.LOCAL.CAT.HEAD.POSSESSOR} & \text{possessive} \end{bmatrix}$

To account for non-derived nouns and action nominals which take determiners I create two additional subtypes of *head-spec-phrase*. *noun-head-spec-phrase* (34) is identical to the original *head-spec-phrase* rule aside from being exclusive to non-derived nouns ([NMZ −]). *det-anc-spec-head-phrase* (35) is a version of the original *head-spec-phrase* rule that can be used by action nominals ([NMZ +]). I only add this type to a grammar if action nominals can take determiners. I constrain the NON-HEAD-DTR to be [POSSESSOR nonpossessive] to ensure that action nominals only use this variant of *head-spec-phrase* when taking determiners. The two [NMZ +] versions of *head-spec-phrase* then account for the differing semantics when action nominals take determiners versus syntactic possessors.

(34) $\begin{bmatrix} \textit{noun-head-spec-phrase} & \\ \text{SYNSEM.LOCAL.CAT.HEAD.NMZ} & - \end{bmatrix}$

(35) $\begin{bmatrix} \textit{det-anc-head-spec-phrase} & \\ \text{SYNSEM.LOCAL.CAT.HEAD.NMZ} & + \\ \text{NON-HEAD-DTR.SYNSEM.LOCAL.CAT.HEAD.POSSESSOR} & \text{nonpossessive} \end{bmatrix}$

4.5 Additional properties of ANCs

The above sections cover the majority of the proposed analysis, using a nominalization lrt to change the HEAD value of the verb, set the valence features of the action nominal and potentially introduce nominal semantics. Slight edits to the adnominal possession library also allow nominalized verbs to use syntactic constraints from the library while making use of a slightly modified semantic representation. The next several subsections cover additional elements that were added to try to better capture the behavior of ANCs including word order differences (§4.5.1), optional arguments (§4.5.2), single possessor argument marking patterns (§4.5.3), adverb/adjective modification (§4.5.4), and coordination (§4.5.5).

4.5.1 Word order

Section 3.2.8 discusses the fact that when head-dependent order differs across finite clauses and noun-phrases, ANCs tend to assimilate to the nominal over the verbal pattern. This can be seen by comparing a finite clause (36), a possessive noun phrase (37), and an ANC (38) in Hixkaryana [hix]. The A argument in an intransitive ANC (*tuna* 'water') precedes the action nominal in the same way a possessor (*kaykusu* 'dog') precedes the head noun in a possessive noun phrase, and in contrast to the subject of an intransitive verb which follows the verb.

(36) (Finite clause word order)

n-niki-yatxhe *kaykusu*
3-sleep-NPST.COLL dog
'Dogs sleep.' [hix] (Derbyshire, 1985)

(37) (possessive NP word order)

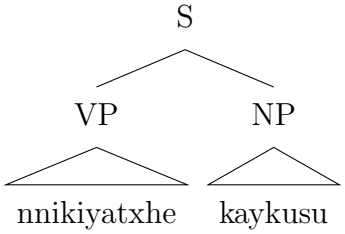
kaykusu *y-kahu-ri*
dog 3.OVERT-car-POSS
'The dog's car sleeps.' [hix] (Derbyshire, 1985)

(38) (ANC word order)

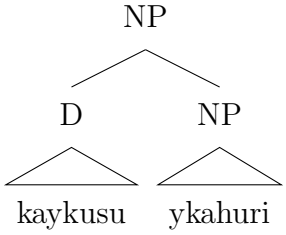
tuna *y-omoki-ni-ri*
water 3.OVERT-come-NMLZ-POSS
'Coming of water.' [hix] (Derbyshire, 1985: 38)

This can also be seen in the trees for these examples where the ANC (41) follows the same dependent-head word order as in possessive noun phrases (40) and not the head-dependent word order in finite clauses (39).

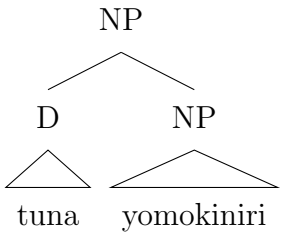
(39)



(40)



(41)



To accommodate these differences in word order, I allow action nominals which take a Poss argument[14] to have a single alternative word order. In the Grammar Matrix, the different word order options in finite clauses correspond to positions of the subject (S), the object (O), and the verb (V). In terms of semantic arguments, S maps to the A argument, and O to the P argument. In contrast, within ANCs, the three important positions are the Poss (POSS), the complement of the action nominal (COMP), and the action nominal (NMZ-verb). These syntactic positions map to different semantic arguments depending on whether the action nominal is of the POSS-ACC/NOMINAL or ERG-POSS types. I show the different syntactic positions that the Grammar Matrix main clause word orders map to in different action nominals in table 4.1.

When I say that action nominals with a Poss argument can take a different word order, I mean that they can take a different word order option from table 4.1 than what is true of main clause constituents. For example, Hixkaryana [hix] has OVS word order in finite clauses, but I would classify it as having SOV (COMP POSS NMZ-verb) word order in ANCs.

[14]For the SENT, SENT-ALT, and ALL-COMPS nominalization types, the word order must agree with the word order in a finite clause. Since nominalized verbs using ALL-COMPS do not have a subject, this means that all complements use the same head-complement order seen in finite clauses. For example, if a language has SVO word order, then both complements of a transitive ALL-COMPS action nominal will appear to the left of the verb (nominalized-verb comp1 comp2).

Table 4.1: Main clause word order options and corresponding ANC syntactic positions

Word order	POSS-ACC/NOMINAL	ERG-POSS
SOV	POSS, COMP, NMZ-verb	COMP, POSS, NMZ-verb
SVO	POSS, NMZ-verb, COMP	COMP, NMZ-verb, Poss
VSO	NMZ-verb, POSS, COMP	NMZ-verb, COMP, Poss
OSV	COMP, POSS, NMZ-verb	POSS, COMP, NMZ-verb
OVS	COMP, NMZ-verb, Poss	POSS, NMZ-verb, COMP
VOS	NMZ-verb, COMP, Poss	NMZ-verb, POSS, COMP
V-final	NMZ-verb-final	NMZ-verb-final
V-initial	NMZ-verb-initial	NMZ-verb-initial
Free	Free word order	Free word order
V2	Nominalized verb in second position, else free word order	Nominalized verb in second position, else free word order

Under my implementation, it is only possible to select a single alternative word order in ANCs which will be shared by all POSS-ACC/ERG-POSS/NOMINAL action nominals. The limitation of a single alternative word order is just an implementation decision to keep the analysis manageable. There also does not seem to be much motivation to allow different kinds of action nominals (POSS-ACC versus ERG-POSS) to use different word orders. However, as will be discussed in chapter 6, there are languages which seem to require ANCs to have multiple different fixed word orders. This is then a known limitation of the analysis which can be improved upon in the future.

In accounting for differences in ANC word order there are two primary elements the analysis needs to handle: (1) the order of daughters in the rules used to combine syntactic possessors and possessums,[15] and (2) the order of daughters in the head-comp rules. This first item is handled almost entirely by the adnominal possession library which has the relevant rules inherit from the types *head-initial* or *head-final* depending on whether the syntactic

[15]I make the assumption that ANCs can only use existing possessive strategies in a language, so that it is not possible for an ANC to use a possessor-possessum order not seen in non-derived noun phrases.

possessum or possessor appears first in the phrase.[16] *head-initial* and *head-final* are core Grammar Matrix types used to control the position of the head in a headed phrase. Since action nominals are able to use the [NMZ +] variants of rules provided by the adnominal possession library, they also abide by their order constraints.

The only time I do not use the word order analysis in the adnominal possession library involves free and V2 word order where I instead use the analysis in the word order library.[17] If a possessive strategy allows either possessor-first or possessum-first word order then the adnominal possession library adds both *head-initial* and *head-final* versions of the rule to the grammar. In contrast, I have the [NMZ +] versions of these rules inherit from subtypes of *head-initial* and *head-final* called *head-initial-head-nexus* and *head-final-head-nexus*. These nexus types come from the word order library and were designed to reduce unwanted ambiguity by forcing a certain order of attachment.[18]

I also achieve the second item (head-complement order) by relying on the analysis from the word order library. Where possible, POSS-ACC/ERG-POSS/NOMINAL action nominals make use of this analysis with no changes. However, there are several situations which require POSS-ACC/ERG-POSS/NOMINAL action nominals to use distinct head-comp rules relative to regular verbs and SENT/ALT-SENT/ALL-COMPS action nominals: (1) POSS-ACC/ERG-POSS/NOMINAL action nominals and regular verbs both use fixed word orders but they have different head-comp orders (SVO versus SOV), (2) POSS-ACC/ERG-POSS/NOMINAL action nominals use free word order and regular verbs use v2 word order or vice versa, (3) POSS-ACC/ERG-POSS/NOMINAL action nominals use free/v2 word order while regular verbs use a fixed word order or vice versa, and (4) POSS-ACC/ERG-

[16] One additional syntactic constraint is added to *anc-poss-phrase* compared to the original *poss-phrase* analysis. This involves copying up the COMPS value between mother and daughter whereas the original *poss-phrase* set the mother's COMPs value to be an empty list. This is necessary to allow for word orders where the possessor argument attaches to the action nominal prior to the complement.

[17] `https://github.com/delph-in/docs/wiki/MatrixDoc_WordOrder`

[18] Both free and V2 word order introduce these nexus rules, though the different word orders use different constraints to achieve the desired behavior.

POSS/NOMINAL action nominals use VSO or OSV word order.[19]

The first exception is the most straightforward since two head-comp rules are needed due to POSS-ACC/ERG-POSS/NOMINAL action nominals and regular verbs using opposite orders. Likewise, separate rules are needed if regular verbs use free word order and POSS-ACC/ERG-POSS/NOMINAL action nominals use v2 word order or vice versa as the free and v2 word order analyses add distinct constraints to the head-comp rules. Completely separating free/V2 word orders from the fixed words orders is slightly less clear, since free/V2 word orders add two head-comps rules to the grammar meaning that one of these could still be shared by both POSS-ACC/ERG-POSS/NOMINAL action nominals and regular verbs. However, as mentioned above, the free/V2 analysis has rules inherit from special nexus subtypes of *head-initial* and *head-final* whereas the head-comp rules of the fixed word orders directly inherit from *head-initial* and *head-final*. Due to this difference, I choose to keep these head-comp rules separate.

Finally, the need to use distinct head-comp rules for ANCs with VSO or OSV word order is the result of the analysis for the six fully fixed word orders. For each of the fixed word orders, the word order library constrains either the subject or the complement to always attach first. For example, for OVS, VOS, SOV, SVO word orders the head-daughter of the head-subj rule is constrained to be COMPS $\langle\ \rangle$. This ensures that in, for example, OVS word order, a verb can not take a subject until it has acquired all its complements. Likewise for VSO, OSV word orders, the head daughter of the head-comps rule is constrained to be SUBJ $\langle\ \rangle$, requiring the subject to be picked up first. In an analogous fashion, I require the rules which combine possessors and possessums to have a COMPS $\langle\ \rangle$ head-daughter (for OVS, VOS, SOV, SVO word order in POSS-ACC/ERG-POSS/NOMINAL ANCs), and the head-comp rules to have a SPR $\langle\ \rangle$ head-daughter (for VSO, OSV word order in POSS-ACC/ERG-POSS/NOMINAL ANCs). Since constraining the SPR value of the head daughter of the head-comp rule is

[19]It is unclear if all these different word order options are truly necessary, since for example, it seems unlikely that a language would have fixed word order in finite clauses, but free word order in ANCs, but all the logical possibilities are included for completeness.

not necessarily a constraint desirable for all lexical items[20], a separate rule is needed for POSS-ACC/ERG-POSS/NOMINAL ANCs with VSO or OSV word order to allow for this constraint to be applied exclusively to POSS-ACC/ERG-POSS/NOMINAL action nominals.

One complication in creating two different sets of head-comp rules, is that certain action nominals (SENT/ALT-SENT/ALL-COMPS) still need to use the existing head-comp rules while others (POSS-ACC/ERG-POSS/NOMINAL) should only use the new rules.[21] It is then not enough to simply restrict the appropriate head-comp rules to either be used by nominalized versus regular verbs. Additionally, it is also not sufficient to constrain different action nominals to use rules of a certain word order (*head-initial* versus *head-final*) as in some cases both sets of rules will have the same word order. For example, as mentioned above, if an ANC has VSO or OSV word order, an additional rule is always necessary even if finite clauses also have the same word order.

To get around this issue, whenever a language requires POSS-ACC/ERG-POSS/ NOMINAL action nominals to use a distinct set of head-comp rules, I add a new boolean HEAD feature ANC-WO to the grammar. All lexcal items aside from intransitive verbs and POSS-ACC/ERG-POSS/NOMINAL actions nominals are set to be [ANC-WO −]. On the other hand, POSS-ACC/ERG-POSS/NOMINAL actions nominals are set to be [ANC-WO +]. The existing head-comp rules constrain their head daughters to be [ANC-WO −], while the added head-comp rules used by POSS-ACC/ERG-POSS/NOMINAL action nominals constrain the head daughter to be [ANC-WO +]. This then ensures that the added head-comp rules can only be used by POSS-ACC/ERG-POSS/NOMINAL action nominals, which in turn cannot use the original [ANC-WO −] head-comp rules. This is likely not the best solution as ANC-WO does not seem to be a well-motivated feature, but it does provide a way of affording multiple word order options without a large spike in ambiguity.

---

[20]For example, in non-derived noun phrases, the specifier cannot attach first since the conventions of building the MRS require the specifier to be the last thing that attaches, as it provides the quantifier.

[21]I include this part of the analysis to allow for the possibility that a language includes both SENT/ALT-SENT/ALL-COMPS and POSS-ACC/ERG-POSS/NOMINAL actions with different word orders, but it is not clear if there are languages that allow this.

4.5.2 Optional arguments

This section deals with the fact that action nominals can often drop arguments that are mandatory for the underlying verb. Due to this increased flexibility around dropping arguments, I choose to make arguments optional in ANCs by default. Constraints can be added to make arguments mandatory, but unless these constraints are added all arguments are treated as optional.

Optional subjects and complements

In the Grammar Matrix, optional arguments are handled by the argument optionality library (Saleem, 2010). Unary rules (*decl-head-opt-subj-phrase* and *head-opt-comps-phrase*) are used to discharge subjects or complements without the need for an overt argument. These rules are only added to a grammar if optional arguments are allowed. A boolean feature (OPT) is used to mark an argument as mandatory ([OPT −]) or with the potential to be dropped ([OPT +] or leaving OPT underspecified).

To allow for optional arguments in ANCs without requiring optional arguments for all verbs, I add [NMZ +] subtypes (*anc-decl-head-opt-subj-phrase* and *anc-head-opt-comps-phrase*) of both *decl-head-opt-subj-phrase* and *head-opt-comps-phrase* to all grammars with defined nominalization strategies.[22] Both of these subtypes require their head daughter to be [NMZ +]. This then allows nominalized verbs to optionally drop both their subject and (if present) complement.

If regular verbs also allow optional arguments then I add additional [NMZ −] subtypes of both optional subject and complement rules (*regular-decl-head-opt-subj-phrase* and *regular-head-opt-comps-phrase*) to the grammar. This allows nominalized and regular verbs to display different behavior regarding optional arguments. If an argument (subject or complement) to an action nominal is indeed mandatory then the relevant argument in the nomi-

[22]It would also be possible to use the same rules (*decl-head-opt-subj-phrase* and *head-opt-comps-phrase*) for both nominalized verbs and regular verbs and just constrain any non-optional arguments of regular verbs to be [OPT −] (mandatory).

nalization lrt can be constrained to be [OPT −].

Optional and Mandatory Specifiers

The equivalent of *decl-head-opt-subj-phrase* and *head-opt-comps-phrase* for specifiers in the Grammar Matrix is the rule *basic-bare-np-phrase*. This is a unary phrase rule which discharges the SPR argument of its daughter without the need for the daughter to actually take an overt specifier. This rule is added to all grammars meaning that specifiers are already optional by default for all nominalized verbs that take specifiers.

As mentioned earlier, it is not possible to add constraints to specifiers on the Morphology page as with subjects or complements. To allow for mandatory specifiers, I instead have users specify the behavior of Poss arguments and determiners on the Nominalized Clauses page. I discuss the options I provide and the corresponding constraints below.

If the Poss argument of an action nominal is specified as being mandatory, I add the constraints shown in (42) to the nominalization lrt which produces the action nominal (*nmz-lex-rule*) .

(42)
$$\begin{bmatrix} \textit{nmz-lex-rule} \\ \text{SYNSEM.LOCAL.CAT.VAL.SPR} \quad \left\langle \begin{bmatrix} \text{OPT} & - \\ \text{LOCAL.CAT.HEAD.POSSESSOR} & \text{possessive} \end{bmatrix} \right\rangle \end{bmatrix}$$

The inclusion of [OPT −] on the SPR ensures the appearance of a specifier while the [POSSESSOR possessive] value forces the specifier to be a syntactic possessor and not a determiner.

With respect to determiners, I give action nominals the same range of possibilities as non-derived nouns in the Grammar Matrix. That is, a determiner can either be optional, impossible, or obligatory. For action nominals which can take Poss arguments, the determiner options are all relative to the Poss argument. Thus, even if a determiner is obligatory this just means that either the Poss argument must appear or a determiner must appear.

If a determiner is obligatory then I set the SPR of the action nominal to be [OPT −]. This forces the action nominal to take a specifier but does not demand that it be either

possessive or nonpossessive. If determiners are impossible then the constraints in (43) are added for POSS-ACC/ERG-POSS/NOMINAL nominalization lrts and the constraint in (44) for ALL-COMPS nominalization lrts.

(43) $\begin{bmatrix} \textit{nmz-lex-rule} \\ \text{SYNSEM.LOCAL.CAT.VAL.SPR} \quad \left\langle \left[\text{LOCAL.CAT.HEAD.POSSESSOR} \quad \text{possessive} \right] \right\rangle \end{bmatrix}$

(44) $\begin{bmatrix} \textit{nmz-lex-rule} \\ \text{SYNSEM.LOCAL.CAT.VAL.SPR.FIRST.OPT} \quad + \end{bmatrix}$

The constraint in (43) requires the action nominal's SPR to be [POSSESSOR possessive] which blocks nonpossessive specifiers including determiners.

The constraint in (43) does not work for ALL-COMPS action nominals since the ALL-COMPS nominalization lrt blocks possessive specifiers. I instead add the constraint [OPT +] to the action nominal's SPR.[23] In languages which have lexical items that cannot appear with specifiers [OPT −] is added to the NON-HEAD-DTR of the relevant head-spec rule. The fact that an action nominal has [OPT +] on its specifier (the non-head daughter) then prevents it from unifying with the head-spec rule. ALL-COMPS action nominals are then blocked from taking specifiers of any kind.

4.5.3 Single arguments

Another pattern described in section 3.2.7 for ANCs is the tendency for single arguments of action nominals to be marked as a Poss. Russian [rus] provides a prime example as when both arguments of a transitive action nominal appear in an ANC, the A takes instrumental case and the P take genitive case (45).

(45) (ERG-POSS)
ispoln-enij-e *sonat-y* *pianist-om*
Perform-NMLZ-NOM sonata-GEN pianist-INSTR
'The performance of the sonata by the pianist.' [rus] (Koptjevskaja-Tamm, 2013)

[23]The SPR of the ALL-COMPS action nominal cannot just be left empty, since the *bare-np-phrase* rule requires its daughter to have a non-empty SPR value. All nouns in the Grammar Matrix need to either take a specifier or go through the *bare-np-phrase* rule so that they can receive a quantifier.

However, whenever only a single argument appears, it is always marked with genitive case (the same case used to mark possessors) and is ambigious between an A (subjective) and a P (objective) reading (46).

(46) *ʧe-ni-e* *puʃkin-a*
read-NMLZ-NOM Pushkin[GEN]
'Pushkin's reading.' (subjective reading) or 'The reading of Pushkin.' (objective reading) [rus] (Comrie, 1976: 182)

This behavior is only relevant for transitive POSS-ACC/ERG-POSS/NOMINAL action nominals as these are the only action nominals that have both a Poss argument and the potential for an ambigous reading. To model this analysis I use two versions of the transitive normalization lrts for these three ANC types: one where the specifier is the P argument and another where the specifier is the A argument. This captures the inherit ambiguity present in (46). I refer to the version of the rule where the specifier is interpreted in the same way as in the case with two overt arguments as the primary rule, and the version where the specifier has the alternative interpretation as the secondary rule. Thus, in the Russian example, the rule leading to the objective reading is the primary rule and the rule leading to the subjective reading is the secondary rule.

The primary rule is the same as the rule used to create transitive POSS-ACC/ERG-POSS/NOMINAL action nominals more generally as described in section 4.2. The action nominal with an objective reading in (46) is then produced by the same nominalization lrt as the one used in (45) with the complement just dropped in (46). This explains why the Poss argument is the P argument in both. The only slight change made to this rule is to make the action nominal's specifier mandatory ([OPT −]). This ensures that the P argument, unlike the A argument, cannot be dropped. This blocks ungrammatical strings in Russian [rus] such as an action nominal appearing with just the instrumental-marked A argument in (45).

To capture the subjective reading in (46), I add an additional rule type which inherits from *anc-low-nmz-lex-rule*. This type is only added to the grammar if the user specifies that a nominalization strategy uses the single-possessor analysis. This rule identifies the INDEX

of the action nominal's SPR with the INDEX of the underlying verb's SUBJ and blocks the action nominal from taking a complement. Like with the primary rule, the rule daughter is required to be a transitive verb (non-empty COMPS value). The constraints for this new type (*trans-erg-poss-subj-only-lex-rule*) are shown in (47).

(47)
$$\left[\begin{array}{l} \textit{trans-erg-poss-subj-only-lex-rule} \\ \text{SYNSEM.LOCAL.CAT.VAL} \left[\begin{array}{l} \text{SPR} \left\langle \left[\text{LOCAL.CONT.HOOK.INDEX} \quad \boxed{1}\right]\right\rangle \\ \text{COMPS} \left\langle \left[\text{OPT} \quad +\right]\right\rangle \end{array}\right] \\ \text{DTR.SYNSEM.LOCAL.CAT.VAL} \left[\begin{array}{l} \text{SUBJ} \left\langle \left[\text{LOCAL.CONT.HOOK.INDEX} \quad \boxed{1}\right]\right\rangle \\ \text{COMPS} \left\langle \ [\ ]\ \right\rangle \end{array}\right] \end{array}\right]$$

Constraining the COMPS value of the rule mother to be [OPT +] conflicts with the [OPT −] constraint on the non-head daughter of the head-comp rules, effectively preventing the action nominal resulting from this rule from taking complements.[24] I block the complement since the subjective reading of the genitive argument in Russian [rus] is only possible when a single argument (the specifier under my analysis) is present. The combination of the primary and secondary rules then allows for action nominals taking specifiers with both semantic representations. I show just the RELS lists (using English [eng] predication names) that result from parsing the ANC in (46) with *trans-erg-poss-lex-rule* (the primary rule) and *trans-erg-poss-subj-only-lex-rule* (the secondary rule) in (48) and (49) respectively. These semantic representations use the verb-only semantic option for ANCs meaning that there is no possessive semantics included. The only difference between (48) and (49) is whether ARG2 or ARG1 of **_read_v_rel** is identified with ARG0 of **_pushkin_n_rel**. This is exactly the difference between interpreting *puʃkina* 'Pushkin' in (46) as the agent or patient of *ʧe-ni-e* 'read-NMZ' in (46).

[24]It would also be possible to just leave the COMPS value of the action nominal empty, and this might ultimately be the simpler analysis.

(48) Objective reading: *trans-erg-poss-lex-rule*

$$\begin{bmatrix} \textit{mrs} \\ \text{RELS} \left\langle \begin{bmatrix} \text{PRED} & \textbf{\_exist\_q\_rel} \\ \text{LBL} & \boxed{h3} \\ \text{ARG0} & \boxed{x2} \\ \text{RSTR} & \boxed{h5} \\ \text{BODY} & \boxed{h4} \end{bmatrix}, \begin{bmatrix} \text{PRED} & \textbf{\_exist\_q\_rel} \\ \text{LBL} & \boxed{h6} \\ \text{ARG0} & \boxed{x8} \\ \text{RSTR} & \boxed{h9} \\ \text{BODY} & \boxed{h7} \end{bmatrix}, \right. \\ \left. \begin{bmatrix} \text{PRED} & \textbf{\_pushkin\_n\_rel} \\ \text{LBL} & \boxed{h10} \\ \text{ARG0} & \boxed{x8} \end{bmatrix}, \begin{bmatrix} \text{PRED} & \textbf{\_read\_v\_rel} \\ \text{LBL} & \boxed{h11} \\ \text{ARG0} & \boxed{e12} \\ \text{ARG1} & \boxed{x15} \\ \text{ARG2} & \boxed{x8} \end{bmatrix}, \begin{bmatrix} \text{PRED} & \textbf{\_nominalized\_rel} \\ \text{LBL} & \boxed{h13} \\ \text{ARG0} & \boxed{x2} \\ \text{ARG1} & \boxed{h14} \end{bmatrix} \right\rangle \end{bmatrix}$$

(49) Subjective reading: *trans-erg-poss-subj-only-lex-rule*

$$\begin{bmatrix} \textit{mrs} \\ \text{RELS} \left\langle \begin{bmatrix} \text{PRED} & \textbf{\_exist\_q\_rel} \\ \text{LBL} & \boxed{h3} \\ \text{ARG0} & \boxed{x2} \\ \text{RSTR} & \boxed{h5} \\ \text{BODY} & \boxed{h4} \end{bmatrix}, \begin{bmatrix} \text{PRED} & \textbf{\_exist\_q\_rel} \\ \text{LBL} & \boxed{h6} \\ \text{ARG0} & \boxed{x8} \\ \text{RSTR} & \boxed{h9} \\ \text{BODY} & \boxed{h7} \end{bmatrix}, \right. \\ \left. \begin{bmatrix} \text{PRED} & \textbf{\_pushkin\_n\_rel} \\ \text{LBL} & \boxed{h10} \\ \text{ARG0} & \boxed{x8} \end{bmatrix}, \begin{bmatrix} \text{PRED} & \textbf{\_read\_v\_rel} \\ \text{LBL} & \boxed{h11} \\ \text{ARG0} & \boxed{e12} \\ \text{ARG1} & \boxed{x8} \\ \text{ARG2} & \boxed{x15} \end{bmatrix}, \begin{bmatrix} \text{PRED} & \textbf{\_nominalized\_rel} \\ \text{LBL} & \boxed{h13} \\ \text{ARG0} & \boxed{x2} \\ \text{ARG1} & \boxed{h14} \end{bmatrix} \right\rangle \end{bmatrix}$$

I have only discussed the secondary rule for ERG-POSS action nominals in this section, but an analogous secondary rule is added for POSS-ACC/NOMINAL action nominals to capture the objective reading (as opposed to the subjective reading standard for the primary rules). I include this rule for POSS-ACC/NOMINAL action nominals as a logical possibility, but I do not know of a specific language that requires it.

4.5.4 Adverb/adjective modification

I provide four different options with respect to whether action nominals can be modified by adjectives or adverbs. Either action nominals are modified (1) only by adjectives, (2)

only by adverbs, (3) both or, (4) neither. All four options make use of a newly introduced HEAD feature called ADV-MOD. This feature is added to all grammars that have both defined nominalization strategies and at least one defined adjective or adverb. ADV-MOD is discussed next along with other elements that hold true for all four variants. The next several subsections then explore what sets the four options apart.

Commonalities

The ADV-MOD feature marks whether a constituent can be modified by an adverb. It is a luk feature which means it can take one of three values: (1) [ADV-MOD +] indicating that the constituent can be modified by an adverb, (2) [ADV-MOD −] indicating that the constituent cannot be modified by an adverb, and (3) [ADV-MOD *na*] indicating that the feature is non-applicable. Non-derived nouns are specified as being [ADV-MOD −] and non-derived verbs as [ADV-MOD +]. *adverb-lex-item* the type which all adverbs in the Grammar Matrix inherit from is specified to modify a [ADV-MOD +] constituent (50). I also require *adverb-lex-item* to modify a constituent with HEAD value *+nv*. This means that adverbs can modify either [HEAD *noun*] or [HEAD *verb*] constituents as long as they have a [ADV-MOD +] value. The above constraints hold true across all four variants to be discussed next.

(50) $$\begin{bmatrix} \textit{adverb-lex-item} \\ \\ \text{SYNSEM.LOCAL.CAT.HEAD.MOD} & \left\langle \begin{bmatrix} \text{LOCAL.CAT.HEAD} & \begin{bmatrix} +nv \\ \text{ADV-MOD} + \end{bmatrix} \end{bmatrix} \right\rangle \end{bmatrix}$$

Adjective-only modification

In the Grammar Matrix, lexical items which inherit from adjective lexical types such as *attr-adj-lex* are specified to modify [HEAD *noun*] constituents. Thus, without any additional constraints, adjectives can modify action nominals. To block action nominals from being modified by adverbs, the mother of the nominalization lrt is set to [ADV-MOD −]. This essentially aligns the behavior of action nominals with that of non-derived nouns.

Adverb-only modification

If action nominals can instead only be modified by adverbs then the mother of the nominalization lrt is set to be [ADV-MOD +]. To prevent adjectives from modifying action nominals, adjective lexical types are constrained to only modify [ADV-MOD −] lexical items.

Both

If both adjectives and adverbs are able to modify action nominals then the ADV-MOD value of the mother of nominalization lrt is left underspecified. Being underspecified for ADV-MOD allows the action nominal to unify with the MOD values for both the adjective and the adverb lexical types. The ADV-MOD feature is still needed in these grammars to prevent adverbs from modifying non-derived nouns.

Neither

Finally, if action nominals can be modified by neither adjectives nor adverbs,[25] I set the mother of the nominalization lrt to be [ADV-MOD *na*] and constrain adjective lexical types to modify [ADV-MOD −] lexical items. This essentially states that this feature is irrelevant for action nominals. Since both adjectives and adverbs specify that their MOD value has a non *na* value for ADV-MOD, neither can unify with the action nominal. This then prevents both adjectival and adverbial modification.

4.5.5 Coordination

The main cross-linguistic observation regarding coordination and ANCs discussed in section 3.2.9 is that languages differ in whether ANCs can coordinate with non-derived nouns or not. The current analysis is able to capture this distinction in a limited manner by making use of the existing functionalities of the Coordination library (Drellishak and Bender,

[25]I do not know if this option is attested in any language, but I include it as a possibility in case the behavior of action nominals with respect to adjectival/adverbial modification in a language is unclear or irrelevant for a particular implementation.

2005).[26]

The Coordination library allows the coordination of nouns, noun phrases (NPs), verb phrases (VPs), and sentences. An agreement pattern can also be tied to each defined coordination strategy which specifies how any feature differences between the conjuncts should be resolved on the mother of the overall coordinated phrase. There are two kinds of agreement pattern: (1) a feature resolution pattern where rules can be defined to control which features appear on the mother of the coordinated phrase and (2) a distinguished conjunct pattern where the features on the coordinated phrase are the same as a conjunct in a predictable position. Currently the features handled by both agreement patterns are restricted to agreement features including person, number, gender, and any user-defined features. If no agreement pattern is tied to a coordination strategy, ANCs can make use of the coordination rules used by either nouns or NPs. ANCs can then coordinate with non-derived nouns aside from a few exceptions that will be outlined below. I add the NMZ feature to the set of features handled by the feature resolution pattern, so that it is possible to define rules such as that the NMZ value between the conjuncts and the mother should always be the same.[27] In this way it is possible to block action nominals from coordinating with non-derived nouns as they have distinct NMZ values.

Coordination behavior of different action nominals

As mentioned above, there are limitations to the current analysis of coordination involving ANCs. For one, only action nominals that have specifiers (POSS-ACC/ERG-POSS/ NOMINAL/ALL-COMPS) are able to go though the noun coordination rules. This is because the type responsible for noun coordination (*n-coord-phrase*) requires the mother of the phrase to have a non-empty specifier. Since in the Grammar Matrix, the VAL feature is identified across both daughters and the mother of a coordinated phrase, it is not possible

[26] `https://delph-in.github.io/docs/matrix/MatrixDoc_Coordination/`

[27] I expand the Coordination library slightly by adding the NMZ feature into the set of features handled by the feature resolution pattern but not the distinguished conjunct pattern, as there is no foreseeable need for this option.

for a coordinated phrase to have a specifier if one of the conjuncts does not. While it might also be ideal to create a version of the noun-coordination rule that requires the mother to have a subject instead of a specifier so that SENT and ALT-SENT action nominals could exploit it, this is left for future work. SENT and ALT-SENT ANCs can use the NP coordination rule as long as they include **nominalized_rel** in their semantics. Unless constrained otherwise, SENT and ALT-SENT ANCs without nominal semantics will still be able to coordinate with non-derived noun phrases and POSS-ACC/ERG-POSS/NOMINAL/ALL-COMPS ANCs, but not with other SENT or ALT-SENT ANCs lacking in nominalization semantics. One final note is that as defined above, all action nominals will be able to make use of any defined noun/NP coordination strategies (barring the exceptions just mentioned). It is then not possible to define a coordination strategy respected by POSS-ACC action nominals but not ERG-POSS action nominals. More data is needed to determine whether this is the correct analysis or something that requires further refinement.

4.6 Summary

The analysis I propose in this chapter covers the four main ANC types in Koptjevskaja-Tamm's typology of ANCs (Koptjevskaja-Tamm, 1993) as well as two additional types included for increased flexibility. The majority of the work of nominalization is handled through nominalization lrts with minor edits made to the adnominal possession library to provide a broader range of semantic options for ANCs with Poss arguments. Additional properties of ANCs handled include alternative word orders, optional arguments, single argument marking, adjectival/adverbial modification, and to a minor extent coordination. The above analysis captures the main phenomena described in chapter 3, while taking as much from existing Grammar Matrix analyses (adnominal possession, word order, coordination...) as possible reflecting the overall parasitic nature of action nominals. In the next chapter, I describe the questionnaire questions needed to collect necessary information about nominalization strategies in a language and then turn to how I implement the analyses presented in this chapter.

Chapter 5
IMPLEMENTATION

This chapter discusses the implementation of the analysis described in chapter 4 within the Grammar Matrix customization system (Bender et al., 2002, 2010). Section 5.1 goes over additions to the web questionnaire aimed at collecting all necessary information from the user regarding ANCs in a language. Section 5.2 then describes the python code which uses the answers taken from the questionnaire (stored in a choices file) to add the appropriate TDL (Type-Description Language; Copestake, 2002) to the files in the downloadable grammar. As mentioned in section 2.3, TDL is a formalism used to represent the types and constraints that have been visualized through AVMs in the previous chapters. Different choices files result in different constraints being added to the grammar files and thus distinct behavior. For the nominalized clauses library the majority of important information is outputted into the TDL file bearing the same name as the language being worked on (`language.tdl`). This file stores all language-specific analyses. All lexical entries, lexical rules, and phrase structure rules also need to be instantiated in specific files (`lexicon.tdl`, `irules.tdl`, `lrules.tdl`, `rules.tdl`). In what follows, unless stated otherwise, all TDL is assumed to be added to the `language.tdl` file. Section 5.3 concludes by walking through a concrete example involving a specific choices file and the resulting TDL.

It should be noted that during both implementation and evaluation I explore ANCs in four different positions: subject position, object position, clausal modifier position, and clausal complement position. Any action nominal with a *ref-ind* INDEX value can appear in subject or object position without any additional work required. For nominalized clausal modifiers and clausal complements, I use the code which allows ANCs to appear in these positions unchanged from the previous nominalized clauses library (Howell and Zamaraeva, 2018; Zamaraeva et al., 2019) and do not discuss that implementation here.

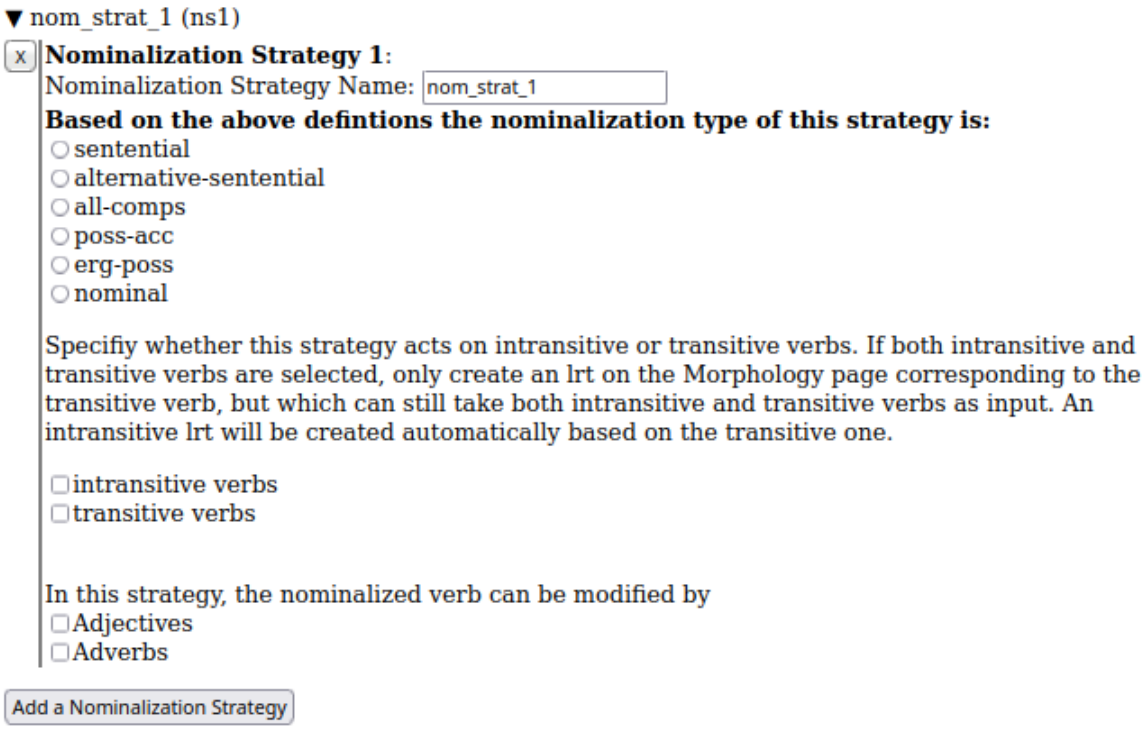


Figure 5.1: Example of a blank nominalization strategy

5.1 Questionnaire

As in the previous nominalized clauses library, the Nominalized Clauses subpage allows users to define one or more nominalization strategies (Howell et al., 2018). A blank example strategy is shown in figure 5.1.

Since not all questions are relevant for all nominalization types, more questions become visible based on a user's answers. By clicking on the 'Add a Nominaliization Strategy button' shown in figure 5.1 users can create as many nominalization strategies as needed for a language. As the library focuses on affixal nominalization these strategies are designed to be used in tandem with lexical rule types (lrts) defined on the Morphology page. Thus each nominalization lrt corresponds to a particular nominalization strategy. Section 5.1.1

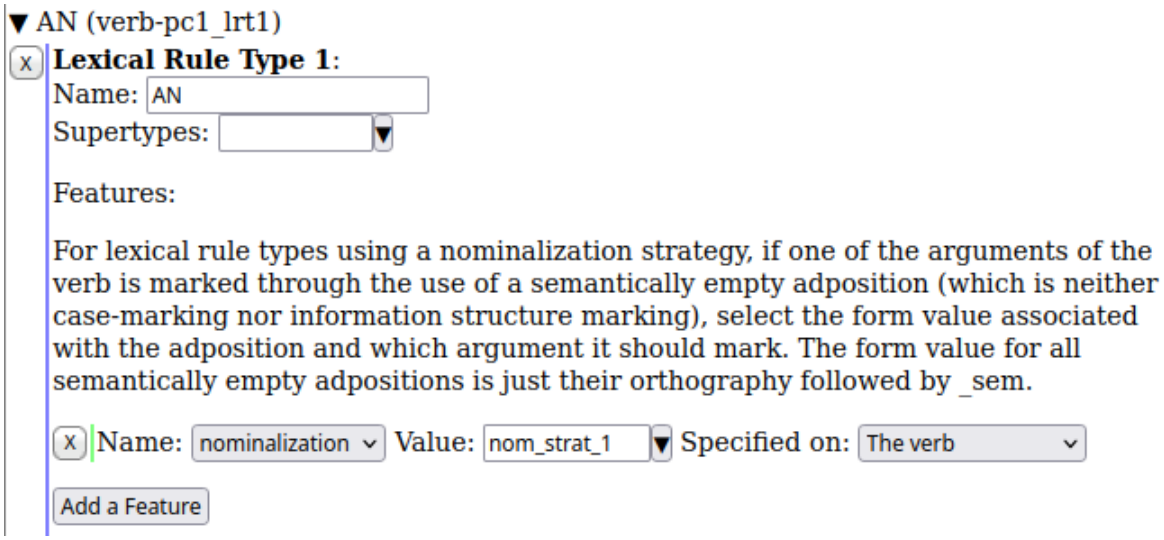


Figure 5.2: Example of a nominalization lrt marked with the nominalization feature

begins by describing the process of creating a nominalization lrt while section 5.1.2 returns to the Nominalized Clauses subpage itself and outlines the full set of questions asked for each nominalization strategy.

5.1.1 Nominalization lrts

In line with the approach in the original library, once a strategy is defined, a nominalization feature becomes available for verbal position classes (pcs) on the Morphology page. The values of this feature are the names of all defined nominalization strategies. A user can then create a verbal pc with an lrt within it that is marked by a specific nominalization strategy. Any verb that takes an affix associated with a nominalization lrt is nominalized and will exhibit the properties specified for action nominals belonging to the selected nominalization strategy. An example nominalization lrt is shown in figure 5.2.

It is also possible to specify additional constraints on the arguments of the nominalized verb. For example, the complement of an action nominal can be made mandatory by adding a new feature to the lrt (Name: OPT, Value: −, Specified on: the object). Likewise other

features such as case or number can also be selected for the arguments of the nominalized verb.

As mentioned in section 3.2, the two dependent marking strategies often seen in ANCs are case-marking and marking with a specific adpositon. Case-marking on a particular argument can be implemented as just described above. The use of specific adpositions, however, poses slightly more of a challenge. Given that adpositions in ANCs are used to mark arguments as one of the participants of the underlying verb, I make the assumption that these adpositions are semantically empty (contribute no semantic relation). The following implementation then focuses solely on marking arguments with semantically empty adpositions.

Semantically Empty Adpositions

The Lexicon subpage allows users to define two different kinds of semantically empty adposition: case-marking adpositions and information structural marking adpositions (Drellishak, 2009; Song, 2014). Information structural marking adpositions have additional complexities and are generally not selected by other lexical items to serve as arguments. Case-marking adpositions have specific case values and the argument of a lexical item can be marked with a case-marking adposition by requiring the argument to take that particular case value. However, there is no direct way to require an argument to be marked by an adposition of a particular orthographic form. As explained in section 2.1, in the Grammar Matrix, the FORM value can be used to select complementizers of a particular form, but all semantically empty adpositions are underspecified for this feature. As a result, even if a lexical item requires that its argument take a particular FORM value, there is nothing to block all the adpositions with underspecified FORM values from appearing in the same position. This makes it difficult to mark arguments with semantically empty adpositions in any language that does not have case.

To allow the option where semantically empty adpositions can be selected based on FORM value, whenever an adposition is defined that takes neither a case feature or an informational structural feature, I constrain the FORM value of all adpositions (both se-

mantically contentful and not). By default, I give all semantically contentful adpositions the value [FORM adpform]. Unlike with case-marking and informational structural adpositions, it is possible for users to specify the FORM value for semantically contenful adpositions. If there is a user-specified FORM value for the adposition, I keep that as the FORM value over the default adpform value. Semantically empty adpositions get a FORM value consisting of the orthography of the adposition followed by an underscore and a string representing the adposition category (orth_case, orth_infostr, orth_sem). This solves the problem of adpositions having underspecified FORM values and allows adpositions to be selected based on their orthographies.

The process of defining two semantically empty adpositions (which are neither case-marking nor informational structural marking) is shown in figure 5.3 for the English [eng] adpositions *of* and *by*. This is done in the same place (on the Lexicon subpage) where case-marking and information structural adpositions are defined with the adposition just not receiving any case or information structural features.

On the Morphology page, users can require an argument to take a defined semantically empty adposition by specifying the appropriate FORM value on the argument. The example shown in figure 5.4 provides an instance of a nominalization lrt which marks its first complement with the adp *of* ([FORM of_sem]) and its second complement with the adp *by* ([FORM by_sem]).

5.1.2 Nominalization Strategy Questions

Returning to the Nominialized Clauses subpage, listed down below are the full set of questions asked on this page. These questions are designed to get the relevant information regarding nominalization strategies in a language needed to output the correct TDL to the grammar. Certain questions are common to all nominalization strategies such as nominalization type ((1b) under per strategy), the valence of the verb the strategy acts on ((1c) under per strategy), and whether the resulting action nominal is modified by adjectives or adverbs ((1d) under per strategy). Other questions are only relevant to particular nominal-

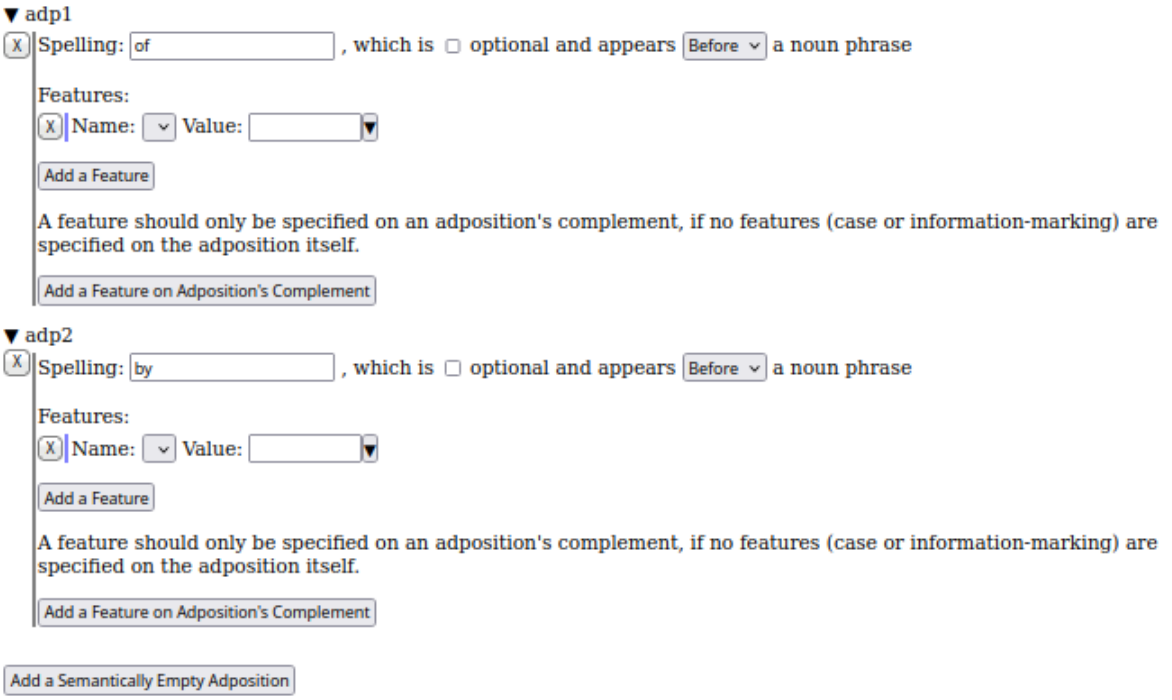


Figure 5.3: Example of entering adpositions on the Lexicon Page

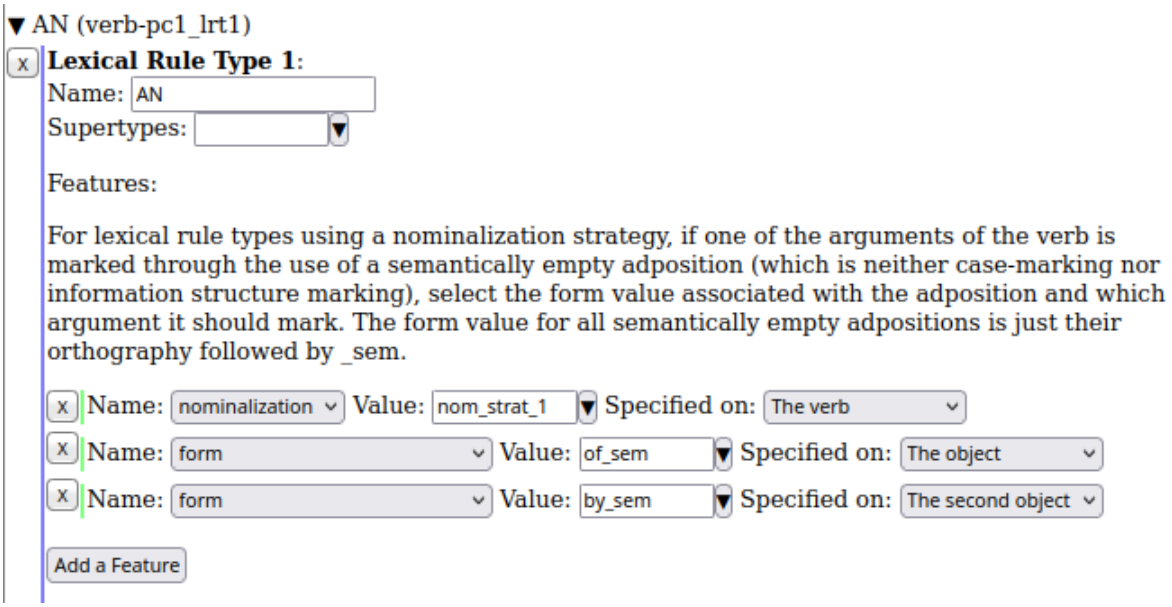


Figure 5.4: Example of marking the arguments of a nominalization lrt with semantically empty adpositions

ization types such as whether POSS-ACC/ERG-POSS/NOMINAL action nominals use an alternative ANC word order ((5a) under asked once) or whether SENT/ALT-SENT ANCs action nominals contain nominal semantics in their semantic representation ((2a) under per strategy). This second set of questions only become visible if the relevant nominalization type is selected for a strategy.

I also make a distinction between questions asked only once on the Nominalized Clauses page (ANC word order) which hold across all defined strategies and questions asked per strategy (ANC type) where the answer can differ for each strategy. A question such as the one concerning ANC word order is only asked once since the same word order must be used across all POSS-ACC/ERG-POSS/NOMINAL nominalization strategies. Thus, it is not possible for a user to define a NOMINAL nominalization strategy with free word order and an ERG-POSS nominalization strategy with a fixed word order within the same grammar.

In what follows, listed beside 'choices' for each question is also how the response is represented within the choices file. All choices that are made per strategy have the prefix ns_# indicating which strategy they apply to. ns_1 will be used when showcasing the choices below.

```
(1) Always Visible on the nominalized clause subpage:
    asked once:
        (a) (Checkbox) Across all strategies, are any arguments
                       of a nominalized verb mandatory?
            choices: mandatory-arg=on
    per strategy:
        (a) Provide a name as a way to refer to the nominalization strategy.
            choices: ns1_name=NAME
        (b) According to the typology in Koptjevskaja-Tamm 1993 what
           nominalization type does this strategy belong to?
           (sentential, alt-sent, all-comps, poss-acc, erg-poss, or nominal)
           choices: ns1_nmz_type={sentential, alt-sent, all-comps,
                                  poss-acc, erg-poss, nominal}
        (c) Specifiy whether this strategy acts on
                     (Checkbox) intransitive verbs
                     (Checkbox) transitive verbs
            choices: ns1_intrans=on
                     ns1_trans=on
```

```
        (d) In this strategy, the nominalized verb
                      can be modified by
                      (Checkbox) Adjectives
                      (Checkbox) Adverbs
            choices: ns1_adj=on
                     ns1_adv=on
(2) Visible if sentential or alternative-sentential is selected:
    per strategy:
        (a) Is the nominalization syntactic only
            or should it also be reflected in the semantics?
        choices: ns1_nmzRel={yes, no}
(3) Visible if all-comps, poss-acc, erg-poss, or nominal is selected:
    per strategy:
        (a) For nominalized verbs of this type, a determiner is:
            Obligatory, Optional, Impossible.
           choices:ns1_det={obl, opt, imp}
(4) Visible if all-comps is selected:
    per strategy:
        (a) For transitive action nominals, which argument serves
            as the first complement to the nominalized verb:
            the more agent-like or the more patient-like argument?
            choices: ns1_all_comps_arg_order={agent, patient}
(5) Visible if poss-acc, erg-poss, or nominal is selected:
    asked once:
        (a) do nominalized clauses have the same word order
            as regular verb phrases?
            If no, indicate which pattern best describes the word order
            in nominalized clauses.
        (sov, svo, vso, osv, ovs, vos, v-final, v-initial, free, or v2)
        choices: same-word-order={yes, no}
                 nmz-clause-word-order={sov, svo, vso, osv, ovs, vos,
                  v-final, v-initial, free, v2}
        (b) Please select which adnominal possessive strategies
            can be used when combining the nominalized verb
            with a possessor.
           choices: nmz_poss_strat1_name=POSS_STRAT
        (c) When the nominalized verb appears with a syntactic possessor
            the semantic represenation should be:
            both: Include both possessive and verbal semantics
            (the argument marked by the possessive strategy
             will be interpretated
             as both a verbal argument and a possessor)
            verb-only: Include only verbal semantics
            (the argument marked by the possessive strategy
```

```
        will be represented as only a verbal argument)
        noun-only: Include only  possessive semantics
        (the argument marked by the possessive strategy
        will be represented as just the possessor)
        choices: non_sent_sem={noun-only, verb-only, both}
per strategy:
    (a)  (Checkbox) Is the argument marked by the
                    possessive strategy mandatory?
         choices: ns1_mand-spr=on
    (b)  (Checkbox) Is a single argument always marked as a posssessor?
         ns2_single-arg=on
```

Figure 5.1 showcases the four questions asked for all nominalization strategies. Question (1a) under per strategy allows the user to enter a name for the strategy which can then be used as a nominalization feature value on the Morphology page. Skipping (1b) for now, question (1c) asks whether the affix associated with the nominalization strategy can attach to intransitive verbs, transitive verbs or both. This question is mandatory meaning one or both of the boxes must be selected. If both boxes are selected, I instruct the user to create a nominalization lrt which models the behavior of the transitive action nominal only on the Morphology page. As I will discuss in the next section, I make the intransitive equivalent automatically from the transitive version. Question (1d) concerns whether action nominals can be modified by adjectives, adverbs, both or neither. The neither option is selected if both the adjective and adverb boxes are left unchecked.

Question (1b) regarding the nominalization type is the most significant as it determines which further hidden questions are shown. Additionally, since the arguments corresponding to the A and P differ across nominalization type, the user is also given slightly different directions based on the strategy type. Specifically, across different nominalization types, the two syntactic arguments of the nominalized verb no longer necessarily map to subject and object position, which means the user needs instructions on which arguments are available to them when selecting features on the Morphology page. The syntactic positions (in terms of subject (subj), object (obj), second object (obj2) and Poss) of the A and P arguments of the different types of action nominals and the arguments the user can constrain on the

Morphology page are shown in table 5.1.

Table 5.1: Arguments of action nominals based on nominalization type

Type	**A**	**P**	**User can constrain**
SENT	subj	obj	neither
ALT-SENT	subj	obj	subj
ALL-COMPS	obj (intrans) obj/obj2 (trans)	obj/obj2	both
POSS-ACC	Poss	obj	obj
ERG-POSS	obj	Poss	obj
NOMINAL	Poss	obj	obj

The arguments listed in the third column correspond to the positions the user can specify constraints on such as case-marking or adpositional marking. Poss in this table refers to the syntactic possessor (either the specifier or the non-head daughter of *poss-phrase*). Subject refers to the external argument of the action nominal and object and second object correspond to the the first and second complements of the action nominal respectively. For transitive (trans) ALL-COMPS ANCs, the user decides whether the A or P is mapped to the first or second object (question (4a)). For intransitive (intrans) ANCs, the object is always the A. If the user specifies that a particular ALL-COMPS strategy acts on both intransitive and transitive verbs, then any feature constraints placed on the object of the transitive lrt will also be shared by the single complement of intransitive action nominals. I discuss the implementation of this fact in section 5.2.1. This allows for the fact that in English [eng], *of* can be used to mark both the A of an intransitive ANC (*the sleeping of the girl*) and the P of a transitive ANC (*the destroying of the evidence by the girl*).

Any marking affecting a Poss or determiner (a nonpossessive specifier) is handled on the Nominalized Clauses page (and the Adnominal Possession page for Poss arguments) and not the Morphology page. This is because the Morphology page only makes available verbal arguments such as subject and object. Beyond just the Poss, other arguments the user cannot constrain on the Morphology page are those whose marking is identical to that of the

corresponding dependent of the underlying verb. Thus, neither argument can be changed for the SENT nominalization type as both arguments will be marked in a manner equivalent to the underlying verb.

I also provide instructions to the user on how to handle any mandatory arguments. As described in section 4.5.2, all arguments of ANCs are treated as optional by default. However, either or both the P and A argument of an action nominal can be made mandatory if necessary. Question (1a) under asked once asks about mandatory arguments and is presented as a checkbox. Selecting the checkbox makes the OPT feature available on the Morphology page. Users can then mark any non-Poss arguments as mandatory by marking that argument as [OPT −]. Poss arguments can be marked as mandatory by selecting the checkbox associated with question (5a) under per strategy. Question (3a) allows users to specify that determiners are either obligatory (obl), optional (opt), or impossible (imp). As discussed in section 4.5.2, for POSS-ACC/ERG-POSS/NOMINAL action nominals, this decision is relative to the Poss argument.[1]

Possessive strategies used in ANCs (question (5b) under asked once) can be added in the manner shown in figure 5.5. The dropdown menu contains all the possessive strategies defined on the Adnominal Possession page for that language. Selecting a possessive strategy indicates that the Poss argument of all POSS-ACC/ERG-POS/NOMINAL action nominals can be marked with that strategy.

Question (5a) under asked once involving the word order in ANCs also requires clarification. In section 4.5.1, I map each of the Grammar Matrix word order options to a set of syntactic positions within ANCs depending on the ANC type. On the questionare, I ask users to choose a word order option based on the semantics of the arguments with S (subject) mapping to an A argument and O (object) mapping to a P argument. V maps to the action nominal. The exact wording of the question is given in figure 5.6. I use a function to

[1]For example, an obligatory determiner means that the action nominal must appear either with a Poss argument or with a determiner. An impossible determiner means that the action nominal can appear with a Poss, or with no specifier at all, but it cannot appear with a determiner.

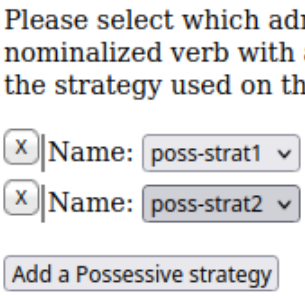


Figure 5.5: Example of entering possessive strategies on the Nominalized Clauses page

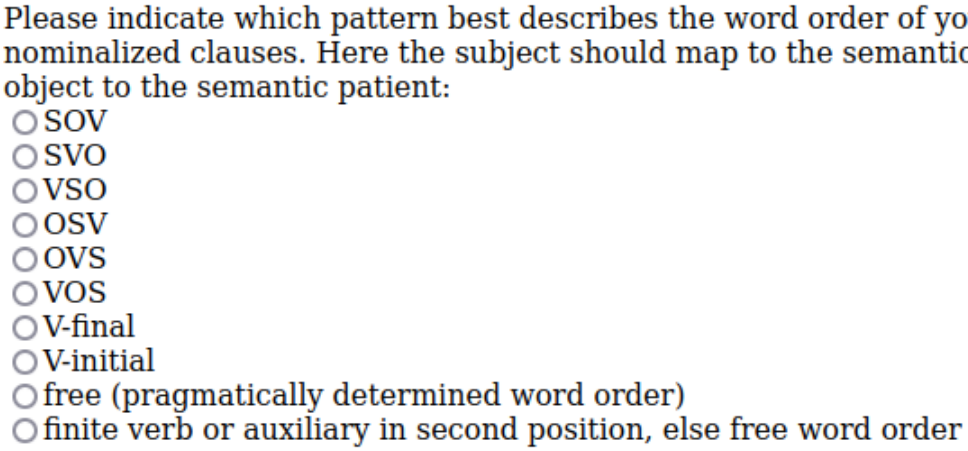


Figure 5.6: Instructions provided to the user on how to select the word order in ANCs

translate between the user's word order selection and the appropriate syntactic positions in a given ANC.

Question (4a) is specific to ALL-COMPS strategies and asks whether for transitive action nominals the action nominal's first complement should be the A or the P. This then determines which transitive ALL-COMPS type (*comp-subj-trans-lex-rule* or *comps-obj-trans-lex-rule* discussed in 4.2.2) to add to the grammar.

The remaining questions of the questionnaire are mostly straightforward. Selecting the checkbox linked with question (5b) under per strategy adds the lexical rules responsible for

having single arguments appear as possessors described in section 4.5.3 (the single-possessor analysis). Once the questionnaire has been filled out, the choices file is parsed to output a TDL analysis which reflects the choices selected. The python implementation of this process is the topic of the next section.

5.2 Python implementation

This section outlines the code that is responsible for outputting the TDL to match the requested analysis based on a user's choices. All code files to be discussed are stored in the GitHub Grammar Matrix repository under `/matrix/gmcs/linglib`. The `linglib` directory contains a number of files corresponding to different grammar libraries. The majority of the code responsible for implementing the analysis of ANCs is stored in a file called `nominalized_clauses.py`. Section 5.2.1 outlines the main functions used in this file. I also make updates to a number of additional library files (`morphotactics.py`, `adnominal_possession.py`, `word_order.py`, `argument_optionality.py`, `lexical_items.py`, and `coordination.py`) in order to extend the analysis in these files to account for ANCs. The remaining sections provide an overview of the main additions made to each of these files.

5.2.1 nominalized_clauses.py

The main purpose of the code in `nominalized_clauses.py` is to add all the TDL constraints relevant for nominalization lrts. For the analysis to work, the correct set of types from the set described in section 4.2 must be added to the grammar, with each nominalization lrt defined on the Morphology page by the user receiving the correct supertype based on the lrt's ANC type. Constraints pertaining to additional choices regarding a rule's nominalization strategy such as mandatory versus optional specifiers and whether action nominals can be modified by adjectives or adverbs are also handled by `nominalized_clauses.py`. Thus, all constraints concerning action nominals themselves are dealt with in this file. Aspects of the analysis which involve interacting with different libraries (morphology, adnominal pos-

session, word order...) are dealt with in those respective library files and will be discussed in later sections.

`nominalized_clauses.py` is broken up into several functions with the main function `customize_nmcs()` responsible for calling all the other functions and outputting the parts of the analysis discussed above. This function is a carryover from the previous library, although I modify which functions it calls.

In my implementation `customize_nmcs()` calls the following functions in the provided order. Beside each function, I put in parenthesis whether: (1) I added the function (new), (2) I modified a function in the previous library (modified), or (3) the function is used unchanged from the previous library (unchanged). In the modified functions, the core idea has stayed the same, but I have adjusted the code to be based around the different ANC types and not the three levels of nominalization (low, mid, high) present in the previous library.

add_anc_lexrules() (new):

`add_anc_lexrules()` adds additional lrts to the choices file that were not entered by the user. In other words, this is a function that handles fine details of the analyses that the user doesn't need to know about at the level of the questionnaire. All the user's choices including choices which define lrts are stored within data structures in a ChoicesFile object. `add_anc_lexrules()` adds additional choices corresponding to new lrts so that all remaining code functions as if the user had defined the lrts themselves. There are four cases where an extra lrt is necessary: (1) The user checks that the nominalization strategy acts on both intransitive and transitive verbs, (2) the user specifies that an action nominal uses an affix pronoun possessive strategy, (3) the user checks the box corresponding to the single-possessor analysis, and (4) the user defines at least one POSS-ACC/ERG-POSS/NOMINAL nominalization strategy and specifies that determiners are either optional or obligatory for action nominals created from that strategy.

In the first case, a user should only define a single lrt which contains all the feature constraints necessary for transitive action nominals. The choices corresponding to an intransitive version of the lrt will then be automatically added to the ChoicesFile object by

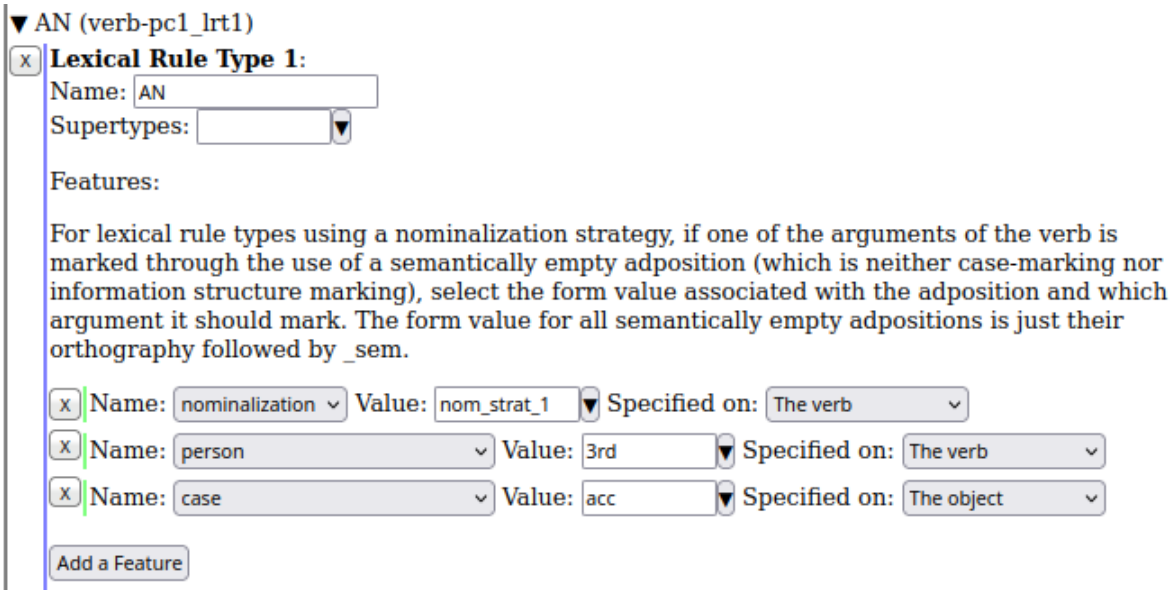


Figure 5.7: Example of creating a transitive nominalization lrt

copying all features from the transitive version aside from constraints on the object. The one exception is ALL-COMPS lrts, since intransitive ALL-COMPS action nominals still take a complement. Thus, for ALL-COMPS lrts, an intransitive lrt is created by copying all features aside from those on the second object. An example of a transitive nominalization lrt entered on the Morphology page is shown in figure 5.7.

I show the TDL that would be subsequently added to the grammar if the nominalization strategy used by this transitive lrt acted on both intransitive and transitive verbs in (1).

(1)
```
AN-lex-rule :=  trans-nominal-lex-rule &
  [ SYNSEM.LOCAL [ CONT.HOOK.INDEX.PNG.PER 3rd,
                   CAT [ VAL.COMPS < [ LOCAL.CAT.HEAD noun &
                                                 [ CASE acc ] > ] ] ].
AN_intrans-lex-rule := non-sent-anc-intrans-lex-rule &
  [ SYNSEM.LOCAL.CONT.HOOK.INDEX.PNG.PER 3rd ].
```

The TDL for the user-defined rule is stored within *AN-lex-rule*. The intransitive version is called *AN_intrans-lex-rule* and is distinguished from the transitive version by lacking any

constraints specified on the object (the first items on the COMPS list) and inheriting from an intransitive nominalization supertype (*non-sent-anc-intrans-lex-rule*) and not a transitive one (*trans-nominal-lex-rule*).[2]

In a similar vein, if the user defines a pronominal possessor affix[3] strategy on the Adnominal Possession page and then selects that ANCs can use this possessive strategy, `add_anc_lexrules()` will copy over all the choices corresponding to the pronominal possessor affix lrt (defined on the Morphology page) to a new lrt. The user-defined lrt is used by non-derived nouns while the automatically created lrt is used by nominalized verbs. Two distinct lrts are necessary as the one used by non-derived nouns is constrained to only take [NMZ −] daughters while the other takes [NMZ +] daughters as well as potentially different semantic constraints corresponding to the noun-only, verb-only, and both options. However, since all the choices regarding the lrt are copied, any information the user specified for the [NMZ −] version of the rule on the Morphology page is also shared by the [NMZ +] version.[4]

In the third case, the user has specified that a single argument of a transitive POSS-ACC/ERG-POSS/NOMINAL action nominal should always appear marked as a Poss. As discussed in section 4.5.3, this analysis is realized by adding additional lexical rules which forbid the complement of the verb from being realized. These additional lexical rules are automatically added whenever the user selects the box corresponding to the single-possessor analysis. Any information that the user specifies for the regular transitive version of the lrt is transferred over to the new lrt.

The final case is the situation described in section 4.2.2, when an action nomoinal can take both a Poss and a determiner. This requires the addition of an extra rule to capture the correct semantics when an action nominal combines with a determiner. On the Morphology

---

[2]The TDL itself is not added by `add_anc_lexrules()`. Instead the function just adds the choices corresponding to an intransitive lrt which will trigger later steps of the code to output this TDL.

[3]In this kind of possessive strategy, the possessor is not realized as a separate word but as an affix on the possessum.

[4]These two lrts don't currently have a shared supertype (beyond very generic types used by most lexical rules) which contains many of their shared constraints, but it would be ideal to add one in the future.

page, the user defines the features for the version of the rule where the action nominal takes a Poss. All the features of this lrt are copied to a new lrt which inherits from the appropriate *det-* prefixed type (such as *det-non-sent-anc-intrans*).

Since I add these extra lrt choices directly to the ChoicesFile object, the lexical rule instantiations for these rules are added to `irules.tdl` and `lrules.tdl` in `morphotactics.py` in the same manner as all the lrts defined by the user on the morphology page.

update_lexical_rules() (modified)

`update_lexical_rules()` is responsible both for adding the appropriate supertype to each nominalization lrt, and for adding any features to the lrt that will not be handled elsewhere. Each nominalization lrt must inherit from one of the leaf subtypes discussed in section 4.2. The correct supertype is determined by the lrt's nominalization type and the valence of the verb it acts on. For example, a transitive nominalization lrt which uses the ERG-POSS nominalization type would inherit from *trans-erg-poss-lex-rule*. As a reminder, no single grammar will contain all the different possible types described in section 4.2, but only a subset based on the user's choices. Two later functions (`add_anc_lex_supertype()` and `add_nmz_lexrules()`) are responsible for adding the TDL constraints for only the types needed in a particular grammar.

In addition, in a couple of cases update_lexical_rules() adds feature constraints to some lrts. The majority of feature constraints specified on the Morphology subpage are added to the appropriate lrt by the morphotactics library (in `morphotactics.py`). This is an existing functionality of the morphotactics library which works the same for all lrts. There are two exceptions which force `update_lexical_rules()` to add certain constraints to nominalization lrts instead. The first involves the HEAD value of any argument which takes a different case value compared to the corresponding argument of the underlying verb. The Morphology subpage of the questionnaire allows users to specify the CASE value of an argument, but not the HEAD value, which leaves the HEAD value underspecified. `update_lexical_rules()` calls a function which determines which lexical items in a given language can take case (either nouns, adpositions, or both). The appropriate HEAD value is then added to constrain the

case-taking argument. This approach to case-marking changes is taken from the original nominalized_clauses library (Howell et al., 2018).

Arguments marked by the FORM feature also have a similar issue where the HEAD value is underspecified. For semantically empty adpositions this means that not only can the actual semantically empty adposition appear as the argument, but so can any lexical item (nouns, verbs...) underspecified for FORM. To fix this, `update_lexical_rules()` adds the constraint [HEAD adp] to any argument of a nominalization lrt that is marked with a FORM value ending in the string '_sem'. This ending string uniquely identifies semantically empty adpositions and ensures that arguments marked by such FORM values can only be occupied by adpositions. An example of the TDL for an lrt which marks the subject of an ALT-SENT action nominal with the semantically empty adposition *adp1* is shown in (2).

(2)
```
AN-lex-rule := trans-sent-alt-lex-rule &
[ SYNSEM.LOCAL.CAT.VAL.SUBJ.FIRST.LOCAL.CAT.HEAD adp & [ FORM adp1_sem ] ].
```

`update_lexical_rules()` is responsible for setting the HEAD value to adp (adposition) and for specifying that the lrt should inherit from the nominalization supertype *trans-sent-alt-lex-rule*.

add_nmz_feature() (unchanged)[5]

`add_nmz_feature()` adds the TDL which introduces the NMZ feature to the grammar as a HEAD feature and sets the NMZ value for both *noun-lex* and *verb-lex* (the types inherited by the majority of nouns and verbs in a language respectively) to −.

add_anc_lex_supertype() (new)

`add_nmz_feature()` adds the TDL which describes the constraints introduced by *anc-lex-rule*, the type inherited by all nominalization lrts. This type is added separately from all the others as it is always added to a language containing nominalization strategies regardless of which specific nominalization types are present.

[5]Some code is removed from this function since it is no longer relevant for the current implementation, but all the code within the function is taken unchanged from the original library.

add_nmz_clause_phrases() (modified)

`add_nmz_clause_phrases()` adds the TDL for *high-nominalized-clause-phrase* if a language has either a SENT or an ALT-SENT nominalization strategy which includes **nominalized_rel** in its semantic representation. This phrase structure rule is necessary to introduce the nominal semantics for action nominals of these types. This function also adds *high-nominalized-clause-phrase* to `rules.tdl` which stores all phrase structure rule instantiations.

add_nmz_lexrules() (modified)

`add_nmz_lexrules()` adds the TDL for all the remaining nominalization subtypes needed for a given nominalization strategy. Thus, for a SENT nominalization strategy which acts on transitive verbs, this function would add the TDL corresponding to *sentential-lex-rule* and *trans-sent-lex-rule*.

add_nmz_mod_constraints() (new)

For languages with defined nominalization strategies and adjectives or adverbs, `add_nmz_mod_constraints()` introduces ADV-MOD as a new HEAD feature to the grammar and specifies the ADV-MOD value for *noun-lex*, *verb-lex*, and the MOD value of *adverb-lex-item*. `add_nmz_mod_constraints()` also calls a function which constrains the ADV-MOD value for all defined nominalization lrts based on the user's choice of whether action nominals can be modified by adjectives, adverbs, both, or neither.

add_anc_coord_constraints() (new)

The bulk of the work of coordination is handled by already existing code in the coordination library (in `coordination.py`), but a few necessary constraints are introduced by this function. `add_anc_coord_constraints()` constrains certain coordination rules to pass up the NMZ value. This is necessary since coordination rules are not headed and thus a HEAD feature such as NMZ is not passed up automatically. This function also constrains both daughters of *np-coord-phrase*, the type responsible for NP coordination, to have a *ref-ind* INDEX if a language has SENT/ALT-SENT nominalization strategies that include **nominalized_rel** in their semantics. This forces SENT/ALT-SENT ANCs to have obtained

nominal semantics before they can be coordinated.

handle_anc_spr_restrictions() (new)

`handle_anc_spr_restrictions()` uses the user's responses to questions (5a) under per strategy and (3a) to add the relevant constraints to specific nominalization lrts. This function adds the constraints which require a syntactic possessor to overtly appear as well as constraints that either force a determiner to appear whenever a syntactic possessor does not or make it impossible for a determiner to appear.

set_anc_wo_value() (new)

`set_anc_wo_value()` introduces the ANC-WO feature to the grammar whenever separate head-comp rules are necessary for POSS-ACC/ERG-POSS/NOMINAL action nominals. It also sets the value of ANC-WO to − for SENT/ALT-SENT/ALL-COMPS action nominals and to + for POSS-ACC/ERG-POSS/NOMINAL action nominals. The majority of remaining lexical items have their ANC-WO value set to − when needed in `lexical_items.py`, although `set_anc_wo_value()` is also responsible for setting the ANC-WO value for any lexical items not covered by `lexical_items.py`.[6]

Summary

For each nominalization lrt defined by a user, `nominalized_clauses.py` is responsible for adding the correct subset of types the lrt inherits from, introducing new features such as NMZ, ADV-MOD, and ANC-WO as necessary, and adding any constraints to the lrt specified on the Nominalized Clauses page such as mandatory versus optional specifiers. This covers all the constraints that are placed on the nominalization lrt itself. The remaining parts of the implementation concern how action nominals interact with other libraries. This includes implementing aspects of nominalization lrts and their resulting action nominals which differ from other lexical items such as their morphotactics and optional arguments as well as

[6]This follows the same general split seen with the NMZ feature in that NMZ and ANC-WO are both added to noun and verb types in `nominalized_clauses.py` and to most remaining lexical types in `lexical_items.py`. However, it might ultimately be better to condense all this similar code into one file.

choices a user can make about a nominalization strategy that relate to other libraries such as possessive strategies used in ANCs and ANC word order. The next sections then go over the additions made to other library files outside of `nominalized_clauses.py`.

5.2.2 morphotactics.py

Changes are made to this file to (1) prevent nominalization lrts from passing up verbal flags and (2) ensure required nominal flags are placed directly on nominalization pcs/lrts. To achieve (1), the step which copies the inflectional flags from the daughter to the mother of a rule is skipped for any pc which contains a nominalization lrt. Since each lrt inherits from a pc supertype, not copying up the flags on the supertype ensures none of the lrts in the pc pass up their inflectional flags. This does have the consequence that all lrts belonging to a pc containing at least one nominalization lrt will fail to pass up their verbal inflectional flags even if the lrt itself is not a nominalization lrt. Due to this, it is recommended to store nominalization lrts in separate pcs from all other lrts. As for (2), a function within `morphotactics.py` (`convert_obligatoriness_to_req()`) iterates through the inputs of all obligatory pcs and places a flag constraint on all the lexical types that lead into it (Goodman, 2013). This code is slightly changed, so that whenever the input is either a nominalization lrt or a pc that contains a nominalization lrt, the constraint is put not on a lexical type but on the nominalization pc supertype.

5.2.3 adnominal_possession.py

For each possessive strategy that can be used by ANCs, functions in this script create [NMZ +] and [NMZ −] subtypes of all necessary rules. This includes the rules that add possessive semantics as well as the head-spec rules when they are needed to combine a possessor with a possessum. The instantiations of any added phrase structure rules (such as *anc-spec-head-phrase*) are also added to `rules.tdl`. Additionally, functions in this script add the appropriate semantic constraints to the [NMZ +] rules that introduce possessive semantics based on the user's selection of noun-only, verb-only or both semantics.

5.2.4 word_order.py

Code is added to this file to handle alternative word orders in ANCs. Since I only allow non main clause word orders in nominalization types that use possessive strategies, the main function that deals with ANC word order (`customize_nmz_clause_word_order()`) is called by the adnominal possession library to add constraints to the [NMZ +] phrase structure (head-spec-phrase and poss-phrase) rules added by that file. These rules from the adnominal possession library are those that combine the action nominal with its Poss.

`customize_nmz_clause_word_order()` follows a similar logic to another function within `word_order.py` namely `customize_major_constituent_order()` which is responsible for handling word order in main clauses. If needed based on the POSS-ACC/ERG-POSS/ NOMINAL ANC word order, this function introduces [ANC-WO +] head-comp rules and sets any existing head-comp rules to [ANC-WO −], so that POSS-ACC/ERG-POSS/NOMINAL action nominals are handled separately from all other lexical items. This function also adds the rule instantiations to `rules.tdl` for any added head-comp rules. For all fully fixed (not free, v2, v-initial, or v-final) word orders, this function also adds all necessary constraints to either the head-comp rules or the head-spec/poss-phrase rules to force the correct order of attachment.[7]

5.2.5 argument_optionality.py

Since arguments of ANCs are treated as optional by default, any language that defines at least one nominalization strategy has [NMZ +] versions of both *head-opt-comp-phrase* and *decl-head-opt-subj-phrase* added to the grammar in this file. If a language also allows argument dropping in finite clauses, the regular versions of both the rules are also constrained to be [NMZ −]. `argument_optionality.py` is also responsible for adding the rule instantiations for all the added phrase structure rules to `rules.tdl`.

---

[7] A similar set of constraints is also added to *anc-head-opt-comp-phrase* and *anc-decl-head-opt-subj-phrase*, where the HEAD-DTR of *anc-head-opt-comp-phrase* is constrained to be SPR ⟨ ⟩ and SUBJ ⟨ ⟩ and the HEAD-DTR of *anc-decl-head-opt-subj-phrase* to be COMPS ⟨ ⟩.

5.2.6 lexical_items.py

`lexical_items.py` outputs the TDL for the majority of lexical types in the Grammar Matrix. Whenever a language has nominalization strategies, the NMZ value for all lexical types output by `lexical_items.py` is set to [NMZ −]. This ensures that no lexical type is underspecified for the NMZ value, and thus that only verbs that have been through a nominalization lrt can be [NMZ +].[8] In a similar manner, when needed, this file sets the ANC-WO value of all lexical items added to a grammar to [ANC-WO +]. This file also holds the code for adding [ADV-MOD −] to the MOD values of all adjective lexical types, which is only done when an action nominal can only be modified by adverbs or can be modified by neither adjectives nor adverbs.

5.2.7 coordination.py

I make a small update to this file to allow NMZ to serve as a feature (in addition to person, number, gender, and user-defined features) that can be selected for in coordination feature resolution agreement patterns.

Summary

This section has gone over all the code necessary to implement the analysis discussed in chapter 4. The bulk of the work is handled in `nominalized_clauses.py` which is responsible for adding all necessary constraints to nominalization lrts and adding any features to the grammar that are specific to ANCs. The exact types and constraints outputted by `nominalized_clauses.py` depend on which choices the user has selected for nominalization strategies in their language. Code added to other files is primarily only called as needed if the user selects a choice that requires the use of an analysis from another library such as the adnominal possession or word order libraries. Together the code in these files then adds all the necessary TDL based on a user's choices to `language.tdl`, `irules.tdl`, `lrules.tdl`,

[8] Adpositions can be [NMZ +] without going through a nominalization rule, but only if they take a [NMZ +] complement.

and `rules.tdl`.

5.3 Example

To demonstrate the functioning of both the questionnaire and the python code, the following steps through the TDL output[9] produced by an example choices file for Hixkaryana [hix].[10] An ANC from hixkaryana is shown in (3).

(3) (ERG-POSS)

k-rata-no	*uro*	*[biryekomo*	*wya*	*katu*	*y-tayma-ni-ri*	*ke]*
1-wept-IMM.PST	1	[boy	by	cat	3.OVERT-push-NMLZ-POSS	because]

'I wept because the boy pushed the cat.' [hix] constructed example based on (Derbyshire, 1985)

The suffix *-ni* can be used to nominalize both transitive and intransitive verbs in Hixkaryana [hix]. The P *katu* 'cat' is marked in the same way as a possessor with the action nominal *tayma-ni* 'push-NMLZ' taking a possessive suffix *-ri* and a prefix *y-* agreeing in person with the preceding noun *katu* 'cat'. The A is marked with the postposition *wya*. In intransitive action nominals, the A is also marked as Poss. In terms of their modification properties, Hixkaryana [hix] action nominals can be modified by only adverbs.

To model an ANC such as (3) which marks one of the arguments as a Poss, I first need to define a possessive strategy on the Adnominal Possession page. I create a possessor-first (head-final) specifier possessive strategy involving an affix (*y-*) on the possessum which agrees in person with the possessor. The choices for this possessive strategy are shown in (4).

(4)
```
section=adnom-poss
  poss-strat1_order=head-final
  poss-strat1_mod-spec=spec
  poss-strat1_mark-loc=possessum
  poss-strat1_possessum-type=affix
  poss-strat1_possessum-affix-agr=agree
```

[9]To save space, I only show the most relevant TDL constraints. I omit any constraints which just pass up information or else are there to reduce ambiguity.

[10]I provide more detail about Hixkaryana in chapter 6, as it is one of the illustrative languages I used to develop the analysis and implementation.

I then define two nominal pcs on the Morphology page, one for possessive prefixes (*y-*) and one for possessive suffixes (*ri-*). The possessive prefix pc serves as input to the possessive suffix pc.

Since the action nominal in (3) marks its A with the *wya* postposition and Hixkaryana [hix] does not use case-marking, I define *wya* as a semantically empty adposition on the Lexicon page. The choices for this are shown in (5).

(5)
```
section=lexicon
  adp1_orth=wya
  adp1_order=after
```

These choices indicate the orthographic form of *wya* and that it should appear after the noun it marks.

Returning to the ANC in (3) itself, since the P is marked as a Poss, on the Nominalized Clauses page, I create an ERG-POSS nominalization strategy which can act on both intransitive and transitive verbs and can be modified by adverbs. As Hixkaryana [hix] does not have determiners, I state that a determiner appearing with an action nominal is impossible (det=imp). The word order in terms of semantic arguments in (3) is A P action.nominal. This contrasts with the OVS word order in Hixkaryana finite clauses which in terms of semantic arguments maps to a P V A word order. I thus specify that Hixkaryana [hix] uses SOV (A P action.nominal) word order in ANCs. Finally, I allow this nominalization strategy to use the possessive strategy I defined above (poss-strat1) and state that ANCs should use the both semantic option (the P argument will be interpreted as both the possessor and the patient of the action nominal). Specifying that ANCs can use poss-strat1 ensures the P argument of the ANC will receive the correct possessive marking. The marking of the A argument is handled on the Morphology page which I discuss next. I give the strategy the name subordinate_poss, which I will use to refer to it on the other library pages. I show all the choices from the Nominalized Clauses page in (6).

(6)
```
section=nominalclause
  ns1_name=subordinate_poss
```

```
    ns1_nmz_type=erg-poss
    ns1_det=imp
    ns1_intrans=on
    ns1_trans=on
    ns1_adv=on
  same-word-order=no
  nmz-clause-word-order=sov
    nmz_poss_strat1_name=poss-strat1
  non_sent_sem=both
```

Now that a nominalization strategy has been defined, I create a nominalization pc on the Morphology page which accepts any verb as input. I then define an lrt called action_nom for the nominalizaing *ni-* suffix. I mark this lrt with the feature [NOMINALIZATION subordinate_poss] indicating that it should use the subordinate_poss nominalization strategy. I also specify that the object of the lrt (the action nominal's complement) should take the FORM value *wya_sem* indicating that it should be marked by the semantically empty *wya* postposition. I give the choices for this lrt in (7).

(7)
```
section=morphology
  verb-pc71_name=Nominalization
  verb-pc71_order=suffix
  verb-pc71_inputs=verb
    verb-pc71_lrt1_name=action_nom
      verb-pc71_lrt1_feat1_name=nominalization
      verb-pc71_lrt1_feat1_value=subordinate_poss
      verb-pc71_lrt1_feat1_head=verb
      verb-pc71_lrt1_feat2_name=form
      verb-pc71_lrt1_feat2_value=wya_sem
      verb-pc71_lrt1_feat2_head=obj
      verb-pc71_lrt1_lri1_inflecting=yes
      verb-pc71_lrt1_lri1_orth=-ni
```

Finally, (3) has the ANC appearing within an adverbial *ke* 'because' phrase. I then define a clausal modifier strategy on the Clausal Modifiers page which uses a free *ke* 'because' morpheme to mark the modifier phrase. I also specify that the *ke* phrase must use the subordinate_poss nominalization strategy. This ensures that the complement of *ke* 'because' will be nominalized. I show the relevant choices in (8).

(8)
```
section=clausalmods
    cms1_freemorph1_orth=ke
    cms1_freemorph1_pred=_because_subord_rel
    cms1_feat1_name=nominalization
    cms1_feat1_value=subordinate_poss
```

Turning to how the functions in `nominalized_clauses.py` parse these choices, since the defined nominalization strategy acts on both transitive and intransitive action nominals `add_anc_lexrules()` adds an intransitive version of the action_nom lrt to the choices file. This new lrt has all the same choices as the action_nom lrt aside from its name (action_nom_intrans) and that it lacks the choices regarding the FORM feature requirement on the object. The effect of `add_anc_lexrules()` is then as if the user had defined another lrt which was identical to action_nom aside from the constraints on the object argument. Since none of the other criteria for extra rules are met and action_nom is the only nominalization lrt in this choices file, action_nom_intrans is the only lrt added to the choices file. `update_lexical_rules()` is then called and loops through all the lexical rules which take the nomilization feature, of which after `add_anc_lexrules()`, there are now two (*action_nom-lex-rule* and *action_nom_intrans-lex-rule*).

`update_lexical_rules()` adds the appropriate nominalization leaf type for each lrt. *action_nom-lex-rule* inherits from *trans-erg-poss-lex-rule* and *action_nom_intrans-lex-rule* inherits from *non-sent-anc-intrans-lex-rule* since both are ERG-POSS strategies which act on transitive and intransitive verbs respectively.

`update_lexical_rules()` also adds any constraints that will not be handled by the morphotactics library which in this situation involves adding the constraint [HEAD adp] to the COMPS value of *action_nom-lex-rule*. This ensures the complement of the action nominal can only be an adposition. I show the results of this function in terms of TDL in (9).

(9)
```
action_nom-lex-rule := trans-erg-poss-lex-rule &
  [ SYNSEM.LOCAL.CAT.VAL.COMPS < [ LOCAL.CAT.HEAD adp ] > ].

action_nom_intrans-lex-rule :=  non-sent-anc-intrans-lex-rule.
```

The next function called `add_nmz_feature()` introduces NMZ as a boolean head feature and sets the NMZ value of both *noun-lex* and *verb-lex* to −. This is shown below in (10) using just *noun-lex* as an example.

(10)
```
head :+ [ NMZ bool ].

noun-lex := [ SYNSEM.LOCAL.CAT.HEAD.NMZ - ].
```

The function `add_anc_lex_supertpe()` then adds the TDL for *anc-lex-rule* (the type inherited by all nominalization lrts) with the most relevant constraints shown in (11).

(11)
```
anc-lex-rule := [ SYNSEM.LOCAL.CAT.HEAD noun & [ NMZ + ],
                  DTR.SYNSEM.LOCAL.CAT.HEAD verb ].
```

The next function `add_nmz_clause_phrases()` is only called if either a SENT or ALT-SENT strategy is defined and is thus skipped for this choices file. The following function `add_nmz_lexrules()` adds the TDL for the remaining necessary nominalization subtypes, which for this nominalization strategy includes *anc-low-nmz-lex-rule*, *trans-erg-poss-lex-rule_supertype*, *non-sent-anc-intrans-lex-rule_supertype*, *non-sent-anc-intrans-lex-rule* and *trans-erg-poss-lex-rule*.[11] I show the most relevant constraints from these types in (12).

(12)
```
anc-low-nmz-lex-rule := anc-lex-rule &
      [ SYNSEM.LOCAL.CAT.VAL.SUBJ < >,
        C-CONT [ RELS.LIST < [ PRED "nominalized_rel", LBL #ltop,
                               ARG0 ref-ind & #arg0, ARG1 #arg1 ] >,
                 HCONS.LIST < qeq & [ HARG #arg1, LARG #larg ] >,
                 HOOK [ XARG #xarg, INDEX #arg0, LTOP #ltop ] ],
        DTR.SYNSEM.LOCAL.CONT.HOOK [ XARG #xarg, LTOP #larg ] ].

non-sent-anc-intrans-lex-rule_supertype := anc-low-nmz-lex-rule &
  [ SYNSEM.LOCAL.CAT.VAL [ COMPS #comps ],
    DTR.SYNSEM.LOCAL.CAT.VAL.COMPS #comps & < > ].
```

[11] In languages without determiners such as Hixkaryana [hix], the constraints across types such as *trans-erg-poss-lex-rule* and *trans-erg-poss-lex-rule_supertype* could be collapsed into one type. The implementation could then be cleaned up by only creating the types ending in *_supertype* when a language has action nominals that take both a Poss and a determiner.

```
non-sent-anc-intrans-lex-rule := non-sent-anc-intrans-lex-rule_supertype &
  [ SYNSEM.LOCAL.CAT.VAL.SPR < [ LOCAL [ CONT.HOOK.INDEX #subj,
                                         CAT.HEAD.POSSESSOR possessive ] ] >,
    DTR.SYNSEM.LOCAL.CAT.VAL.SUBJ < [ LOCAL.CONT.HOOK.INDEX #subj ] > ].

trans-erg-poss-lex-rule_supertype := anc-low-nmz-lex-rule &
  [ SYNSEM.LOCAL.CAT.VAL.COMPS < [ LOCAL [ CONT.HOOK.INDEX #subj ] ] > ,
    DTR.SYNSEM.LOCAL.CAT.VAL.SUBJ < [ LOCAL.CONT.HOOK.INDEX #subj ] > ].

trans-erg-poss-lex-rule := trans-erg-poss-lex-rule_supertype &
  [ DTR.SYNSEM.LOCAL.CAT.VAL.COMPS < [ LOCAL.CONT.HOOK.INDEX #obj ] >,
    SYNSEM.LOCAL.CAT [ VAL.SPR < [ LOCAL [ CONT.HOOK.INDEX #obj,
                                CAT.HEAD.POSSESSOR possessive ] ] >] ].
```

Since this choices file contains at least one adverb lexical entry, `add_nmz_mod_constraints()` adds all the common constraints involved in the ADV-MOD feature analysis and also sets the ADV-MOD value for each nominalization lrt. Since all the nominalization lrts in this choices file use a nominalization strategy that can only be modified by adverbs, the ADV-MOD value for all nominalization lrts is set to +. I show these constraints in (13).

(13)
```
head :+ [  ADV-MOD luk ].

noun-lex := [SYNSEM.LOCAL.CAT.HEAD.ADV-MOD - ].

verb-lex := [ SYNSEM.LOCAL.CAT.HEAD.ADV-MOD + ].

adverb-lex-item := [ SYNSEM.LOCAL.CAT.HEAD.MOD < [ LOCAL.CAT.HEAD +nv &
                                                   [ ADV-MOD + ]]  >] .

action_nom_intrans-lex-rule := [ SYNSEM.LOCAL.CAT.HEAD.ADV-MOD +  ].

action_nom-lex-rule := [ SYNSEM.LOCAL.CAT.HEAD.ADV-MOD + ].
```

Since the Hixkaryana [hix] choices file does not contain any coordination strategies, the function `add_anc_coord_constraints()` is skipped. The next function `handle_anc_spr_restrictions()` iterates through all the nominalization lrts and adds any

constraints related to mandatory specifiers and determiners. For this nominalization strategy, it is specified (ns1_det=imp) that a determiner appearing with an action nominal is impossible. Thus, in accordance with the analysis in section 4.5.2, the constraint shown in (14) is added to all nominalization lrts using this strategy.[12]

(14)
```
action_nom_intrans-lex-rule := [ SYNSEM.LOCAL.CAT.VAL.SPR
                       < [ LOCAL.CAT.HEAD.POSSESSOR possessive ] > ].

    action_nom-lex-rule :=  [ SYNSEM.LOCAL.CAT.VAL.SPR
                       < [ LOCAL.CAT.HEAD.POSSESSOR possessive ] >  ] ].
```

Constraining the specifier of the action nominal to be [POSSESSOR possessive] requires the specifier to be a syntactic possessor and not a determiner.

The final function `set_anc_wo_value()` is called whenever extra head-comp rules need to be added to the grammar. The nominalization strategy in this choices file uses SOV word order in ANCs while finite clauses in Hixkaryana [hix] use OSV word order. Since Hixkaryana [hix] has OSV word order, *comp-head-phrase* is added to the grammar by existing code in `word_order.py`.

Using table 4.1 from section 4.5.1, in ERG-POSS action nominals, an SOV word order maps to the following syntactic positions COMP POSS NMZ-verb. This means that ANCs also require a comp-head rule. However, with a COMP POSS NMZ-verb word order the action nominal must first combine with its Poss argument before it can take its complement. The comp-head rule is thus constrained to be [SPR $\langle\ \rangle$] to ensure the Poss argument (realized as a specifier) is picked up first. Since a constraint is placed on a comp-head rule which would be unnecessary for other verbs a version of the comp-head rule exclusive to action nominals is necessary.

The new comp-head rule itself is added by `word_order.py`, but `set_anc_wo_value()` is responsible for introducing and setting the ANC-WO value on any transitive POSS-ACC/ERG-POSS/NOMINAL leaf subtypes (only *trans-erg-poss-lex-rule* in this grammar) as well as var-

[12]In a language without determiners, it is also possible to just not answer the question regarding determiners and then no constraints would be added.

ious other lexical items including *transitive-verb-lex* which is the type inherited by all transitive verbs. Setting the type *trans-erg-poss-lex-rule* to have a value of [ANC-WO +] ensures that all transitive ERG-POSS action nominals will use the added comp-head rule and not the original one. Likewise setting the ANC-WO value of *transitive-verb-lex* to [ANC-WO −] guarantees that regular transitive verbs will only use the original comp-head rule and not the added one. I show the added TDL for *transitive-verb-lex* and *trans-erg-poss-lex-rule* in (15).

(15)
```
head :+ [ ANC-WO bool]

transitive-verb-lex := [ SYNSEM.LOCAL.CAT.HEAD.ANC-WO - ].

trans-erg-poss-lex-rule := [ SYNSEM.LOCAL.CAT.HEAD.ANC-WO + ].
```

Looking back at the choices from the nominalized clauses section, there are two remaining parts that have yet to be addressed: (1) the use of possessive strategies and (2) a full account of ANC word order. The first is handled by `adnominal_possessives.py` which adds the TDL for the appropriate [NMZ +] and [NMZ −] subtypes. For the particular possessive strategy defined in this choices file, possessive semantics are added by the unary rule (*poss-unary-phrase*) which takes the possessor as a daughter. The possessor and possessum are then combined through the *spec-head-phrase* rule. Code in `adnominal_possessives.py` creates [NMZ +] (*anc-poss-unary-phrase*) and [NMZ −] *noun-poss-unary-phrase* subtypes of *poss-unary-phrase* and adds the constraints corresponding to the both (non_sent_sem=both) semantic option to only *anc-poss-unary-phrase*.

`adnominal_possessives.py` also creates [NMZ +] (*anc-spec-head-phrase*) and [NMZ −] (*noun-spec-head-phrase*) subtypes of *spec-head-phrase* so that action nominals contain the more verbal semantics expected when verbs combine with their arguments as opposed to the nominal semantics seen when nouns combine with their specifiers. `adnominal_possessives.py` is also responsible for adding the rule instantiations of both versions of the *spec-head-phrase* rule (a phrase structure rule) to `rules.tdl`. I show the TDL for the [NMZ +] subtypes (looking only at the most relevant constraints) in (16).

(16)
```
language.tdl:
anc-poss-unary-phrase-1 := poss-unary-phrase-1 &
  [ C-CONT [ PRED "poss_rel", LBL #lbl,
             ARG1 #possessum, ARG2 #possessor ],
    ARGS < [ SYNSEM.LOCAL.CONT.HOOK.INDEX #possessor ] >,
    SYNSEM.LOCAL [ CONT.HOOK.INDEX #possessor,
                   CAT.VAL.SPEC < [ LOCAL [ CAT.HEAD.NMZ + ] ] > ] ].

anc-spec-head-phrase := spec-head-phrase & head-compositional &
  [ SYNSEM.LOCAL.CAT.HEAD.NMZ +,
    NON-HEAD-DTR.SYNSEM.LOCAL.CAT.HEAD.POSSESSOR possessive ].

rules.tdl:
anc-spec-head := anc-spec-head-phrase.
```

The constraints on *anc-poss-unary-phrase-1* identify the ARG2 (the possessor) of **poss_rel** with the INDEX of the rule daughter and mother. Identiying ARG2 with the the INDEX of the rule daughter ensures that the daughter is interpreted as the possessor. The result of *anc-poss-unary-phrase* goes on to be the NON-HEAD-DTR (the specifier) of *anc-spec-head-phrase*. Identifying ARG2 with the INDEX of the mother then allows this value to be available to *anc-spec-head-phrase* where it can be further identified with one of the arguments of the action nominal. The constraints on *anc-poss-unary-phrase-1* then allow the daughter of *anc-poss-unary-phrase* to be interpreted as both a possessor and as one of the arguments of the action nominal (the both semantic option). *anc-spec-head-phrase* inherits from *head-compositional* which ensures it will have the correct verbal semantics. The NON-HEAD-DTR of *anc-spec-head-phrase* is constrained to be possessive which requires all specifiers to be syntactic possessors.[13]

The addition of *anc-spec-head-phrase* allows action nominals to combine with their P argument (the Poss argument) in the correct order. However, as mentioned above, an extra comp-head rule is also necessary. This comp-head rule (*anc-comp-head-phrase*) is

[13]This constraint is redundant in grammars without determiners, as in these grammars there are no nonpossessive specifers. However, this constraint could become useful if a future analysis introduces a specifier which is neither a determiner nor a syntactic possessor.

needed for the action nominal to combine with its A argument. The head daughter of *anc-comp-head-phrase* is constrained to be [ANC-WO +] which only allows [ANC-WO +] POSS-ACC/ERG-POSS/NOMINAL action nominals to use it. *anc-comp-head-phrase* is added by `word_order.py` which also constrains the head-daughter of the original *comp-head-phrase* to be [ANC-WO −]. This prevents POSS-ACC/ERG-POSS/NOMINAL action nominals from using *comp-head-phrase*. `word_order.py` is also responsible for adding the rule instantiation of *anc-comp-head-phrase* to `rules.tdl`. I show the TDL added by `word_order.py` in (17).

(17) `language.tdl:`

```
comp-head-phrase := [ HEAD-DTR.SYNSEM.LOCAL.CAT.HEAD.ANC-WO - ].

anc-comp-head-phrase :=  [ HEAD-DTR.SYNSEM.LOCAL.CAT.HEAD.ANC-WO +,
                           SYNSEM.LOCAL.CAT.VAL.SPR < > ].

rules.tdl:
anc-comp-head := anc-comp-head-phrase.
```

As mentioned above, *anc-comp-head-phrase* is constrained to have an empty SPR value to ensure that an action nominal combines with its Poss argument before it takes a complement.

While not a choice, since all action nominals have optional arguments by default, `argument_optionality.py` adds the TDL for the [NMZ +] optional subject and optional complement rules for all choices files that have defined nominalization strategies. Since Hixkaryana allows for dropped arguments in finite clauses as well, [NMZ −] versions of these rules are also added to the grammar to be used by non-nominalized verbs. `argument_optionality.py` adds the rule instantiations for all of these rules to `rules.tdl`. (18) shows the added TDL.

(18) `language.tdl:`

```
anc-head-opt-comp-phrase := basic-head-opt-comp-phrase &
  [ HEAD-DTR.SYNSEM.LOCAL.CAT.NMZ +].

anc-decl-head-opt-subj-phrase := decl-head-opt-subj-phrase &
  [ HEAD-DTR.SYNSEM.LOCAL.CAT.HEAD.NMZ + ].
```

```
regular-decl-head-opt-subj-phrase := decl-head-opt-subj-phrase &
  [ HEAD-DTR.SYNSEM.LOCAL.CAT.HEAD.NMZ - ].

regular-head-opt-comp-phrase := basic-head-opt-comp-phrase &
  [ HEAD-DTR.SYNSEM.LOCAL.CAT.HEAD.NMZ - ].

rules.tdl:
anc-head-opt-comp := anc-head-opt-comp-phrase.

anc-decl-head-opt-subj := anc-decl-head-opt-subj-phrase.

regular-decl-head-opt-subj := regular-decl-head-opt-subj-phrase.

regular-head-opt-comp := regular-head-opt-comp-phrase.
```

The final component that needs to be addressed is how this choices file specifies that the action nominal should take nominal morphology. In general this is done by selecting a nominalization lrt as input to a nominal pc. This allows flexibility as to which affixes an action nominal should take. In this example Hixkaryana [hix] choices file, I add the nominalization pc as input to the possessive prefix pc. This allows action nominals to take both possessive prefixes and suffixes. `morphotactics.py` is responsible for removing any verbal position flags and adding the appropriate nominal flags to the nominalization pc. I give the corresponding TDL in (19).

(19)
```
nominalization-lex-rule-super :=
  [ INFLECTED [ POSS_PERSON_PREFIX-FLAG -,
                POSS_SUFFIX-FLAG - ] ].
```

nominalization-lex-rule-super is the type associated with the overall nominalization pc. Setting both POSS_PERSON_PREFIX-FLAG and POSS_SUFFIX-FLAG to – ensures that all lrts in this pc will have to go through both the possessive prefix and possessive suffix pcs to be inflectionally satisfied. Aside from `lexical_items.py` constraining all lexical types to be [NMZ −] and [ANC-WO +] this then covers all the TDL added to the grammar based on this set of choices.

5.4 Summary

The chapter has provided an overview of both the questionnaire questions used to elicit information about nominalization strategies in a given language, as well as how those answers are translated into TDL by Python code to be added to a downloadable grammar. The next chapter discusses the evaluation of the implemented grammar by looking at the coverage and over-generation achieved on a set of pseudo, illustrative, and held-out languages.

Chapter 6
EVALUATION

I evaluate the action nominal constructions (ANCs) library by testing the coverage and over-generation of the implemented analysis on a series of pseudo, illustrative, and held-out languages. Pseudo languages are constructed languages designed to test different combinations of questionnaire choices, while illustrative and held-out languages are real languages. Development is guided by both the pseudo and illustrative languages which help to flesh out the analysis and uncover any bugs. Once development is frozen, I use the held-out languages to test the robustness of the implemented library on unseen language data.

In the same style of the regression tests discussed in section 2.3, each test consists of a testsuite and a choices file. The testsuite contains both grammatical and ungrammatical strings. The customization system is then used with the choices file to produce a grammar, which is subsequently used to parse all the sentences in the testsuite. The performance of a grammar is based on its coverage of grammatical strings as well as its over-generation of ungrammatical strings. Each parsed sentence is also manually observed to ensure the resulting tree and semantic representation are correct. In all the below results, 100% coverage means that each grammatical string in a testsuite received the expected parse tree and MRS representation. This does not mean that this is the only interpretation the string received as some sentences are ambiguous. The choice files and testsuites for all the languages discussed in this chapter can be found in the GitHub Grammar Matrix repository under `matrix/tests/regression/choices` and `matrix/tests/regression/txt-suites` respectively. Sections 6.1, 6.2, and 6.3 in turn discuss the evaluation of the pseudo, illustrative, and held-out languages.

6.1 Pseudo Languages

I constructed pseudo languages by creating valid combinations of the choices shown in table 6.1.

Table 6.1: Different Pseudo language choice options

choice	**choice-answer-options**
mandatory-arg	on, off
regular-word-order	SOV, SVO, SVO, VSO, OSV, OVS, VOS, V-final, V-initial, free, v2
nmz-type	SENT, ALT-SENT, ALL-COMPS, POSS-ACC, ERG-POSS, NOMINAL
valence	intrans, trans, both
nmz_mod	adj, adv, both, none
mandatory-arg-on	subj, obj, mand-spr, obj2
same-word-order	yes, no
ANC-word-order	SOV, SVO, SVO, VSO, OSV, OVS, VOS, V-final, V-initial, free, v2
nmzRel	yes, no
non-sent-sem	noun-only, verb-only, both
poss_strat	
mand-spr	on, off
has-dets	yes, no
noun-det-order	noun-det, det-noun
det	obl, opt, imp
all_comps_order	agent-like, patient-like
single-arg	on, off
nmz-position	subj-position, obj-position, clausalcomp, modifier

mod-marker	free morph, free morph pair, no free morph
alt-arg-marking	case, adp

To keep the number of pseudo languages manageable, I sampled different sets of choices until every choice had been used in a language at least once. This resulted in a total of 32 pseudo languages. Each test contains a single nominalization strategy. I also created a few additional tests on top of these 32 to test aspects such as coordination and interactions across nominalization strategies which are not covered by the choices in table 6.1. I first discuss the set of 32 initial pseudo languages and then discuss the extra tests that were added.

Table 6.2 provides explanations for the different choices displayed in table 6.1 along with the questionnaire web page where the question corresponding to the choice is found. I explain what the psuedo library page option means below. If a choice is only relevant for a particular ANC type, I put all the applicable ANC types at the end of the explanation in parenthesis. For example, the same-word-order choice is only relevant for POSS-ACC/ERG-POSS/NOMINAL action nominals.

Table 6.2: Pseudo language choice explanations

choice-name	**library page**	**question/explanation**
mandatory-arg	nmz_clause	Do ANCs have any mandatory arguments?
regular-word-order	word_order	Main clause word order
nmz-type	nmz_clause	What is the nominalization type of the strategy?
valence	nmz_clause	Does this nominalization strategy apply to intransitive verbs, transitive verbs or both?
nmz_mod	nmz_clause	Can this nominalization strategy be modified by adjectives or adverbs?

mandatory-arg-on	pseudo	Which argument of the action nominal should be marked obligatory (subject, object, Poss, second object)
same-word-order	nmz_clause	Do ANCs have the same word order as finite clauses? (POSS-ACC/ERG-POSS/NOMINAL)
ANC-word-order	nmz_clause	The word order in ANCs (POSS-ACC/ERG-POSS/NOMINAL)
nmzRel	nmz_clause	Is **nominalized_rel** included in the semantics of ANCs? (SENT/ALT-SENT)
non-sent-sem	nmz_clause	Possessive semantic type of action nominals (POSS-ACC/ERG-POSS/NOMINAL)
poss_strat	nmz_clause	One of 41 specifier possessive strategies selected from the regression tests created for the adnominal possession library (POSS-ACC/ERG-POSS/NOMINAL)
mand-spr	nmz_clause	Whether a possessor must overly appear in ANCs (POSS-ACC/ERG-POSS/NOMINAL)
has-dets	word_order	Does the language have determiners?
noun-det-order	word_order	Noun-det word order in languages that have determiners
det	nmz_clause	Are determiners optional (opt), obligatory (obl), or impossible (imp) in ANCs when a syntactic possessor is not present? (POSS-ACC/ERG-POSS/NOMINAL/ALL-COMPS)

all_comps_order	nmz_clause	Whether the patient-like or agent-like argument is the action nominal's first complement in ANCs (ALL-COMPS)
single-arg	nmz_clause	Is the single-possessor analysis (discussed in section 4.5.3) used by this nominalization strategy? (POSS-ACC/ERG-POSS/NOMINAL)
nmz-position	pseudo	Which position is nominalized (subject, object, clausal complement, or clausal modifier)
mod-marker	clausal_mods	If a clausal modifier is nominalized, does the modifier have a free subordinator morpheme, a free subordinator morpheme pair or no free subordinator morpheme?
alt-arg-marking	pseudo	In all strategies where there are arguments that take distinct marking from either the underlying verb or possessives strategies, are these arguments marked by a case-change or by an adposition?

All the questions on the nominalized clauses (nmz_clause) web page are included in table 6.2 along with necessary questions on other pages such as what the word order is for finite clauses. It is necessary to know the regular word order in a language to answer the question regarding whether ANCs also use this word order. Table 6.2 also contains a number of pseudo questions (library page = pseudo). These are not actual questions on the questionnaire, but are included to increase the range of options explored by the pseudo languages. For example, nmz-position refers to which position in a sentence is nominalized with the possibilities ranging from subject position, object position, clausal modifier position,

or clausal complement position. This is not a question that the user directly has to answer, but they do have the option to, for example, define clausal complement or clausal modifier strategies and require them to use a particular nominalization strategy.

I then change which sentences I add to the testsuite and potentially add new choices to the choices file based on the nmz-position choice selection. For example, a pseudo language with a clausal modifier nmz-position will include a defined clausal modifier strategy in the choices file, and all the examples I add to the testsuite will have nominalized clausal modifers. Any pseudo language which has a different value for nmz-position lacks these choices regarding clausal modifiers. Unlike the clausal complement and clausal modifier nmz-position options, no additional choices are added to a pseudo language if subject or object is selected as the nmz-position. This is because any language with action nominals taking a *ref-ind* INDEX can appear in these positions. Instead a language having subject or object nmz-position just changes what sentences I include in the testsuite. For example a language with a subject nmz-position will include only ANCs in subject position in the testsuite. Additionally, both subject and object nmz-position are incompatible with SENT/ALT-SENT ANCs that do not contain nominal semantics. This combination of choices (nmzRel no, nmz-position subject) is then considered invalid. Including nmz-position as a choice ensures the library is being tested for a variety of locations where nominalized clauses can appear, as well as testing for successful interactions between different related libraries such as clausal complements and clausal modifiers.

The two remaining pseudo choices mandatory-arg-on and alt-arg-marking. The mandatory-arg-on choice selects one of the arguments of the action nominal to be obligatory. I then add the corresponding choices to the pseudo language. The alt-arg-marking option is relevant for ERG-POSS/NOMINAL/ALT-SENT/ALL-COMPS action nominals where there are arguments that are both distinct from the arguments of the underlying verb and are not marked by possessive strategies. For example, in ERG-POSS action nominals, the complement of the action nominal can take a distinct form of marking. This alternative marking can either be a case-change or a semantically empty adposition. Both of these markings are

specified on the Morphology page by placing feature (CASE or FORM) constraints on the arguments of a nominalization lrt. While a user always has the option to use either or both marking strategies in ANCs, I choose one for each pseudo language where relevant to ensure that both strategies are working as expected.

The poss_strat choice in table 6.1 which is left empty refers to the possessive strategies that are used in POSS-ACC/ERG-POSS/NOMINAL ANCs. Since there are a potentially infinite number of possibilities for different combinations of possessive strategies, one of the 41 specifier[1] adnominal possession pseudo language regression tests are selected at random to serve as the possessive strategy in the generated ANC pseudo languages (Nielsen, 2018). Each of these 41 tests contains choices for at least one adnominal possession strategy which are reused to serve as possessive strategies in ANCs. A possessive strategy is selected for each pseudo language, even those containing SENT/ALT-SENT/ALL-COMPS ANCs to test if the ANC library is interfacing correctly with the possessive strategy library.

As mentioned above, in addition to the 32 generated pseudo languages, I also added several additional pseudo languages to test other possibilities. Six of these extra pseudo languages test how coordination interacts with different nominalization and possessive strategies. Each coordination-focused pseudo language contains two noun/NP coordination strategies. One coordination strategy requires the NMZ feature to be the same between both conjuncts and the mother, while the other allows for conjuncts with different NMZ values, but constrains the mother to be [NMZ +] in that scenario. In the latter coordination strategy the value of the mother is still the same as the conjuncts when the conjuncts have the same NMZ value. Two of the six languages test these two coordination strategies with a SENT nominalization strategy that either contains **nominalized_rel** in its semantics or lacks **nominalized_rel** in its semantics. The four remaining languages contain a POSS-ACC nominalization strategy which takes one of four possessor-first possessive strategies: a juxtaposition strategy, a pronominal possessor affix , an affix on the possessor, and a non-affix

[1]All possessive strategies discussed for any of the three language types (pseudo, illustrative, or held-out) are defined to be specifier strategies as modifier possessor strategies cannot be used in ANCs.

on the possessor.

The final five tests consist of languages which contain two nominalization strategies to test how different strategies interact with one another. One language consists of both a POSS-ACC and an ERG-POSS nominalization strategy. The four remaining pseudo languages contain a SENT strategy paired with a POSS-ACC strategy and vary based on whether the SENT strategy contains nominalization semantics or not and whether the POSS-ACC ANC has the same word order as the SENT ANC or not.

With these added pseudo languages, the total number of pseudo languages is 43. The number of grammatical and ungrammatical strings in each testsuite along with the percent of coverage and over-generation for each language is shown in table 6.3. Also displayed is the average number of parses per parsed string in each testsuite.

Table 6.3: Pseudo language test results

	# grammatical	# ungrammatical	% coverage	% over-generation	avg-analysis-#
anc1	10	2	100	0	1.3
anc2	7	1	100	0	1
anc3	9	3	100	0	1
anc4	9	3	100	0	1
anc5	13	8	100	0	2
anc6	13	14	100	0	1
anc7	10	6	100	0	1.1
anc8	12	6	100	0	1
anc9	13	7	100	0	1.23
anc10	13	10	100	0	1.08
anc11	8	8	100	0	1
anc12	11	8	100	0	1.09

anc13	10	6	100	0	1.5
anc14	9	9	100	0	1.11
anc15	12	5	100	0	1
anc16	8	9	100	0	1.25
anc17	12	8	100	0	1.17
anc18	11	2	100	0	4.27
anc19	10	8	100	0	1.3
anc20	7	7	100	0	1.14
anc21	13	10	100	0	1.62
anc22	7	8	100	0	1
anc23	14	10	100	0	1.29
anc24	5	8	100	0	1
anc25	10	19	100	0	1
anc26	10	14	100	0	4.43
anc27	9	12	100	0	1
anc28	9	13	100	0	3.44
anc29	8	13	100	0	1.13
anc30	13	11	100	0	1.15
anc31	8	9	100	0	1.5
anc32	7	8	100	0	1.29
anc33	10	9	100	0	1.5
anc34	6	8	100	0	1.17
anc35	8	13	100	0	1.13
anc36	12	10	100	0	1.42
anc37	6	1	100	0	1.33
anc38	4	3	100	0	1

anc39	8	8	100	0	1.25
anc40	10	12	100	0	1
anc41	10	9	100	0	1.1
anc42	10	5	100	0	1
anc43	10	9	100	0	1.1

6.2 Illustrative Languages

The four languages I used as illustrative languages were English [eng], Hixkaryana [hix], Russian [rus], and Korean [kor]. I picked these languages as they contain a good range of the phenomena handled by the library including all 6 nominalization types as well as different ANC dependent marking strategies (case-marking vs. adposition marking). The next four sections will describe ANCs in each language and how I modeled them using the implemented library. I report coverage, over-generation, and the average number of parses per string at the end of this section for all four languages in table 6.5.

As a general implementation note, for all languages with SENT/ALT-SENT strategies I include **nominalized_rel** in the semantics, and for all languages with POSS-ACC/ERG-POSS/NOMINAL strategies I use the both (the Poss argument is interpreted as a possessor and an argument of the verb) semantic option. I choose to include nominal semantics for all the SENT/ALT-SENT strategies since the grammar resources I use either do not deny the possibility that ANCs can appear in subject/object position or explicitly state that they can. This then requires all SENT/ALT-SENT action nominal to have a *ref-ind* INDEX. Thus, in all the below implemented grammars, any ANC can appear in either subject or object position. If a language additionally allows nominalized clausal complements or clausal modifiers, I define the appropriate clausal complement or modifier strategies and specify that they can use one of the defined nominalization strategies. In all the below languages, I mention which positions (subject, object, clausal complement, clausal modifier) are covered by the sentences in the testsuite. With regard to POSS-ACC/ERG-POSS/NOMINAL ANCs,

none of the resources I use provide in-depth descriptions of the semantics of ANCs containing Poss arguments and so I stick to the both semantic option for consistency. Since these details are kept the same across all languages I do not repeat them below.

One final important note is that the majority of strings in the testsuites for these languages are constructed based on grammars and have not been vetted by native speakers of these languages. Thus, there are very likely nuances or more fine-grained grammatically judgments[2] that are missed by these tests.

6.2.1 English [eng]

Data/Description

I base the following description of English [eng] ANCs and the sentences in my English [eng] testsuite off of the discussion in Malouf (1998) and the examples provided in Koptjevskaja-Tamm (2013). English [eng] is an Indo-European language which has SVO word order in finite clauses and det-noun word order in noun phrases. It has a limited nominative-accusative case-marking system which is primarily visible only in pronouns. English [eng] has three different possessive strategies that can appear in ANCs shown below in (1)–(3).

(1) *Sam 's friend*
Sam POSS friend
'Sam's friend.' [eng]

(2) *His friend*
3SG.POSS friend
'My friend.' [eng]

(3) *The friend of Sam*
The friend of Sam
'The friend of Sam.' [eng]

[2]For example, as will be discussed below, the ungrammatical sentence that parses for English [eng] is one that I can tell is ungrammatical as a native speaker, but not one that I would have known was ungrammatical looking at my available resources. There are potentially similar examples (at least among the sentences I construct) in the other languages that are ungrammatical, but the resources I use do not go into enough detail for me to be able to detect as such.

(1) and (2) are both possessor-first strategies where the possessor in (1) is marked by the *'s* marker, and the possessor in (2) is marked by a possessive pronoun (*his*). (3) contains a possessum first strategy where the possessor is marked by the presposition *of*.

The different kinds of ANCs in English are shown in (4)–(7) with the ANC appearing in a *by* clausal modifier phrase. The *-ing* suffix acts as a nominalizer.

(4) (ALT-SENT)

The	*DA*	*was*	*shocked*	*by*	*him*	*destroy-ing*	*the*
The	DA[NOM]	was	shocked	by	3SG[ACC]	destroy-NMLZ	the

evidence
evidence[ACC]
'The DA was shocked by him destroying the evidence.' [eng] adapted from (Malouf, 1998: 121)

(5) (POSS-ACC)

The	*DA*	*was*	*shocked*	*by*	*Pat*	*'s/his*	*destroy-ing*
The	DA[NOM]	was	shocked	by	Pat	POSS/3SG[POSS]	destroy-NMLZ

the evidence
the evidence[ACC]
'The DA was shocked by Pat's/his destroying the evidence.' [eng] adapted from (Malouf, 1998: 121)

(6) (NOMINAL)

The	*DA*	*was*	*shocked*	*by*	*Pat*	*'s/his*	*destroy-ing*
The	DA[NOM]	was	shocked	by	Pat	POSS/3SG[POSS]	destroy-NMLZ

of the evidence
of the evidence
'The DA was shocked by Pat's/his destroying of the evidence.' [eng] adapted from (Malouf, 1998: 121)

(7) (ALL-COMPS)

The	*DA*	*was*	*shocked*	*by*	*the*	*destroy-ing*	*of*	*the*	*evidence*
The	DA[NOM]	was	shocked	by	the	destroy-NMLZ	of	the	evidence

by *him*
by him[ACC]
'The DA was shocked by the destroying of the evidence by him.' [eng] modeled based on the English datapoint in Koptjevskaja-Tamm (2013)

(4) and (5) both have the action nominal *destroying* taking the same complement as the underlying verb, with (4) marking the agent-like (A) argument with accusative case and (5) marking this argument as a Poss. (6) marks the A as a Poss and the patient-like argument (P) using an *of* preposition. (7) marks the A with a *by* preposition and the P with an *of* preposition. (5) and (6) can be classified as POSS-ACC and NOMINAL respectively. As discussed in sections 4.2.1 and 4.2.2, examples such as (4) and (7) were the main motivation for the creation of the ALT-SENT and ALL-COMPS nominalization types.

For all four sentences, it is possible to use the same strategy on intransitive verbs. For intransitive ANCs using the strategies in (4)–(6), the A argument of the intransitive action nominal is marked in the same way as the A argument for transitive action nominals. In contrast, in ALL-COMPS intransitive ANCs (the strategy used in (7)), the A argument of the intransitive action nominal is marked in the same way as the P argument for transitive action nominals (with an *of* preposition).

The ANCs in (4)–(5) can be modified only by adverbs, while the ANCs in (6)–(7) can be modified only by adjectives (Malouf, 1998). The A can be dropped in both (4) and (5), while (6) requires an overt Poss or determiner, and (7) requires a determiner. For transitive verbs, the P is obligatory in (4)-(5), but optional in (6)-(7).

Moving on to coordination in ANCs, grammaticality judgements in this area are not the most clear cut, but I base the following discussion and the later implementation on my own judgments as a native speaker as well as judgments from Malouf (1998). Specifically, the sentences and corresponding grammatically judgments in (8) and (13) come from Malouf (1998) and the rest are my own.

As the below examples demonstrate, NP (8) and noun (9) coordination is acceptable in English [eng] between ANCs of the same type as well as between non-derived noun phrases and ANCs (10). Coordination between nouns and action nominals is acceptable for those action nominals that take either a determiner or a Poss (11) versus (12). NP coordination between ANCs with different external arguments (subject versus determiner/Poss) is prohibited (13).

(8) *Pat come-ing and Chris leave-ing bother-s me*
Pat come-NMLZ and Chris leave-NMLZ bother-3SG 1SG[ACC]
'Pat coming and Chris leaving bothers me.' [eng] (Malouf, 1998: 38)

(9) *Pat 's come-ing and leave-ing bother me*
Pat POSS come-NMLZ and leave-NMLZ bother 1SG[ACC]
'Pat's coming and leaving bother me.' [eng]

(10) *Pat 's smile-ing and the fresh air are pleasant*
Pat POSS smile-NMLZ and the fresh air be.PL pleasant
'Pat's smiling and the fresh air are pleasant.' [eng]

(11) *the smile-ing and fresh air are pleasant*
the smile-NMLZ and fresh air be.PL pleasant
'the smiling and fresh air are pleasant.' [eng]

(12) **the cat and read-ing book-s are pleasant*
the cat and read-NMLZ book-PL be.PL pleasant
'the cat and reading books are pleasant.' [eng]

(13) **Pat 's come-ing and Chris leave-ing bother-s me*
Pat POSS come-NMLZ and Chris leave-NMLZ bother-3SG 1SG[ACC]
'Pat's coming and Chris leaving bothers me.' [eng] (Malouf, 1998: 39)

To test this implementation, I construct a testsuite including examples of intransitive and transitive verbs across the different ANC types. Strings are included of ANCs both in *by* clausal modifier phrases, and in subject position. Ungrammatical strings are created by removing necessary arguments, using the wrong modifier (adjective versus adverb), or using finite clause marking (nominative case on subjects) within ANCs. I include 99 grammatical

stings and 65 ungrammatical strings. A few of the sentences are taken directly from Malouf (1998), but I construct the majority myself based on the information provided by both Malouf (1998) and Koptjevskaja-Tamm (1993).

Implementation

To implement the possessive strategies in English [eng] and following the approach in Nielsen (2018), I model the strategy in (1) as a possessor-first strategy where the possessor is marked by a non-affix marker (*'s*) and the strategy in (2) as a possessor-first non-affix pronominal strategy where the pronoun *his* takes third person and singular number, reflecting information about the possessor only. However, (3) cannot be modeled as a specifier strategy as the specifier position of the possessum *friend* is already occupied by a determiner. Since ANCs can only use specifier possessive strategies, this prevents ANCs from taking *of* prepositional phrase (PP) possessors. However, since NOMINAL and ALL-COMPS ANCs can still take *of* PPs, I define *of* as a semantically empty adposition on the Lexicon page. Nominalization lrts can then constrain their arguments to be marked with an of_sem FORM value as explained in section 5.1.1.

Moving on to ANCs, I define five nominalization strategies on the Nominalized Clauses webpage. All four of the strategies illustrated in (4)–(7) can act on both intransitive and transitive verbs, but POSS-ACC/NOMINAL intransitive ANCs are indistinguishable when they do not take either an adjectival or adverbial modifier. Thus, the ANC *Pat's sleeping* is ambiguous between a POSS-ACC and NOMINAL reading. This is further shown by the fact that the action nominal *sleeping* can take both an adverb *Pat's sleeping peacefully* (as expected for a POSS-ACC action nominal) and an adjective *Pat's peaceful sleeping* (as expected for a NOMINAL action nominal). I then create four nominalization strategies corresponding to the four ANC types present in English [eng] and specify that the ALT-SENT and ALL-COMPS strategies can act on both transitive and intransitive verbs. I restrict the POSS-ACC and NOMINAL strategies to only act on transitive verbs and create a fifth strategy to handle both POSS-ACC and NOMINAL intransitive ANCs. I define this

fifth strategy as a POSS-ACC strategy since every strategy requires a unique ANC type, although this strategy could also be defined as a NOMINAL strategy without it making a difference.

I specify that the ALT-SENT and transitive POSS-ACC strategies can be modified by adverbs, the ALL-COMPS and transitive NOMINAL strategies by adjectives, and the intransitive POSS-ACC/NOMINAL strategy by both. For the ALL-COMPS nominalization strategy, I require the P to be the first complement of transitive action nominals. To model the behavior of action nominals with Poss arguments and determiners, I constrain the transitive POSS-ACC and the intransitive POSS-ACC/NOMINAL strategies to take a mandatory syntactic possessor, the NOMINAL strategy to take a determiner whenever a Poss argument is not overt, and the ALL-COMPS strategy to obligatorily take a determiner. Requiring the A (the Poss argument) of a transitive POSS-ACC action nominal to be mandatory helps avoid spurious ambiguity in phrases such as *destroying the evidence*, which is parsed only as an ALT-SENT action nominal and not a POSS-ACC action nominal with a dropped argument. Likewise, requiring the intransitive POSS-ACC/NOMINAL nomnialization strategy to always appear with a Poss ensures that intransitive ANCs such as *The sleeping* are only parsed as ALL-COMPS ANCs. Finally, I allow all three of the POSS-ACC, NOMINAL and intransitive POSS-ACC/NOMINAL nominalization strategies to use either of the two defined specifier possessive strategies in the language.

Moving on to the Morphology page, I create a nominalization position class with five lrts (all with the orthographic form *-ing*) each marked by a different nominalization strategy. I constrain the subject of the lrt associated with the ALT-SENT strategy to have accusative case and the object to be mandatory ([OPT −]). I likewise require the transitive POSS-ACC lrt to have a mandatory complement. I constrain the complement of a NOMINAL action nominal to take the semantically empty *of* preposition, and specify the appropriate prepositions (*of* on the first and *by* on the second) on both the ALL-COMPS lrt's complements and require the second complement (the *by* complement) to be obligatory. Making the *by* PP obligatory avoids ambiguity with phrases such as *the destroying of the evidence* which

could be parsed as either NOMINAL or ALL-COMPS. Since the intransitive ALL-COMPS lrt is created automatically from the transitive lrt (see section 5.2.1), the single complement of an intransitive ALL-COMPS action nominal is also marked by an *of* preposition.

To implement the facts surrounding coordination as described above, I add an NP and noun coordination strategy which uses a feature resolution strategy.[3] This feature resolution strategy specifies that the NMZ value of the mother should be the same as that of the conjuncts when the conjuncts have the same NMZ value. Thus, two [NMZ −] conjuncts result in a [NMZ −] coordinated phrase. If the conjuncts do not have the same NMZ value then the overall phrase is labeled as [NMZ +]. I add a feature resolution strategy to avoid ambiguity between the version of *head-spec-phrase* which joins determiners and nouns (*noun-head-spec-phrase*) and the one that joins determiners and action nominals (*anc-head-spec-phrase*). Without a feature resolution strategy to specify what NMZ value the overall coordinated phrase should take, the phrase is underspecified for this value meaning it can use either of the two head-spec rules.

This coordination strategy correctly allows NP and noun coordination between ANCs of the same nominalization type as well as between ANCs and non-derived noun phrases. Since coordination in the Grammar Matrix requires both the conjuncts and the mother to have the same VAL features (Drellishak and Bender, 2005),[4] coordination between a non-derived noun and an action nominal which requires a subject (12) is correctly blocked.

Results

The grammar produced by the customization system for my English [eng] choices file achieved 100% coverage, 4.6% over-generation, and an average of 1.61 parses per sentence on this testsuite. The over-generation is due to coordination sentences of the type shown in

[3]Malouf (1998) mentions that coordinated ALT-SENT ANCs tend to trigger plural agreement while coordinated POSS-ACC ANCs tend to trigger singular agreement. However, he concludes that these coordination facts are not fully clear and seem to be part of a larger phenomena surrounding coordination. Due to this lack of clarity, number is left unspecified in coordinated phrases.

[4]`https://github.com/delph-in/docs/wiki/MatrixDoc_Coordination`

(13) as well as the ungrammatical string in (14).

(14) **The DA was shocked by destroy-ing the evidence*
The DA[NOM] was shocked by destroy-NMLZ the evidence[ACC]
'The DA was shocked by destroying the evidence.' [eng]

(13) is ungrammatical since an action nominal which takes a Poss (*Pat's coming*) is coordinating with an ANC that takes a subject (*Chris leaving*). However, the coordination strategy I implement cannot block this parse since, at the NP level, there is no way to distinguish these two different ANC types. Both ANCs are [NMZ +] and have the same VAL features as they are valence-saturated noun phrases. There is then nothing in the current analysis which can prevent two ANCs of these two types from coordinating. While the ungrammatically of (13) might be questioned, this is still a known short-coming of the analysis.

Moving on to (14), this sentence parses because ALT-SENT action nominals can generally drop their subjects as shown in (15).

(15) *destroy-ing the evidence is a crime*
destroy-NMLZ the evidence be.3SG a crime
'Destroying the evidence is a crime.' [eng]

Since action nominals are otherwise able to drop their subjects, the cause of the ungrammaticality of (14) is not fully clear and requires more data to elucidate. This type of over-generation then seems to be out of scope of the current thesis.[5]

6.2.2 Hixkaryana [hix]

Data/Description

The below description of Hixkaryana [hix] as well as the sentences I include in my test-suite for the language are based entirely on the grammar description in Derbyshire (1985).

[5]As one reader points out, sentences very similar in structure to (14) are grammatical with the one potential distinction that the agent of the action nominal can co-refer with the subject of the main clause. This further suggests that there are additional factors which complicate the analysis of (15).

Hixkaryana [hix] is a Carib language spoken by groups living on the Nhamundá and Mapuera rivers in Brazil (Derbyshire, 1985). The language has basic OSV word order and no determiners. In finite clauses, nouns are unmarked and the verb takes both a prefix encoding the person of the subject and object (if present) as well as a tense-aspect-number-mood suffix. The different possessive strategies in the language are shown in (16)–(17).

(16) *r-amo-ri*
1-hand-POSS
'My hand.' [hix] (Derbyshire, 1985: 5)

(17) *haname y-awo-ru*
haname 3.OVERT-mother's.brother-POSS
'Haname's uncle.' [hix] (Derbyshire, 1985: 5)

Possessed nouns are marked by a prefix marking the person of the possessor and a possessive suffix (16). For third person possessors, it is also possible for the possessum to be optionally preceded by a noun which serves as the possessor (17). As will be seen below, both of these strategies can be used in ANCs.

I show examples of both intransitive and transitive action nominals in (18)–(20) with the action nominal appearing in a *ke* 'because' or *me* 'DENOMINALIZER'[6] clausal modifier phrase.

(18) (ERG-POSS)

n-niki-yaha	*kaykusu*	*tuna*	*y-omoki-ni-ri*	*ke*
3-go.to.sleep-NPST	dog	water	3.OVERT-come-NMLZ-POSS	because

'The dog sleeps because it is raining.' [hix] constructed example based on (Derbyshire, 1985)

(19) (ERG-POSS)

teryewryero	*r-wanota-ni-ri*	*me*	*k-ewehi-yaha*
loudly	1-sing-NMLZ-POSS	DENOMLZR	1-take.a.bath-NPST

'I take a bath singing loudly.' [hix] (Derbyshire, 1985: 38)

[6] *me* 'DENOMINALIZER' is a postposition used to mark a noun phrase as an adjunct or as the complement of a copula (the overt copula can not take nominal complements in Hixkaryana [hix]) when no other postpostion can be used. It is then called a denominalizer, since it allows a noun phrase to appear in a position where it would otherwise be blocked from (Derbyshire, 1985).

(20) (ERG-POSS)

k-rata-no	*uro*	*biryekomo*	*wya*	*katu*	*y-tayma-ni-ri*	*ke*
1-wept-IMM.PST	1	boy	by	cat	3.OVERT-push-NMLZ-POSS	because

'I wept because the boy pushed the cat.' [hix] constructed example based on (Derbyshire, 1985)

A verb is nominalized through the addition of the suffix *-ni*. The A of an intransitive action nominal and the P of a transitive action nominal are both marked in the exact same way as a possessor in a possessive noun phrase. For transitive ANCs, the A is marked by the *wya* postposition. This meets the definition of an ERG-POSS ANC. Hixkaryana [hix] does not have a category corresponding to adjectives, but as seen in (19), the action nominal *r-wanota-ni-ri* '1-sing-NMLZ-POSS' can be modified by an adverb *teryewryero* 'loudly'.

The preferred word order in ANCs is A P action.nominal for transitive action nominals and A action.nominal for intransitive action nominals. The A of an action nominal can also sometimes appear after the action nominal (action.nominal A) often when an adjunct appears in the ANC or the ANC is more complex (Derbyshire, 1985). This can either mean appearing directly after the action nominal or being moved outside the subordinate clause entirely. Since it is not the baseline preferred word order, this type of variability is considered out of scope of the current library. I do, however, include an example with the nonpreferred ordering in the testsuite to indicate this limitation of the library.

The testsuite for Hixkaryana [hix] contains examples of transitive and intransitive action nominals appearing in different clausal modifier phrases as well as in subject position. Ungrammatical examples are created by removing necessary affixes, using finite morphology in ANCs, or using ANC-specific morphology in stand-alone clauses. There is a total of 11 grammatical examples and 22 ungrammatical examples. Aside from (19), taken directly from Derbyshire (1985), I construct all the examples in the testsuite based off of the description and examples in Derbyshire's description. Many of the constructed examples are simplified versions of sentences in Derbyshire (1985) where I remove certain phenomena that add complexity to the sentence but that are not directly related to ANCs such as discourse particles.

Implementation

Starting with the possessive strategies, I model the strategy in (16) as a pronominal possessive prefix, and the strategy in (17) as a possessive prefix on the possessum which agrees in person with the preceding possessor. I define both a possessive prefix and a possessive suffix position class on the Morphology page. The possessive prefix position class serves as input to the possessive suffix position class.

Since the P argument is marked as a Poss in transitive action nominals, I implement the sentences in (18)-(20) by creating an ERG-POSS nominalization strategy which acts on both intransitive and transitive verbs and can be modified by adverbs. I specify that this nominalization strategy has SOV (A P action.nominal) word order as opposed to the OVS word order used in finite clauses. I also allow this strategy to use either of the two defined possession strategies in the language. I then define a *-ni* lrt which takes this nominalization strategy on the Morphology page and require that the object be marked by the semantically empty *wya* adposition. This nominalization position class serves as input to the nominal possessive prefix position class meaning that the nominalized verb can take possessive prefixes (and suffixes) just like any other noun.

Results

The grammar produced by the customization system based on the Hixkaryana [hix] choices file achieves 91.7% coverage, 0% over-generation, and an average of 3 parses per sentence on the testsuite. The lack of coverage comes from the example involving the non-preferred word order in ANCs as discussed above.

6.2.3 Russian [rus]

Data/Description

The following description of Russian [rus] as well as the testuite I use to evaluate my implemented grammar are based off of Comrie (1976), Comrie and Thompson (1985), Koptjevskaja-Tamm (1993), and Timberlake (2004). Russian [rus] is an Indo-European language which

uses nominative-accusative case marking in finite clauses and has relatively free word order with SVO being the most neutral (Koptjevskaja-Tamm, 1993). In the present tense, finite verbs agree with the subject in person and number, and in the past tense, finite verbs agree with the subject in gender (in the singular) and number (Koptjevskaja-Tamm, 1993; Timberlake, 2004). The language lacks determiners and all nouns are divided into one of three genders (masculine, feminine, and neuter). In addition to nominative and accussative case, nouns can take one of four other case values: genitive, dative, instrumental, and prepositional.

Russian has two ways to mark a possessor in a noun phrase shown in (21)–(24).

(21) *sɐbak-ə* *malčik-a*
dog[FEM]-NOM.SG boy[MASC]-GEN.SG
'The boy's dog.' [rus] constructed example based on (Timberlake, 2004)

(22) *mɐja* *sɐbak-ə*
1SG.NOM.FEM.POSS dog[FEM]-NOM.SG
'My dog.' [rus] constructed example based on (Timberlake, 2004)

(23) *jevo* *sɐbak-ə*
3SG.MASC.POSS dog[FEM]-NOM.SG
'His dog.' [rus] constructed example based on (Timberlake, 2004)

(24) *ix* *sɐbak-ə*
3PL.POSS dog[FEM]-NOM.SG
'Their dog.' [rus] constructed example based on (Timberlake, 2004)

In the first strategy shown in (21) a non-pronoun third person possessor is marked with genitive case. The remaining three strategies involve pronominal possessors which are marked using possessive pronouns. First and second person possessive pronouns agree with the possessed noun in gender, number and case as shown in (22). The third person pronouns (feminine, masculine, neuter, and plural) are fixed and appear using the same form with all possessums (Timberlake, 2004). An example of the single masculine possessive pronoun *jevo* is shown in (23) and of the plural possessive pronoun *ix* in (24). Possessors marked by

genitive case tend to appear after the possessum while possessive pronouns tend to occur before the possessum although it seems both of these orders can be switched for stylistic reasons (Timberlake, 2004; Comrie and Thompson, 1985).

Moving on to ANCs, action nominals in Russian [rus] can be formed through adding a nominalization suffix *-ij* to the verb root (Timberlake, 2004). Action nominals take neuter gender and verbal categories such as aspect are not distinguished.[7] Action nominals inflect for features such as number and case as would any other neuter noun.

Intransitive action nominals mark the A using any of the possessive strategies discussed above. An example of a genitive-marked A is shown in (25).

(25) (ERG-POSS)

grom	*prerva-l-Ø*	*mirn-i*
thunder[MASC.SG]	interrupt[PFV]-PST-SG.MASC	peaceful-PL.ACC

razmyš-enij-a	*malčik-a*
meditate-NMLZ.NEUT-PL.ACC	boy[MASC]-GEN.SG

'Thunder interrupted the boy's peaceful meditation.' [rus] constructed example based on (Koptjevskaja-Tamm, 1993: 26)

Transitive action nominals can either mark the P as a Poss and the A using instrumental case (26), or mark the A with a possessive pronoun and the P using genitive case (27). The first strategy is an ERG-POSS ANC while the second is a NOMINAL ANC since both arguments are marked as a Poss. Any of the arguments can be dropped across all the ANC types.

(26) (ERG-POSS)

grom	*prerva-l-Ø*	*pakare-nij-e*
thunder[MASC.SG]	interrupt[PFV]-PST-SG.MASC	conquest-NMLZ.NEUT-ACC.SG

everest-a	*alexandr-om*
everest[MASC]-GEN.SG	alexander[MASC]-INSTR.SG

[7] In Russian [rus] it is possible for there to be a perfective/imperfective verb pair with only one corresponding action nominal (Comrie and Thompson, 1985). This action nominal is ambiguous between the perfective/imperfective aspect. My current implementation cannot handle this kind of ambiguity.

'Thunder interrupted the conquest of Everest by Alexander.' [rus] constructed example based on (Koptjevskaja-Tamm, 1993: 26)

(27) (NOMINAL)
grom *prerva-l-Ø* *mojo*
thunder[MASC.SG] interrupt[PFV]-PST-SG.MASC 1SG.NEUT.ACC

pakare-nij-e *everest-a*
conquest-NMLZ.NEUT-ACC.SG everest[MASC]-GEN.SG
'Thunder interrupted the conquest of Everest by me.' [rus] constructed example based on (Koptjevskaja-Tamm, 1993: 26) and (Timberlake, 2004: 119)

As discussed in section 4.5.3, I used Russian to flesh out the single-possessor analysis, as whenever a single argument of a action nominal appears it must be marked as a Poss. This single argument is ambiguous between an agent-like and a patient-like reading as shown in the repeated (28).

(28) *ʧe-ni-e* *puʃkin-a*
read-NMLZ.NEUT-NOM.SG Pushkin[MASC]-SG.GEN
'Pushkin's reading.' (agent-like reading) or 'The reading of Pushkin.' (patient-like reading) [rus] (Comrie, 1976)

One final note is that action nominals created from psychological predicates show a distinct pattern. With these action nominals the A is marked as a Poss while the P is marked with an preposition such as *k* 'towards' which specifies dative case on its complement. Notably, This *k* PP is not used with the underlying verb (Comrie and Thompson, 1985; Koptjevskaja-Tamm, 1993; Timberlake, 2004). (29) provides an example.

(29) *uvaža-enij-e* *petrov-a* *k*
respect-NMLZ.NEUT-NOM.SG petrov[MASC]-GEN.SG towards
ivanov-u *izmeni-l-o* *stran-u*
ivanov[MASC]-DAT.SG change[PFV]-PST-SG.NEUT country[FEM]-ACC.SG
'Petrov's respect towards Ivanov changed the country.' [rus] constructed example based on (Koptjevskaja-Tamm, 1993: 26, 29) and (Timberlake, 2004: 220)

With respect to adjective versus adverb modification, Russian ANCs can only be modified by adjectives (Comrie and Thompson, 1985).

The Russian testsuite contains examples of all the mentioned ANCs in subject and clausal complement positions including examples where ANCs omit one or all of their arguments, take adjectival modifiers, and use different possessive strategies. Ungrammatical examples are constructed by using an adverb instead of an adjective in an ANC, or using the wrong form of marking for one of the arguments in an ANC. The testsuite has 100 grammatical examples and 2o ungrammatical examples. I either take sentences directly from Koptjevskaja-Tamm (1993) or construct additional sentences based off of the examples in Koptjevskaja-Tamm (1993) and the grammatical information provided by Comrie (1976), Comrie and Thompson (1985), Koptjevskaja-Tamm (1993), and Timberlake (2004).

Implementation

To implement the possessive strategy in (21), I define a juxtaposition possessive strategy where the possessor is required to take genitive case. To handle the various possessive pronouns, I define two different pronominal non-affix strategies. The first requires agreement between the relevant features for first and second person pronouns, while the second does not and is used to define the third person possessive pronouns. All the defined possessive strategies can use either possessor-possessum or possessum-possessor word order.

To capture the facts regarding ANCs, I define two nominalization strategies. The first is an ERG-POSS strategy which acts on both transitive and intransitive verbs, is modified by only adjectives and uses the single-possessor analysis. The second strategy is a NOMINAL strategy, acts only on transitive verbs, is modified by adjectives, and requires a mandatory Poss argument. Both nominalization strategies are permitted to use any of the defined possessive strategies.

I then create a nominalization position class which contains three nominalization lrts. One lrt uses the ERG-POSS nominalization strategy while the other two use the NOMINAL nominalization strategy. The nominalization position class serves as input to a nominal

position class for case and number affixes, so that action nominals can take the necessary nominal morphology. Each of the lrts in the nominalization position class takes a neuter gender feature modeling the fact that action nominals decline like neuter nouns.

The ERG-POSS lrt further constrains its object to take instrumental case. Since the ERG-POSS nominalization strategy acts on both intransitive and transitive verbs, this lrt covers all intransitive ANCs as well as all transitive ANCs where the A is marked with instrumental case. This lrt is also responsible for handling all cases where only a single argument appears in a transitive ANC, which is covered by the single-possessor analysis. Finally, this lrt handles the case where an action nominal has no overt arguments. since neither the Poss argument or the object are constrained to be mandatory.

Switching to the two NOMINAL lrts, the first of the two is used to model the sentences where the A is marked as a possessive pronoun and the P is marked with genitive case. As mentioned in section 4.2.2, since an ANC can only contain one true syntactic possessor, in ANCs where both arguments are marked as a Poss, only the A argument makes use of the analysis provided by the adnominal possession library. In Russian [rus] this means that the A will be marked as a possessive pronoun, while the P is simply treated as a genitive complement. The NOMINAL lrt then requires its object to take genitive case. This lrt also constrains its object to be mandatory ([OPT −]). This avoids ambiguity with the ERG-POSS lrt, since the NOMINAL lrt will only be used when both the A and P arguments are overt.

The last NOMINAL lrt is used exclusively by pyschological (psych) predicates. This is accomplished by creating a separate verb type on the Lexicon page for pysch predicates. This verb type is the only possible input to the psych predicate nominalization lrt. Likewise, this verb type is barred from the other two nominalization lrts. On the Lexicon page, I also define a semantically empty adposition *k* 'towards' which requires its complement to take dative case. The psych predicate nominalization lrt then constrains its object to have a k_sem FORM value. For the same reasons as the other NOMINAL lrt, the object is also made to be obligatory. Right now this blocks the possibility of a *k* marked argument appearing on

its own (without the possessive pronoun A) with the action nominal. Its not fully clear if this is the correct choice, but more data is needed to be certain.

It also needs to be mentioned that treating *k* 'towards' as semantically empty is not an ideal semantic analysis. As evident from the gloss, *k* 'towards' has the meaning of *towards*. However, semantically contentful adpositions are primarily treated as heading phrases which serve as modifiers and not complements by the Grammar Matrix. This means that the *k* PP would both have too free of a distribution (be capable of modifying elements outside of psych predicate action nominals) as well as have the incorrect semantics. To the latter point, the current analysis assumes the A and P arguments of action nominals will always be linked to either the subject, complement, or specifier valence positions. Thus, if the *k* 'towards' PP is treated as a modifier (neither a subject, complement, nor specifier), it would not be interpreted as the semantic patient of the action nominal.

What would then likely be ideal for a preposition such as *k* 'towards' is to have the option of treating a semantically contentful prepositional phrase as a complement. This would allow more control over both the semantics (including the ability to represent *towards* in the semantics) and the distribution of the *k* marked phrase. However, such an addition is out of scope for this thesis and is left for future work.

Results

The grammar produced by the customization system based on the Russian [rus] choices file achieves 100% coverage, 0% over-generation, and 1.73 average parses per sentence over the testsuite.

6.2.4 Korean [kor]

Data/Description

I base my description of Korean [kor] as well as the testsuite I construct to test my implemented grammar on the work in Koptjevskaja-Tamm (1993), Chang (1996), Kim and Yang (2004), and Yoon (2022). Korean [kor] is Koreanic language with SOV word order

and which uses nominative-accusative case-marking[8] in finite clauses. The language has determiners and uses det-noun word order. In possessive noun phrases, the possessor appears prior to the possessum and is marked with genitive case (30).

(30) *mia-uy* *phal*
mia-GEN arm
'Mia's arm.' [kor] constructed example based on (Chang, 1996: 79)

Korean [kor] has two main nominalizer suffixes (*-ki* and *-um*). ANCs appearing in typical nominal positions such as subject and object position take the expected case-marking (Chang, 1996). Action nominals resulting from both nominalization suffixes mark their arguments in the same way as in finite clauses, with the A taking nominative case and the P taking accusative case. Examples of both nominalizers are shown in (31) and (32).

(31) (SENT)
john-i *ecey* *ku* *chayk-ul* *ilk-ess-um-i* *nollap-ta*
john-NOM yesterday that book-ACC read-PST-NMLZ-NOM to.be.surprising-SE
'John's having read the book yesterday is surprising.' [kor] constructed example based on (Kim and Yang, 2004)

(32) (SENT)
cheli-ka *pap-ul* *mek-ko* *iss-Ø-ki-lul* *palan-ta*
cheli-NOM meal-ACC eat-COMP be-PRS-NMLZ-ACC hope-SE
'I hope that Cehli is taking his meals.' [kor] constructed example based on (Yoon, 2022: 432)

This type of ANC falls under the SENT nominalization type. For the *-ki* suffix this is the only possibility. For the *-um* suffix, it is marginally acceptable for the A to be marked with genitive case as a possessor would be (33) (Yoon, 2022). This type of nominalization is classified as a POSS-ACC ANC.

[8]Following Kim and Yang (2004) Korean [kor] case-markers are treated as suffixes and not stand-alone particles.

(33) *(POSS-ACC)*
hananim-uy wuli-lul salangha-si-Ø-um-ul molu-ni
god-GEN we-ACC love-SBJ.HON-PRS-NMLZ-ACC not.know-Q
'Do you not know that God loves us?' [kor] constructed example based on (Yoon, 2022: 432)

Across all ANCs, only adverb modification is permitted, and no action nominal (even POSS-ACC action nominals) can appear with determiners (Yoon, 2022). Aside from the potential of a genitive-marked A, Korean ANCs are nearly identical to finite clauses including being able to take tense and honorific suffixes as shown throughout (31)–(33).[9]

The Korean testsuite contains examples of the different types of Korean [kor] ANCs in subject, clausal complement, and clausal modifer positions using both intransitive and transitive action nominals, as well as including examples with adverbial modification. Ungrammatical examples are constructed by using the wrong morphology on either action nominals or non-nominalized verbs or using an inappropriate modifier (a determiner or an adjective). The testsuite has 16 grammatical examples and 27 ungrammatical examples. I either take sentences directly from or else modify examples in Chang (1996), Kim and Yang (2004), and Yoon (2022).

Implementation

Moving on to the implementation, I model the possessive strategy in (30) as a possessor-first juxtaposition strategy where the possessor is required to have genitive case. For the ANCs, I define both a SENT and a POSS-ACC nominalization strategy. Both strategies act on transitive and intransitive verbs and can be modified only by adverbs. To prevent ambiguity between the two strategies when the A argument is dropped the POSS-ACC strategy is required to appear with a Poss (the one defined possessive strategy in the language).

[9]The other main difference between ANCs and finite clauses is that ANCs can not contain any topic-marking which is generally true of all subordinate clauses (Koptjevskaja-Tamm, 1993). Since this fact is general to all subordinate clauses and is not specific to ANCs, the interaction between topic-marking and subordinate clauses is considered out of scope for this thesis.

On the Morphology page, I define a nominalization position class with two lrts. This position class serves as input to a defined nominal case position class allowing action nominals to take case marking. The first lrt in the nominalization position class uses the SENT nominaliztion strategy and has two orthographic forms (*-um* and *-ki*). The second uses the POSS-ACC nominalization strategy and has only a single othograhic form (*-um*). This ensures only the *um* nominalizer will be able to create POSS-ACC ANCs.

Results

The grammar produced by the customization system on the Korean [kor] choices file achieves 100% coverage, 0% over-generation, and an average of 1.19 parses per sentence on the testsuite.

6.2.5 Summary

The results across all the illustrative languages are shown below in table 6.4. All the implemented grammars perform reasonably well with the English [eng] grammar over-generating and the Hixkaryana [hix] grammar having slightly limited coverage for the reasons described above.

Table 6.4: Illustrative language test results

	# gramm-atical	# ungramm-atical	% coverage	% over-generation	avg-analysis-#
eng	99	65	100	4.6	1.61
hix	11	22	91.7	0	3.00
rus	100	20	100	0	1.73
kor	16	27	100	0	1.19

6.3 Held-out Languages

In order to evaluate the coverage of the library on unseen data, I select five languages at random to serve as held-out languages. I choose these languages from the collection of 168 different languages which have a WALS feature value for action nominal constructions (Koptjevskaja-Tamm, 2013). The possible values for the action nominal construction feature include the 4 main types (SENT, POSS-ACC, ERG-POSS, NOMINAL), as well as several more minor types, and an option to state that a language does not contain an action nominal class. I discard languages if they either come from the same language family as an already selected held-out[10] language, or if their WALS feature value indicates they do not have at least one ANC type that the library is built to handle.[11] The five held-out languages are: Wayana [way] (Cariban), Maltese [mit] (Afro-Asiatic), Dutch [nld] (Indo-European), Lango [laj] (Eastern Sudanic), Finnish [fin] (Uralic). The same general implementation decisions I make for all the illustrative languages (including **nominalized_rel** in the semantics for SENT/ALT-SENT action nominals and using the both semantic option for POSS-ACC/ERG-POSS/NOMINAL action nominals) also apply for all the held-out languages. Additionally, I implement all described possessive strategies in all the below languages even if they are not used in ANCs to test for any poor interactions between the nominalized clauses and adnominal possession libraries. In sections 6.3.1–6.3.5, I discuss the data, implementation and testsuite results (coverage, over-generation, and average number of parses per sentence) for each language. Section 6.4 describes a few implementation changes I make after evaluation, as well as a few areas of the analysis that require additional work.

---

[10] I do allow language family repetition between the illustrative and held-out languages so that English [eng], Russian [rus], and Dutch [nld] are all Indo-European languages and Hixkaryana [hix] and Wayana [way] are both Cariban) languages.

[11] These correspond to the minor ANC types in Koptjevskaja-Tamm's hierachy, which are out of scope for this thesis. I thus skip any language that only contains these minor ANC types (Koptjevskaja-Tamm, 1993).

6.3.1 Wayana [way]

Data/Description

The following description of Wayana [way] as well as the sentences I use in my testsuite are based off of the description of the language in da Silva Tavares (2005). Wayana [way] is a Cariban language spoken in Wayana villages located on the borders of Surinam, French Guyana, and Brazil (da Silva Tavares, 2005). It has relatively free word order and no determiners. The language has two distinct kinds of finite clauses: (1) set I clauses and (2) t-V-(h)e clauses (da Silva Tavares, 2005). In a set I clause, verbs take a personal prefix marking the subject and/or object of the verb as well as a suffix marking a combination of tense, aspect, number or mood. Different personal prefixes are used in the following four conditions (1) direct: a non-third-person subject acts on a third person object (2) inverse: a third person subject acts on a non-third person object, (3) local: a non-third person subject acts on a non-third person object, and (4) both arguments are in the third person. The overall configuration can be analyzed as a direct-inverse system where the first and second person outrank the third. Intransitive set I clauses are split into two groups: one which marks the subject with the same prefix used to mark the subject in transitive verbs (S_a intransitives) and one which marks the subject with the same prefix used to mark the object in transitive verbs (S_o intransitives).

In contrast, in t-V-(h)e clauses, verbs take a *t-* prefix and a *-(h)e* suffix. The subject of intransitive verbs and the object of transitive verbs is unmarked while the subject of transitive verbs takes the ergative marker *ja*. In t-V-(h)e clauses, verbs cannot take either the personal prefixes or the tense-aspect-number-mood suffixes. The factors which determine whether a set I or t-V-(h)e) clause is used seem to be primarily pragmatic as all verb stems can use both.

A possessor can be marked with a pronominal prefix on the possessum (34) or an overt third person possessor preceding the possessum (35). The co-occurrence of a possessive prefix and an overt possessor is prohibited. In both cases, the possessum also takes a possessive

suffix.

(34) *i-malija-n*
3-knife-POSS
'His knife.' [way] (da Silva Tavares, 2005: 121)

(35) *nila malija-n*
nila knife-POSS
'Nila's knife.' [way] (da Silva Tavares, 2005: 121)

Wayana [way] has two different action nominalizers. A null nominalizer (36) and the suffix *-top(o)* (37).

(36) (ERG-POSS)
i-kilili-tomo *ekale-Ø-li* *ke* *w-umxky-Ø*
1-thing-POSS-COLL give-NMLZ-POSS because $1S_a$-come-REC.PST
'I came because they were giving away my things.' [way] constructed based on (da Silva Tavares, 2005: 455)

(37) (ERG-POSS)
w-eha-Ø *tuwale* *j-ulu-topo-npili-Ø* *kuni-tomo* *ja*
1-be-REC.PST knowingly 1-talk.to-NMLZ-DVL-POSS grandmother-COLL OBLAGT
'I was knowledgeable of the story my grandmothers told to me.'[12] [way] constructed based on (da Silva Tavares, 2005: 337)

For both suffixes, the A of intransitive verbs and the P of transitive verbs is marked as a Poss while the A of transitive verbs is marked by the postposition *ja*. This description fits under the ERG-POSS nominalization type. No overt arguments appear to be necessary. Wayana [way] has an adverb class, but no adjectives, and the documentation that I had access to did not specify whether adverbs can modify action nominals.

The Wayana [way] testsuite contains examples of both nominalizers appearing in different positions (subject, object, adverbial phrase). I construct ungrammatical examples by either

[12]The devaluative suffix *-npili* interacts with the nominal root to give meanings such as 'old' or 'spoiled' (da Silva Tavares, 2005). In nominalized verbs, the devaluative suffix adds the meaning that the event has already taken place. I do not attempt to capture the nuanced semantics of this suffix, instead I create a new boolean semantic feature DEVALUATION for nominal categories (a feature of *ref-ind* indices). A devaluative suffix sets this new feature to be [DEVALUATION +].

leaving out necessary morphology or using incorrect marking on the action nominal or its arguments. The testsuite contains 12 grammatical examples and 30 ungrammatical examples. I take and construct sentences based on the examples and grammatical description in da Silva Tavares (2005). The majority of the sentences in the testsuite are constructed.

Implementation

I implement[13] (35) as a possessor-first strategy with a null[14] possessive prefix on the possessum, and (34) as a pronominal possessor prefix. The position class containing all the possessive prefixes serves as the only input to a position class containing the possessive suffixes.

To implement ANCs, I create an ERG-POSS nominalization strategy which can act on both intransitive and transitive verbs and can use either of the two possessive strategies in the language. I then create a position class with two lrts, one for each nominalization suffix. Both of these lrts use the defined ERG-POSS strategy and require their object to be marked with a semantically empty *ja* postposition.[15] I also mark these lrts with the third person feature so that action nominals are not underspecified for person. This ensures that ANCs will not appear in positions reserved for first or second person pronouns. This nominalization position serves as one input to an obligatory case position class which feeds into the possessive

[13]I attempted to implement the Wayana [way] test grammar using the direct-inverse library (Drellishak, 2009). However, this revealed bad interactions between the direct-inverse library, the adnominal possession library, and the argument optionality library. These interactions exist even in the original code base my work is adding to. Due to this, I instead implement set I clauses by specifying the person marking of both the subject and the object on all transitive set I prefixes. This captures the syntactic facts though does not acknowledge the privileged position of non-third person pronouns.

[14]I implement this strategy as a null prefix since, in the Grammar Matrix, only one affix on a noun can be marked as possessive. I choose to always mark the prefix (either a pronominal possessive prefix or this null prefix) as possessive and then require nouns with a possessive prefix to take a suffix such as *-n* 'POSS'.

[15]As mentioned above, *ja* is also used to mark ergative case in t-V-(h)e finite clauses. I currently implement the one in finite clauses as a case-marking adposition and the one in ANCs as a semantically empty adposition (without case-marking), but it is not clear if this is fully necessary. I separate the two since da Silva Tavares (2005) glosses *ja* differently when its marks the subject in an t-V-(h)e clause versus when it marks the A in an ANC, however it would also be possible to analyze *ja* as an ergative case marker in both environments.

position classes. This allows action nominals to take possessive morphology as is necessary in both (36) and (37) where the action nominal serves as a syntactic possessum.

Results

The implemented grammar produced by the customization system on my Wayana [way] choices file achieves 100% coverage and 0% over-generation on the testsuite. The grammar has a relatively high level of ambiguity with 9.25 analyses on average per sentence. For the most part these are due to reasons unrelated[16] to the nominalized clauses library, however, one source of ambiguity is due to a bug in the implementation. Clausal subordinators such as *ke* 'because' in (37) are treated by the Grammar Matrix as adpositions. However, I did not include them in the set of adpositions that receive FORM values whenever a semantically empty adposition is defined. This means that *ke* 'because' is underspecified for FORM and that a nominalized verb which requires a *ja_sem* FORM value can either correctly take the semantically empty *ja* postposition or incorrectly take the underspecified for FORM *ke* adposition. Assigning a FORM value to *ke* reduces the average number of analyses per sentence to 8.25. As I discuss in section 6.4, I fix this bug in the library post-evaluation.

6.3.2 Maltese [mit]

Data/Description

The below description of Maltese [mit] and the sentences in my Maltese [mit] testsuite are fully based on Azzopardi-Alexander and Borg (1996). Maltese [mit] is an Afro-Asiatic language spoken in the Maltese Archipelago located near the center of the Mediterranean (Azzopardi-Alexander and Borg, 1996). It has neutral SVO word order in finite clauses and det-noun[17] word order in noun phrases (Azzopardi-Alexander and Borg, 1996). Verbs in

[16]Much of this unrelated ambiguity is due to various rules in the grammar being underspecified for case.

[17]Azzopardi-Alexander and Borg (1996) describe the definite article in Maltese [mit] as a clitic which attaches to a noun and creates a unit with it. However, for the sake of implementation, I treat the definite article as a completely separate word. This is equivalent to treating the definite article as a true clitic (a syntactically independent word). The fact that I implement the definite article as a separate word is also reflected in how I gloss all the Maltese [mit] examples.

finite clauses inflect for the person, number and gender (in the third person singular) of the subject. Nouns take one of either masculine or feminine gender.

Maltese [mit] uses templatic morphology with action nominals being one form of the verb. In dealing with such morphology, the Grammar Matrix assumes a separate morphophonological analyzer such that the strings passed to the grammar are sequences of morphemes (Bender and Good, 2005). I linearize the morphology for implementation by having the verb stem consist of just the root consonants of a verb and templates being represented instead as suffixes consisting of vowels.[18]

Moving on to possessive strategies, adnominal possession can be marked through juxtaposition (38), adding a pronominal possessor suffix to the possessum (39), or marking the possessor with the preposition *ta* 'of' (40). In both (38) and (40), the possessum precedes the possessor.

(38) *kelmet il general*
word the general
'The general's word.' [mit] (Azzopardi-Alexander and Borg, 1996: 112)

(39) *kelmt-u*
word-3SG.MASC.POSS
'His word.' [mit] (Azzopardi-Alexander and Borg, 1996: 76)

(40) *il mart ta pietru*
the woman of peter
'Peter's wife.' [mit] (Azzopardi-Alexander and Borg, 1996: 14)

Turning to action nominals, the A of intransitive verbs and the P of transitive verbs is marked with the *ta* 'of' preposition and the A of transitive verbs is marked with the *minn* 'from' preposition. This results in a word order of action.nominal P A which opposes the SVO order of finite clauses. Both arguments can be dropped. As shown in (41), the action nominal *tfigh* 'throwing' can appear with all three of an A, P and a determiner.

[18]I do not show this lineratization in the examples I give for Maltese [mit] for the sake of clarity.

(41) (ALL-COMPS)

it	*tfigh*	*ta*	*l*	*gebla*	*minn*	*pietru*	*holog*
the	throw\NMLZ	of	the	stone	from	peter	created\3SG.MASC

paniku	*kbir*
panic	big

'Peter's throwing of the stone brought about great panic.' [mit]
(Azzopardi-Alexander and Borg, 1996: 12)

Azzopardi-Alexander and Borg (1996) state that adverbials can appear in nominalized clauses, but only provide an example with a PP. It is then unclear if ANCs can also be modified by adjectives or adverbs. One final fact to model is that Maltese [mit] allows non-derived noun phrases to coordinate with ANCs.

The Maltese [mit] testsuite contains examples of both intransitive and transitive ANCs in subject and clausal modifier positions. I also include an example involving an ANC coordinating with a noun phrase. I construct ungrammatical examples by using the incorrect word order in ANCs or the incorrect marking of either the action nominal or its arguments. The testsuite has 12 grammatical examples and 8 ungrammatical examples. I include and construct sentences based on the examples in Azzopardi-Alexander and Borg (1996).

Implementation

I implement (38) as a possessum-first juxtaposition strategy, (39) as a pronominal suffix, and (40) as a possessum-first modifier strategy where a non-affix marks the possessor. I implement (40) as a modifier strategy since the possessum can take both a determiner and a possessor. Given that an action nominal can take a determiner along with both A and P arguments, the only possible analysis is to treat Maltese [mit] nominalized clauses as ALL-COMPS ANCs. I create an ALL-COMPS nominalization strategy with an accompanying lrt. The ALL-COMPS strategy acts on both intransitive and transitive verbs and requires the P argument to be the first complement of a transitive action nominal. Since the resources I use do not specify whether action nominals can be modified by adjectives or adverbs, I implement the strategy as not being able to be modified by either.

I also define a semantically empty preposition for both *ta* 'of' and *minn* 'from' and require the ALL-COMPS lrt to mark its first complement with *ta* 'of' and its second with *minn* 'from'. The library then automatically creates an intransitive lrt which requires the only complement of the action nominal to be marked with *ta* 'of'. While ALL-COMPS strategies do not allow for word order changes, since Maltese [mit] has VO word order and the transitive action nominal has two complements, the correct ordering is still achieved. To model coordination, I define an NP coordination strategy. This allows any ANC to coordinate with any non-derived noun phrase. Since coordination occurs at the NP level, there is no need to use an NMZ feature resolution pattern to remove ambiguity as I did above in English [eng]

Results

The implemented grammar produced by the customization system based on my Maltese [mit] choices file achieves 100% coverage, 0% over-generation, and 3.25 average parses per sentence on the testsuite. This testsuite did reveal one bug with the library that led to an increase in ambiguity. The bug is that action nominals can still take possessive modifiers (nouns marked with the possessive preposition *ta* 'of'). Thus, each time a sentence has an action nominal followed by a *ta* 'of' phrase, there is one parse where the action nominal takes a semantically empty *ta* 'of' complement PP, and additional parses where the action nominal is modified by the possessive *ta* 'of' modifier PP. After I specify that the MOD value of the possessive *ta* preposition should be [NMZ −], the average number of analyses drops from 3.25 to 1.58. This is an oversight in implementation, since I designed the analysis used by the library with only specifier possessors in mind. As discussed in section 6.4, I add this fix more generally to the library so that all possessive modifiers modify [NMZ −] lexical items (in languages with defined nominalization strategies) which blocks them from occurring with action nominals.

6.3.3 Dutch [nld]

Data/Description

The following description of Dutch [nld] as well as all the sentences in my testsuite for the language are based on Looyenga (1992), Shetter and Ham (2007), and (Havranová, 2020). Dutch [nld] is an Indo-European language spoken in the Netherlands and in Flanders (Shetter and Ham, 2007). It has V2 word order in finite main clauses, verb-final word order in subordinate clauses and det-noun word order in noun phrases. Nouns are divided into two genders: common (covers what was historically masculine and feminine) and neuter. In the singular, determiners agree with their nouns in gender. Verbs in finite clauses agree with the subject in person and number.

A possessor can be marked using a possessive pronoun (42), the *van* 'of' preposition (43), or a possessive *s* marker (44).

(42) *mijn* *gras*
1SG.POSS grass[NEUT.SG]
'My grass.' [nld] constructed based on (Shetter and Ham, 2007)

(43) *de* *ram-en* *van* *het* *huis*
the window[NEUT]-PL of the[NEUT.SG] house[NEUT.SG]
'The windows of the house.' [nld] (Shetter and Ham, 2007: 69)

(44) *jan* *s* *moeder*
jan[CG.SG] POSS mother[CG.SG]
'Jan's mother.' [nld] constructed based on (Shetter and Ham, 2007)

Moving on to action nominals, there are three different constructions in Dutch [nld] that can be analyzed as ANCs: (1) determined nominal infinitives, (2) *-ing* nominalized verbs, and (3) *ge-* nominalized verbs (Havranová, 2020).[19] Nominal infinitives have the same form as infinitives (taking the *-en* suffix), while *-ing* action nominals are formed with the

[19]Dutch [nld] also has bare nominal infinitives which do not take determiners (Havranová, 2020). I exclude these from implementation since it is not clear if it is possible to simultaneously express both arguments of transitive verbs, which is one of the library's defining criteria for ANCs.

nominalizing suffix *-ing* and *ge-* action nomianls with the nominalizing prefix *ge-*. All three nominalizations can take determiners (Havranová, 2020). Additionally, all three ANCs can coordinate with each other and regular noun phrases. This holds true for NP coordination and the resources I use do not state what the facts are at the noun level (whether an action nominal can coordinate with a noun).

The relevant characteristics for each of the three types of action nominal are listed below under their respective names. Each section includes: (1) the gender of the resulting action nominal, (2) whether the action nominal can be modified by adjectives, adverbs, or both, and (3) the marking of the A and P in intransitive and transitive action nominals. In terms of the third category, possessive *s* refers to an a argument marked by the possessive *s* marker (44), possessive pronoun to an argument marked as one (42), and *van*/*door* phrases to arguments marked by these two prepositions. If an argument can be marked in multiple different ways, I list each of the options. In the explanations that follow, I refer to both the possessive *s* and the possessive pronoun marking strategies as possessive strategies and marking with a *van* or *door* preposition as such.

Determined nominal infinitives

Gender: Neuter

adj/adv modification: allow both with a set order (adj adv)

A of intransitives: possessive *s*, possessive pronoun, *van*-phrase

A of transitives: possessive *s*, possessive pronoun, *door*-phrase

P: *van*-phrase

The different marking strategies of the A results in two different ANC possibilities. If the A is marked with one of the possessive strategies then it results in a NOMINAL ANC (45). If the A is marked with the *door* 'by' preposition then it results in an ALL-COMPS ANC since the action nominal can appear with both P and A arguments as well as a determiner (46). As listed above, determined nominal infinitives can be modified by both adverbs and adjectives. (47) further demonstrates that this includes taking both kinds of modifiers simultaneously. However, a sentence such as (47) is only grammatical if the adjective *irritante* 'irritating'

precedes the adverb *langzaam* ‘slowly’.

(45) (NOMINAL)
jan *s* *verzamel-en* *van* *postzegel-s* *is*
john[CG.SG] POSS collect-NMLZ.NEUT.SG of stamp[CG]-PL be.3SG

tijdrovend
time-consuming
‘John’s collecting of stamps is time-consuming.’ [nld] (Havranová, 2020: 60)

(46) (ALL-COMPS)
het *verzamel-en* *van* *postzegel-s* *door* *jan*
the[NEUT.SG] collect-NMLZ.NEUT.SG of stamp[CG]-PL by john[CG.SG]

is *tijdrovend*
be.3SG time-consuming
‘John’s collecting of stamps is time-consuming.’ [nld] (Havranová, 2020: 60)

(47) (adj/adv modification)

dat *irritante* *langzaam* *uitblaz-en* *van* *de*
that[NEUT.SG] irritating slowly puff.out-NMLZ.NEUT.SG of the
rook *is* *moelijk*
smoke[CG.SG] be.3SG difficult
‘The irritating slowly puffing out of the smoke is difficult.’ [nld] constructed based on (Looyenga, 1992: 175)

There is also another possible although more marked construction for determined nominal infinitives where the A of a transitive action nominal takes the *van* preposition and the P is left bare and appears before the nominal infinitive (48). I will refer to this construction as the bare argument infinitive construction.

(48) (Bare argument infinitive construction)

het *postzegel-s* *verzamel-en* *van* *jan* *is*
the[NEUT.SG] stamp[CG]-PL collect-NMLZ.NEUT.SG of john[CG.SG] be.3SG
tijdrovend
time-consuming
‘John’s collecting stamps is time-consuming.’ [nld] (Havranová, 2020: 61)

As will be discussed in the results section, the current library cannot handle a sentence such as (48).

***-ing* action nominals**

Gender: Common

adj/adv modification: adjectives

A: possessive *s*, possessive pronoun, *door*-phrase

P: possessive *s*, possessive pronoun, *van*-phrase

Additional constraints: exclusive to transitive verbs

Unlike either nominal infinitives or *ge-* action nominals to be discussed next, *-ing* action nominals can only be formed from transitive verbs. The above marking options result in three different types of ANCs. The ANC is NOMINAL if the A is marked by a possessive strategy and the P takes a *van* 'of' preposition (49). The ANC is ALL-COMPS if the A takes the *door* 'by' preposition and the P takes the *van* 'of' preposition (50). Finally, the ANC is ERG-POSS if the A is marked with the *door* 'by' preposition and the P takes possessive marking (51).[20]

(49) (NOMINAL)

jan	*s*	*behandel-ing*	*van*	*de*	*patient*
john[CG.SG]	POSS	treat-NMLZ.CG.SG	of	the[CG.SG]	patient[CG.SG]

was	*successvol*
be.SG.PST	successful

'John's treatment of the patient was successful.' [nld] (Havranová, 2020: 26)

(50) (ALL-COMPS)

de	*behandel-ing*	*van*	*de*	*patient*	*door*
the[CG.SG]	treat-NMLZ.CG.SG	of	the[CG.SG]	patient[CG.SG]	by

de	*dokter*	*was*	*successvol*
the[CG.SG]	doctor[CG.SG]	be.SG.PST	successful

[20]While the argument marking possibilities listed by Havranová (2020) suggest (51) should be possible, I could not find any actual examples of an action nominal simultaneously taking both a possessive marked P and a *door* 'by' marked A. However, Havranová (2020) does provide examples of action nominals taking these arguments individually with the appropriate meaning of a patient or agent.

'The treatment of the patient by the doctor was successful.' [nld] constructed based on (Havranová, 2020)

(51) (ERG-POSS)

jan	*s*	*behandel-ing*	*door*	*de*	*dokter*
John[CG.SG]	POSS	treat-NMLZ.CG.SG	by	the[CG.SG]	doctor[CG.SG]

was	*successvol*
be.SG.PST	successful

'John's treatment by the doctor was successful.' [nld] constructed based on (Havranová, 2020)

ge- action nominals

ge- action nominals show the exact same behavior as determined nominal infinitives aside from lacking the bare argument infinitive construction shown in (48), where the A of transitive action nominals is marked with the *van* 'of' preposition and the P is left bare.

I construct a testsuite using examples of all the different ANCs across the three different affixes in subject, object, and clausal modifier positions. I also include examples containing coordination. I create ungrammatical examples by using incorrect argument marking, using the wrong modifier (adjective versus adverb as well as the determiner of the wrong gender), or using the wrong order among modifiers. My testsuite contains 50 grammatical examples and 12 ungrammatical examples. I include direct examples from Looyenga (1992), Shetter and Ham (2007), and (Havranová, 2020). I also use the examples and information in these three works to construct additional grammatical and ungrammatical examples.

Implementation

I implement (42) as a non-affix pronominal strategy with a pronoun such as *mijn* 'my' having inherent number and person marking, and both (43) and (44) as possessive strategies with a non-affix marking the possessor. (43) is a possessum-first strategy and (44) is a possessor-first strategy. Since the possessum in (43) takes a determiner I cannot implement it as a specifier strategy, but only as a modifier strategy. This subsequently means that

under my analysis the *van* possessive marking strategy cannot be used in ANCs. I instead implement the *van* 'of' preposition used in ANCs as a semantically empty preposition.

Since determined nominal infinitives and *ge-* action nominals pattern in the same way, I implement two nominalization strategies (NOMINAL and ALL-COMPS) used by both the infinitive suffix *-en* and the *ge-* prefix. Both strategies can act on intransitive as well as transitive verbs and can be modified by both adjectives and adverbs. To avoid spurious ambiguity, I require the NOMINAL strategy to have an overt Poss, and the ALL-COMPS strategy to have a determiner. For all three different kinds of action nominals I make the assumption that the action nominal must appear with either a Poss or a determiner, but it is not fully clear if this is correct and more data is needed to be certain.

For the *-ing* action nominals, I define three additional nominalization strategies (NOMINAL, ALL-COMPS, ERG-POSS). These can only act on transitive verbs and can be modified only by adjectives. The NOMINAL and ERG-POSS strategies require mandatory Poss arguments and the ALL-COMPS strategy requires a mandatory determiner.

I specify that all NOMINAL/ERG-POSS strategies can use all the defined possessive strategies aside from the *van* modifier strategy. I also specify that ANC word order should be the same as the word order in main clauses, which means that ANCs will have v2 word order contrary to other subordinate clauses which use v-final word order.

I model both *door* and *van* as semantically empty prepositions and define one lrt per each nominaization strategy constraining the appropriate lrts to have the correct preposition markings. I specify neuter gender on the nominal infinitive and *ge-* lrts and common gender on the *-ing* lrts. I add the nominalization position classes as input to a nominal number position class so that action nominals will receive the number marking necessary to agree correctly with determiners.

Results

The implemented grammar produced by the customization system based on my Dutch [nld] choices file achieves 90.0% coverage, 8.3% over-generation and an average of 17.62 parses

per sentence on the testsuite. The lack of coverage is primarily a result of four sentences which contain the bare argument infinitive construction shown in (48). It is not fully clear whether the bare noun *postzegels* 'stamps' in (48) should be analyzed as serving as the object of the action nominal or as forming a compound with the action nominal (since Dutch [nld] allows infinitives to serve as the head of compounds (Booij, 1992)). The English analog of the compounding option would be the phrase *stamp-collecting*. The current library cannot handle either scenario. In the first case the ANC could be classified as a POSS-ACC ANC (a Poss A and a regular verbal complement P), but this would require treating the *van* phrase as a possessive modifier and not as a complement. This is not currently an option. The analysis involving a compound would fall under one of the minor nominalization types (the incorporating type where the P argument forms a compound with the action nominal (Koptjevskaja-Tamm, 1993)) which are out of the library's scope.

The one additional sentence the grammar can not parse is shown below in (52).

(52) *na* *de* *vernietig-ing* *van* *de* *stad* *vlucht-t-en* *de*
after the[CG.SG] destroy-NMLZ[CG.SG] of the city[CG.SG] flee-PST-PL the
bewoner-s
inhabitant-PL
'After the destruction of the city the inhabitants fled.' [nld] constructed based on (Havranová, 2020: 18)

In Dutch [nld] when an element other than the subject (such as the adverbial *na* 'after' phrase in (52)) appears in first position the subject (*bewoners* 'inhabitants') appears directly after the main clause verb (*vluchtten* 'fled') (Shetter and Ham, 2007). It is not fully clear why this sentence does not parse, but it might be a bug with the v2 word analysis in the Grammar Matrix, as this same problem occurs with v2 grammars created prior to my edits involving non-nominalized clausal modifiers.

The one sentence that leads to over-generation is shown in (53) where the adjective and adverb occur in the wrong order.

(53) **dat* *langzaam* *irritante* *uitblaz-en* *van* *de* *rook* *is*
that[NEUT.SG] slowly irritating puff.out-NMLZ of the smoke be.3SG
moelijk
difficult
'The slowly irritating puffing out of the smoke is difficult.' [nld] constructed based on (Looyenga, 1992: 175)

Under the current implementation, there is no way to constrain the order of adverbs and adjectives when both can modify the action nominal. There are ways the analysis could be updated to account for this. Currently when action nominals can be modified by both adverbs and adjectives the ADV-MOD value of the action nominal is left underspecified. It would be possible for this value to start out underspecified, but then take on a specific value once an adjective or adverb attaches. For example, in Dutch [nld], the action nominal could be become [ADV-MOD −] once the adverb attaches, requiring all remaining modifiers to be adjectives.[21] However, more data is needed to better understand how to represent these facts including whether multiple adverbs can attach prior to an adjective. Likewise, it would be necessary to compare the behavior in Dutch [nld] to other languages that also have simultaneous adverb and adjective attachment. Thus, the adverb/adjective analysis is for now left as is.

Dutch [nld] also has the highest average ambiguity out of all the held-out languages at 17.62. A significant portion[22] of this ambiguity comes from the same bug as mentioned with Maltese [mit] involving possessive modifiers being able to modify action nominals. Constraining possessive modifiers to modify [NMZ −] lexical items drops the average analysis number to 3.31.

[21]For this approach to potentially work, ADV-MOD could no longer be treated as a HEAD feature in languages with these kinds of adjective adverb ordering constraints. This is because headed rules such as the mod-head rules would need to be able to change an element's ADV-MOD value.

[22]Another source of ambiguity in this grammar is that a bug in the Grammar Matrix causes one rule to be added twice under different names.

6.3.4 Lango [laj]

Data/Description

I base my description of Lango [laj] and the contents of my testsuite entirely off the grammar description in Noonan (1992). Lango [laj] is a Eastern Sudanic language spoken in Uganda mainly within the Lango province (Noonan, 1992). It has SVO word order and no independent words acting as determiners. Lanjo is a tone language with the tone of a verb indicating its aspect and mood. Finite verbs also take prefixes which agree in person and number with the subject. Nominalizations have idiosyncratic tone meaning each nominalization requires its own lexical entry. Since this nominalization pattern is not possible with the current library, I instead analyze action nominals as a separate verb type which are required to take a null nominalization affix. Each action nominal then has a lexical entry with its own inherent tone marking and must mandatorily take a null nominalization affix.

Turning to possessive strategies, inalienable possession can be marked through juxtaposition (54), while a non-pronominal alienable possessor is marked with the attributive particle *à* (55). In both constructions the possessor follows the possessum. Pronominal possessors are marked with a suffix on the possessum (56). The possessor affixes do not have to appear on the possessum, but can also appear on the last element of the NP. However, this is not a fact that the adnominal possessum library can currently model. There is also another possessive construction which Noonan (1992) calls a complex associative. This construction is most common with body part possessums appearing in subject position, but can also occur with other possessums. Using a complex associative versus one of the other possessive options puts special attention on the possessor. It involves placing the possessor before the possessum and marking the possesum with a pronominal suffix that agrees with the possessor (57).

(54) (inalienable possession)

wì *rwòt*
head king
'The king's head.' [laj] (Noonan, 1992: 157)

(55) (alienable possession)

gwôkk *à* *lócɔ̀*
dog POSS man
'The man's dog.' [laj] (Noonan, 1992: 156)

(56) *tyɛ̌n-wá*
leg-1PL.POSS
'Our legs.' [laj] (Noonan, 1992: 78)

(57) *lócɔ̀* *gwók-kɛ́rɛ̂*
man dog-3SG.POSS
'The man's dog.' [laj] (Noonan, 1992: 160)

The construction in Lanjo which best meets the library's definition of action nominal is the nominalized infinitive.[23] In nominalized infinitives, the P is expressed in the same way as for finite verbs and the A is marked as a Poss. Lanjo [laj] does not have attributive adjectives and Noonan (1992) does not specify whether adverbs can modifiy ANCs. When using one of the non-pronominal possessive strategies shown in (54) or (55), the word order in ANCs is action.nominal P A (58). This kind of ANC falls under the category of POSS-ACC.

(58) (POSS-ACC)
nèkkò *gwôkk* *à* *dákô* *Ø-ràc*
kill\NMLZ dog POSS woman 3SG-bad\HAB
'The killing of a dog by a woman is bad.' [laj] (Noonan, 1992: 214)

If an ANC uses the complex associative possessive construction in (57), the word order is instead A action.nominal P (59).

(59) *dákô* *nèkkò* *gwók-kɛ́rɛ̂* *Ø-ràc*
woman kill\NMLZ dog-3SG.POSS 3SG-bad\HAB
'The woman's killing of the dog is bad.' [laj] (Noonan, 1992: 214)

[23]Lango [laj] also has a gerund which can use possessive affixes to mark the A (Noonan, 1992). However, I could not find a description of any way to mark the P of transitive gerunds, so as far as I understand, Lango gerunds do not meet the definition of action nominals used by the library.

In this example, the possessive affix *-kɛ́rɛ̂* appears not on the action nominal itself, but on the P. As mentioned above for non-derived possessive noun phases, the possessive affixes are not limited to appearing on the possessum, but can also appear on the last element of an NP. Noonan (1992) does not provide an example of an ANC using the complex associative construction where the possessive affix appears on the action nominal itself, so it is unclear if this is permitted.

The Lango [laj] testsuite contains both intransitive and transitive examples in subject and clausal modifier positions with the A marked by different possessive strategies. I construct ungrammatical examples by using incorrect marking on the action nominal, its arguments, or using an incorrect word order in ANCs. The testsuite has 23 grammatical examples and 16 ungrammatical examples. I include both actual examples from Noonan (1992) as well as additional constructed examples.

Implementation

I implement (54) as a possessum-first juxtaposition strategy, (55) as a possessum-first non-affix possessor-marking possessive strategy, (56) as a pronominal possessor affix strategy, and (57) as a possessor-first possessive strategy where the affix on the possessum agrees in person and number with the preceding possessor. To implement ANCs, I create a POSS-ACC nominalization strategy which acts on both transitive and intransitive verbs and uses VOS word order (as opposed to SVO). This allows for the word order shown in (58). I specify that action nominals can use any of the defined possessive strategies. Since I am unsure about the modification status, I do not allow action nominals to be modified by either adjectives or adverbs.

As mentioned above, nominalizations in Lango [laj] need their own lexical entries. However, currently the library requires each nominalization strategy to be linked to an lrt. In my implementation, I create a verb type specific to nominalized infinitives and require them to go through a null nominalization lrt. All other verb types are forbidden from this lrt. This implementation then assumes a morphophonological analyzer with lexical item-specific

rules which can actually assign the correct tone to each entry in the nominalized infinitive verb type. The position class containing the nominalization lrt serves as input to the position class containing the possessive affixes allowing nominalized infinitives to take possessive morphology. Since it is not clear if the action nominal can take the possessive affix in the complex associative construction, I forbid the nominalization lrt from co-occurring with that affix.

Results

The implemented grammar produced by the customization system from my Lango [laj] choices file achieves 91.1% coverage, 26.7% over-generation and an average of 1.45 parses per sentence. The missing coverage is due to ANCs containing the complex associative construction. The current library cannot handle a sentence such as (59) for two reasons. The first is that the adnominal possession library cannot account for an analysis where the possessive suffix appears not on the possessum *nèkkò* 'kill\NMZ', but on the noun phrase the possessum is a part of. The second reason is that the library requires all ANCs which use possessive strategies to have the same word order. However, both (58) and (59) have distinct and fixed word orders. It might be possible to say that the word order is free and rely on other constraints to rule out ungrammatical combinations, but the library itself will not provide the necessary constraints. This second issue reveals a limitation of the library and suggests that it might be necessary to allow ANCs to take more than one alternative word order. However, the complex associative construction is also very similar to topicalization in Lango [laj] (Noonan, 1992), which suggests that the analysis of this type of possessive construction may require something beyond either just the adnominal possession or nominalized clauses library. More data is then needed to understand how best to handle the complex associate construction.

The over-generation is caused by a potential bug in the adnominal possession library where when a language (without determiners) simultaneously contains a possessor-first and possessum-first specifier strategy, the grammar incorrectly allows both orders in possessive

constructions regardless of the possessive strategy used. An example is shown in (60).

(60) **à dákô nèkkò gwôkk Ø-ràc*
POSS woman kill\NMLZ dog 3SG-bad\HAB
'The killing of a dog by a woman is bad.' [laj] constructed based on (Noonan, 1992)

In (60) the *à* preposition correctly marks the possessor, however the strategy that uses the *à* marker is supposed to be a possessom-first strategy. This same problem shows up in several sentences in the testsuite. Taking the exact same choices file devoid of anything to do with nominalizations and customizing it with the system prior to any of my edits results in a grammar that does not compile.[24] Taking this same nominalizations-free choices file and customizing it with the system after my edits results in a grammar with the same ordering issues described above in regular possessive noun phrases. This bug then seems to extend beyond just the nominalized clauses library. While fixing this bug is considered out of scope for this thesis, it does demonstrate what is both a weakness and a strength of the current analysis. Any issue with the adomonial possession library will also affect the nominalized clauses library, but likewise any fix or improvement to the adnominal possession library will benefit both.

6.3.5 Finnish [fin]

Data/Description

The following description of Finnish [fin] as well as the sentences in my testsuite are based on both Brattico and Leinonen (2009) and Karlsson (2017). Finnish [fin] is a Uralic language with flexible word order with SVO being the most neutral. In finite clauses, the subject is most often marked with nominative case while the object can be marked with accusative, genitive, or paritive case (Karlsson, 2017). When the subject takes nominaitve case, the finite verb agrees in person with the subject.

[24]The grammar does not compile because *head-spec-phrase* is specified as inheriting from both *head-initial* and *head-final*. These two types are incompatible within the same rule. I came across this compilation issue while working on other languages during development and fixed it by adding both *head-spec-phrase* and *spec-head-phrase* to the grammar with each inheriting from the appropriate type. This allows the grammar to compile, but does not address the bug revealed by the Lango [laj] testsuite.

A possessor can be marked in one of three ways: (1) a possessive pronoun with the possessum taking a possessive suffix agreeing in person and number with the possessor (61), (2) a possessor suffix on the possessum (62), and (3) a non-pronominal possessor marked with genitive case (63).

(61) *minun* *kirja-Ø-ni*
1SG.GEN book-NOM-POSS.1SG
'My book.' [fin] (Karlsson, 2017)

(62) *kirja-Ø-ni*
book-NOM-POSS.1SG
'My book.' [fin] (Karlsson, 2017: 6)

(63) *pekka-Ø-n* *auto*
pekka-SG-GEN car[NOM.SG]
'Pekka's car.' [fin] (Karlsson, 2017: 123)

Action nominals can be created in Finnish [fin] with the addition of the *-minen* nominalizer suffix. The action nominal takes case as would any other noun. Both the A and P are marked with genitive case with the A preceding the P and both arguments preceding the action nominal (A P action.nominal). Since both arguments take the same marking as a Poss (63), this ANC falls under the NOMINAL category. An example with an intransitive and transitive verb are shown in (64) and (65) respectively.[25]

(64) (NOMINAL)
pekka *nak-i-Ø* *merja-n* *lahte-misen*
pekka[SG.NOM] see-PST-3SG merja-GEN leave-NMLZ.GEN.SG
'Pekka saw Merja's leaving.' [fin] constructed based on (Brattico and Leinonen, 2009: 9)

(65) (NOMINAL)
isa-n *nopea* *auto-n* *osta-minen* *ol-i* *yllatys*
father-GEN fast[NOM.SG] car-GEN buy-NMLZ.NOM.SG be.3SG-PST surprising

[25]In (64) I modify the gloss somewhat from (Brattico and Leinonen, 2009) to more closely align with the grammar presented in (Karlsson, 2017). Specifically, Karlsson (2017) states that only pronominal objects in Finnish take accusative case and most non-pronominal objects take either genitive or paritive case. This is also how I implement the Finnish test grammar. In contrast, Brattico and Leinonen (2009) gloss the object of *naki* 'saw' as taking accusative case.

> 'The fast buying of a car by the father was surprising.' [fin] constructed based on (Brattico and Leinonen, 2009: 19)

As (65) demonstrates, action nominals can be modified be adjectives (*nopea* 'fast'). In Finnish [fin], ajectives agree with nouns in number and case. The same holds true for action nominals. The resources I use do not specify action nominals can be modified by adverbs.

The Finnish [fin] testsuite includes intransitive and transitive examples of ANCs occurring in subject and object position. It also includes examples with adjectival modifiers. I construct ungrammatical examples by using the incorrect marking on the action nominal, including using the wrong case value, using the wrong case marking on the arguments of the action nominal, and using finite clause word order in ANCs. The testsuite contains 19 grammatical examples and 30 ungrammatical examples. I include both actual and constructed examples from both Karlsson (2017) and Brattico and Leinonen (2009).

Implementation

I implement (61) as a an affix on the possessum which agrees in person and number with the possessor, (62) as a pronominal affix strategy, and (63) as a juxtaposition strategy with the possessor constrained to be non-pronominal and to have genitive case. This is consistent with the treatment of Finnish adnominal possession in Nielsen (2018) who used Finnish as an illustrative language.

To model ANCs, I create a NOMINAL nominalization strategy which acts on both intransitive and transitive verbs and can be modified by adjectives. I specify SOV word order in ANCs and allow action nominals to use only the genitive-marking possessive strategy (63). Since both arguments of the ANC are marked as a Poss, only the A argument can actually use the possessive strategy (see 4.2.2). The P argument is treated as a genitive-marked complement. I then create an lrt for the *-minen* nominalization suffix which uses the NOMINAL nominalization strategy and constrains its object to have genitive case. One of the arguments of the action nominal is then a genitive specifier and the other is a genitive complement. The nominalization position class serves as input to the nominal number position class, so that

action nominals take number and case morphology like other nouns. This also allows action nominals to properly agree with their adjectives. I forbid action nominals from taking any possessive-marking morphology. This prevents either of the possessive strategies shown in (61) or (62) from applying in ANCs.

Results

The implemented grammar produced by the customization system based on my Finnish [fin] choices file achieves 94.4% coverage, 0% over-generation, and 1.33 average parses per sentence on the testsuite.

The one sentence that does not parse is shown in (66).

(66) *nopea* *merja-n* *lahte-minen* *ol-i* *yllatys*
fast[NOM.SG] merja-GEN.SG leave-NMLZ-NOM.SG be.3SG-PST surprising
'Merja's fast leaving was surprising.' [fin] constructed based on (Brattico and Leinonen, 2009: 18)

This sentence does not parse because the adjective *nopea* 'fast' attaches outside the genitive specifier *merja-n* 'Merja-GEN.SG'. If this example consisted of a transitive action nominal with a dropped specifier and a genitive complement there would be no problem. This is because adjectives in the Grammar Matrix modify lexical items with non-empty specifier values. Thus, a adjective can not modify a noun (or in this case an action nominal) once it has already picked up its specifier. This is another case where more data needs to be seen before a change can be made to the analysis. The main question is whether adjectives can attach before genitive possessors in non-derived noun phrases or if this possibility is exclusive to ANCs. If the same option can occur in non-derived noun phrases, then this is a larger problem with the Grammar Matrix's analysis of adjectival modifiers. If this can occur only in ANCs then it would be possible to analyze these ANCs as ALL-COMPS ANCs (taking two genitive complements). However, in the current library, ALL-COMPS ANCs can only use the same head-comp order as used in finite clauses. Since I implement the Finnish [fin] grammar as taking SVO word order, this means that both complements would have to occur

after not before the action nominal. This then suggests that it may be advantageous to allow ALL-COMPS ANCs to use alternative head-comp word orders.

6.4 Summary

The results across all the held-out languages are shown below in Table 6.5.[26] The implementation proves to be reasonably robust showing good coverage and relatively little over-generation.

Table 6.5: Held-out language test results

	# gramm-atical	# ungramm-atical	% coverage	% over-generation	avg-analysis-#
way	12	30	100	0	9.25
mit	12	8	100	0	3.25
nld	50	12	90.0	8.3	17.62
laj	23	15	91.3	26.7	1.45
fin	19	30	94.7	0	1.33

As mentioned above, I discovered two bugs during evaluation that I fixed afterwards. One was that subordinator adpositions had underspecified FORM values and the other was that possessive modifiers could modify action nominals. I address the first problem by requiring subordinator adpositions to take a FORM value whenever a semantically empty adposition is defined. I fix the second bug by requiring all possessive modifiers to modify [NMZ −] lexical items whenever a language contains defined nominalization strategies.[27]

The process of evaluation also revealed three more significant problem areas for the analysis. Those being: (1) control over the order of adjectives and adverbs when both can

[26]The version of the system used for evaluation can be found at `https://github.com/delph-in/matrix/tree/anc-branch` currently on commit 127b21f214ce8af8fef44c008daf6744b4d07992. This is the code that can be used to replicate the results shown in table 6.5.

[27]The version of the system containing these post-evaulation changes can be found at `https://github.com/delph-in/matrix/tree/anc-branch-post-evaluation`.

simultaneously modify an action nominal (Dutch [nld]), (2) the potential ability for adjectives to attach outside of specifiers (Finnish [fin]), and (3) the freedom to allow ANCs with possessive strategies to use multiple word orders when a language has possessive strategies with contrasting possessor-possessum word order (Dutch [nld] and Lango [laj]). All of these issues require more data and potentially development of other parts of the Grammar Matrix (such as topicalization or compounding) to fully resolve.

Chapter 7
CONCLUSION

In this thesis, I have described the analysis and implementation of an action nominal construction (ANC) library for the Grammar Matrix customization system. The main goal of this work was to expand the existing nominalized clauses library to cover languages without case-marking in ANCs and better incorporate nominal syntax into the analysis of nominalized clauses. This was done by shifting the analysis away from a more mixed change-over and lexical rule approach to one that relies primarily on lexical rules. This change allows for a more-fine grained approach to elements such as word order in ANCs and adverb/adjective modification. Relying on existing nominal analyses in the Grammar Matrix such as those in the adnominal possession library also grants the library the ability to handle action nominals with syntactic possessors as well as determiners, on top of action nominals which take subjects and were handled by the previous library.

One of the benefits of these changes is the ability to test out different linguistic theories surrounding the interaction of verbal and nominal syntax. This can be seen in the questions that still remain with the proposed analysis, such as how to handle co-occurring adjectives and adverbs or whether to analyze a particular argument as a complement (more verbal) or a possessive specifier (more nominal). Relying on an existing library such as the adnominal possession library also makes future improvements and bug fixes easier as an upgrade to the adnominal possession analysis carries over to the nominalized clauses one. Its also more efficient as none of the existing analyses in the adnominal possession library need to be duplicated for ANCs.

The implemented library also significantly increases the range of phenomena users can implement in ANCs. This includes allowing for more control over elements such as optional arguments and coordination, as well as offering analyses for very specific configurations such

as cases where the single argument of a transitive action nominal always appears marked as a syntactic possessor. In addition to allowing ANCs which take possessive syntax, I also offer several different possessive semantic options, so that users have more freedom in choosing the final analysis. Finally, the library achieves the aim of covering ANCs in languages without case-marking, as I make it possible to not only mark arguments using possessive strategies, but also with semantically empty adpositions.

As mentioned throughout the thesis, there are numerous potential avenues for future work. One could include adding the minor nominalization types from the ANC typology (Koptjevskaja-Tamm, 1993) or adding certain action nominals that were treated as out of scope such as nominalized copulas or action nominals that cannot express all their arguments in an ANC. Refinements could also be made to the analysis of ANC word order and coordination, so that it is possible to make strategies more specific to certain types of action nominal. For example, future work could make it possible to coordinate one type of ANC with non-derived NPs, but not another.

The analysis of adjective and adverb modification also needs further exploration including the questions brought up during evaluation, but also to see how well it holds up when looked at with other types of modifiers as well as how the ADV-MOD feature interacts with other lexical items (beyond just nouns and verbs) that can be modified by adjectives/adverbs. There are also aspects of action nominals, such as how to negate action nominals, that I have not attempted to implement in the library. Negation (both inflectional and not) presents a particular semantic challenge that will further test the predominantly lexical rule analysis presented in this thesis. The work in this thesis then serves as a base which allows additional questions to be explored and answered.

BIBLIOGRAPHY

Alexiadou, A. (2010). Nominalizations: A probe into the architecture of grammar part I: The nominalization puzzle. *Language and Linguistics Compass*, 4:496–511.

Azzopardi-Alexander, M. and Borg, A. (1996). *Maltese*. Taylor & Francis Group.

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

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

Bender, E. M. and Good, J. (2005). Implementation for discovery: A bipartite lexicon to support morphological and syntactic analysis. In Edwards, Midtlyng, S. and Stensrud, editors, *Chicago Linguistic Society: The Panels*, volume 41.

Booij, G. (1992). Compounding in Dutch. *Rivista di Linguistica*, 4.

Brattico, P. and Leinonen, A. (2009). Case distribution and nominalization: Evidence from Finnish. *Syntax*, pages 1–31.

Chang, S.-J. (1996). *Korean*. John Benjamins, Philadelphia.

Comrie, B. (1976). The syntax of action nominals: A cross-linguistic study. *Lingua*, 40:177–201.

Comrie, B. and Thompson, S. A. (1985). Lexical nominalization. In Shopen, T., editor, *Language Typology and Syntactic Description 3*, pages 349–398. Cambridge University Press, Cambridge.

Copestake, A. (2002). *Collaborative language engineering: A case study in efficient grammar-based processing*, volume 118 of *CSLI Lecture Notes*, page 227–230. CSLI Publications, Stanford, CA.

Copestake, A., Flickinger, D., Pollard, C., and Sag, I. (2005). Minimal recursion semantics: An introduction. *Reseach On Language And Computation*, 3:281–332.

Croft, W. (1991). Syntactic categories and grammatical relations: The cognitive organization of information. University of Chicago Press.

Crysmann, B. (2011). A unified account of Hausa genitive constructions. In de Groote, Egg, P., Kallmeyer, M., and Laura;, editors, *Formal Grammar. 14th International conference, FG 2009, Bordeaux*, Lecture Notes in Artificial Intelligence. Springer. 978-3-642-20168-4.

Crysmann, B. (2012). HaG - A computational grammar of Hausa. In *Selected Proceedings of the 42nd Annual Conference on African Linguistics*, page 17 pp., France. Cascadilla Press.

da Silva Tavares, P. (2005). *A grammar of Wayana*. PhD thesis, Rice University.

Derbyshire, D. C. (1985). *Hixkaryana and linguistic typology*. Summer Institute of Linguistics., Dallas.

Drellishak, S. (2009). *Widespread but not universal: Improving the typological coverage of the Grammar Matrix*. PhD thesis, University of Washington.

Drellishak, S. and Bender, E. M. (2005). A coordination module for a crosslinguistic grammar resource. In Müller, S., editor, *The Proceedings of the 12th International Conference on*

Head-Driven Phrase Structure Grammar, Department of Informatics, University of Lisbon, pages 108–128, Stanford. CSLI Publications.

Fehri, A. F. (1993). *Issues in the structure of Arabic clauses and words*. Dordrecht: Kluwer Academic.

Flickinger, D., Bender, E. M., and Bender, S. O. (2003). *MRS in the LinGO Grammar Matrix: A practical user's guide*.

Goodman, M. W. (2013). Generation of machine-readable morphological rules from human-readable input. *UW Working Papers in Linguistics*, 30.

Havranová, K. (2020). *The nominalization process in English, Dutch, and Czech*. PhD thesis, Palacký University, Olomouc.

Howell, K. and Zamaraeva, O. (2018). Clausal modifiers in the Grammar Matrix. In *Proceedings of COLING 2018*.

Howell, K., Zamaraeva, O., and Bender, E. (2018). Nominalized clauses in the Grammar Matrix. *Proceedings of the International Conference on Head-Driven Phrase Structure Grammar*.

Hudson, R. (2003). Gerunds without phrase structure. *Natural Language & Linguistic Theory*, 21(3):579–615.

Karlsson, F. (2017). *Finnish: A comprehensive grammar*. Routledge.

Kazenin, K. I. (1996). The syntactic structure of action nominal constructions. In Kibrik, A. E., editor, *Godoberi*, pages 160–161. Lincom Europa, München.

Kim, J.-B. and Yang, J. (2004). Projections from morphology to syntax in the Korean Resource Grammar: Implementing typed feature structures. In *CICLing*, volume 2945 of *LNCS*, Seoul, Korea. Heidelberg: Springer.

Koptjevskaja-Tamm, M. (1993). *Nominalizations*. Routledge, London.

Koptjevskaja-Tamm, M. (2003). *Noun phrase structure in the languages of Europe*, chapter Action nominal constructions in the languages of Europe, pages 723–760. De Gruyter Mouton, Berlin, New York, seventh edition.

Koptjevskaja-Tamm, M. (2006). *Encyclopedia of languages & linguistics*, volume 8, chapter Nominalization, page 652–659. Elsevier, second edition.

Koptjevskaja-Tamm, M. (2013). Action nominal constructions (v2020.3). In Dryer, M. S. and Haspelmath, M., editors, *The World Atlas of Language Structures Online*. Zenodo.

Koptjevskaja-Tamm, M. (2015). *Word formation. An international handbook of the languages of Europe.*, chapter Action nouns, page 1195–2009. Handbooks of Linguistics and Communication Sciences (HSK). De Gruyter Mouton, Berlin – New York.

Looyenga, S. (1992). A syntactic analysis of Dutch nominal infinitives. *Linguistics in the Netherlands*, page 173–184.

Malouf, R. (1998). *Mixed categories in the hierarchical lexicon*. PhD thesis, Stanford University.

Nielsen, E. (2018). Modeling adnominal possession in the LinGO Grammar Matrix. Master's thesis, University of Washington.

Noonan, M. (1992). *A grammar of Lango*. Walter de Gruyter.

O'Hara, K. (2008). *A morphotactic infrastructure for a grammar customization system*. PhD thesis, University of Washington.

Pollard, C. and Sag, I. (1994). *Head-driven phrase structure grammar*. The University of Chicago Press.

Pollard, C., Sag, I., and Bender, E. M. (2003). *Syntactic theory: A formal introduction*. Stanford: CSLI Publications.

Saleem, S. (2010). Argument optionality: A new library for the Grammar Matrix customization system. Master's thesis, University of Washington.

Shetter, W. Z. and Ham, E. (2007). *Dutch: An essential grammar*. Routledge.

Song, S. (2014). *A grammar library for information structure*. PhD thesis, University of Washington.

Timberlake, A. (2004). *A reference grammar of Russian*. Cambridge University Press.

Yoon, J. H. S. (2022). *Lexical nominalizations in Korean*, page 429–457. Cambridge Handbooks in Language and Linguistics. Cambridge University Press.

Zamaraeva, O., Chris, C., Guy, E., Antske, F., Michael Wayne, G., Kristen, H., T.J., T., and Bender, E. M. (2022). 20 years of the Grammar Matrix: Cross-linguistic hypothesis testing of increasingly complex interactions. *Journal of Language Modeling*, 10:49–137.

Zamaraeva, O., Howell, K., and Bender, E. M. (2019). Modeling clausal complementation for a grammar engineering resource. In Jarosz, G., Nelson, M., O'Connor, B., and Pater, J., editors, *Proceedings of the Society for Computation in Linguistics (SCiL) 2019*, pages 39–49.