Widespread but Not Universal: Improving the Typological Coverage of the Grammar Matrix

Scott Drellishak

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

Doctor of Philosophy

University of Washington

2009

Program Authorized to Offer Degree: Department of Linguistics

University of Washington
Graduate School

This is to certify that I have examined this copy of a doctoral dissertation by

Scott Drellishak

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Chair of the Supervisory Committee:

Emily M. Bender

Reading Committee:

Emily M. Bender

Barbara Citko

Sharon Hargus

Date: ______________________

Signature____________________________

Date_______________________________

University of Washington

Abstract

Widespread but Not Universal:
Improving the Typological Coverage of the Grammar Matrix

Scott Drellishak

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

The LinGO Grammar Matrix provides a foundation for building grammars of natural languages in HPSG. It includes a web-based questionnaire that allows a linguist to describe a natural language, and then creates a starter grammar for that language based on the answers. In this dissertation, I describe improvements I have made to the typological coverage of this system, including support for core case marking, agreement in person, number, and gender, direct-inverse languages, and a lexicon containing an arbitrarily large number of lexical types, lexical items, and inflectional morphemes.

TABLE OF CONTENTS

Page

LIST OF FIGURES

LIST OF TABLES

Table Number Page

ACKNOWLEDGMENTS

This dissertation would not have been possible without the support of many people, to whom I would like to express my sincere gratitude. First and foremost, my advisor Emily Bender has provided me with a depth of knowledge, abundant enthusiasm, detailed criticism, and artificial deadlines without which I would never have finished, and for which I will be eternally grateful. The other members of my supervisory committee also have my thanks: Barbara Citko for providing an alternative syntactic perspective, Sharon Hargus for directing my attention to Sahaptin and helping me understand its grammar, Fritz Newmeyer for early advice and continued emeritus guidance, and Ed Taylor for his support and kind words of encouragement.

I would also like to thank the other Matrix developers for their invaluable research, suggestions, ideas, and opinions. In particular, I would like to acknowledge Kelly O'Hara for her work on inflectional morphology, which formed an indispensable foundation for my own work on agreement, and Laurie Poulson for her ongoing work on a library for tense and aspect, which inspired several improvements to my own libraries and to the system as a whole. Antske Fokkens, Michael Goodman, Joshua Hou, and Safiyya Saleem also provided a sounding board for new ideas at our weekly meetings. Furthermore, all the students in the UW grammar engineering course helped test and exercise the customization system, exposing several bugs and deficiencies.

I owe a debt of gratitude to Bernard Comrie for providing me the opportunity to visit and study at the Max Planck Institute in Leipzig, where my initial forays into the typology of case and agreement took place. I am also grateful to all my colleagues at the Institute, but especially to Jeff Good, Martin Haspelmath, and Michael Cysouw, for their insight and advice on linguistic phenomena, typology, databases, and questionnaires.

This work would not have been possible without the research and work of many people in the HPSG and DELPH-IN communities. In particular, I wish to thank Dan Flickinger for his work on the Matrix and for early help in refining the focus of my dissertation, Ann Copestake and Stephan Oepen for the software tools that made my work possible, Stefan Müller for calling my attention to TRALE, Anne Abeillé for reminding me about optional adpositional case marking, and all the attendees of the HPSG 2008 conference for their kind attention and probing questions.

Finally, I want to express my gratitude to my wife Renée Drellishak, who not only put up with me disappearing into the basement for many, many late nights, but also generously proofread this dissertation for grammar and spelling errors.

This dissertation is based upon work that was supported by a gift to the Turing Center from the Utilika Foundation, by the Max Planck Institute for Evolutionary Anthropology, and by the National Science Foundation under Grant No. 0644097. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation.

DEDICATION

to my wife, Renée

Chapter 1
INTRODUCTION

Grammar engineering is "the process of creating machine-readable implementations of formal grammars" (Bender 2008a:16). It straddles the border between formal syntax and computational linguistics—grammar engineers both analyze natural languages in formal frameworks and implement those analyses in software. Some grammar engineers focus on a single natural language, developing deep coverage grammars like the English Resource Grammar (ERG) (Copestake and Flickinger 2000); others implement grammars of multiple, sometimes related, languages. Grammar engineers of either sort face a serious hurdle when beginning the analysis of a new language: significant effort must be expended analyzing and implementing structures and phenomena that occur in many or all languages. This is wasteful of time and other limited resources.

The **LinGO Grammar Matrix** (Bender et al. 2002, and see §2.4 for much more detail) is intended to help grammar engineers overcome this hurdle. It provides a foundation for building grammars of natural languages in HPSG. In its earliest form, it provided a set of predefined types that gave grammar engineers a head start, allowing them to avoid duplicating the effort needed to develop analyses of linguistic structures thought to occur in all languages. It soon became clear, however, that supporting only universal phenomena was not enough. Many linguistic phenomena exist that are widespread, but not universal. If the Matrix were restricted to supporting only what is truly universal, it would be a much less useful resource for grammar-writers working on languages containing such non-universal phenomena.

The solution has been to provide the **Matrix customization system**, which presents a linguist with a web-based typological questionnaire designed to elicit a description of a target language and, based on it, create a **starter grammar**—that is, a small grammar in software that parses and generates the target language. These grammars are not encumbered by phenomena that do not occur in the target language; rather, they contain just enough complexity to model it as described. We conceive of the customization system as consisting of a set of **libraries**, each of which supports a particular linguistic phenomenon, and includes a section of the questionnaire and a syntactic analysis of the target phenomenon that can be customized and included in output grammars.

This dissertation describes my contributions to the customization system, which have been twofold. First, I have added libraries for several linguistic phenomena: case, agreement in person, number, and gender, and support for direct-inverse languages. Each of these libraries is based on a thorough survey of the relevant typological literature, and is intended to cover as wide a range of the target phenomenon as possible. For each phenomenon, I have developed both a questionnaire section that elicits a description of the phenomenon in a target language, an HPSG analysis of the phenomenon, and the software code necessary to customize the analysis to the user's answers and emit it in computer-readable form. For each of these libraries, I have performed extensive testing to verify that the grammars produced by the system behave as expected, recognizing and assigning correct analyses to grammatical sentences while rejecting ungrammatical sentences.

Second, in order to support the new libraries, I have greatly enhanced and expanded the capabilities of the software that underlies the customization system. These enhancements include, but are not limited to: support for a structured questionnaire that can expand as it is filled out, allowing the description, for example, of an arbitrary number of lexical types, lexical items, and inflectional morphemes; support for dynamic web pages that help the user avoid answering questions incorrectly, and lead the user to errors when they do occur; support for exporting and importing versionstamped **choices files** that encode the answers (i.e., the linguistic description) the user has entered; and the completion of a port of the system from Perl to Python, a simple but powerful programming language that we hope will make it easier for other linguists to contribute libraries in the future.

1.1 Organization

This dissertation will be organized as follows. After this introductory chapter, Chapter 2 will provide background information on linguistic typology, multi-lingual grammar engineering, and the Grammar Matrix, including a discussion of the history of the project and its architecture.

The next three chapters, 3, 4, and 5, will each discuss the addition of a new linguistic phenomenon to the Grammar Matrix. Each will consist of four major sections: first, a description of the linguistic phenomenon and its typology; second, an HPSG analysis of the phenomenon and a description of how the analysis models it; third, a description of the section of the questionnaire section eliciting a description of the phenomenon; and fourth, a section containing test cases that verify the function of the analyses by producing and testing grammars of language fragments that exemplify the target phenomenon.

Chapter 6 will present a detailed case study of the description and modeling of Sahaptin, a language that displays complex interactions between the phenomena supported by the customization system. Finally, Chapter 7 will discuss the implications of this research for linguistic typology and describe some directions for future work.

1.2 Format

This dissertation conforms to the following formatting conventions. The initial mentions or defining instances of technical terminology are in **bold face**. HPSG feature names are in SMALL CAPS, and types (or feature values) are in *italics*. File names, URLs, excerpts from computer code (such as the code that makes up the Matrix), and the contents of choices files are in a `fixed-width font`. Languages under discussion will be identified by following their names with the appropriate three-letter ISO 639-3 language code in [square brackets] and genetic classification in (parentheses). Linguistic examples are glossed according to the Leipzig Glossing Rules (Bickel et al. 2008), with occasional minor compromises when necessary to remain true to the cited sources. The last line of each linguistic example, which contains the English translation, will also be followed by the three-letter ISO 639-3 code.

1.3 Abbreviations

ABS	absolutive
ACC	accusative case
ADP	adpositional phrase
AFOC	agent focus
ASL	American Sign Language
AVM	attribute value matrix
D	determiner
DU	dual number
ERG	English Resource Grammar
ERG	ergative case
F	feminine gender
FUT	future
GB	Government and Binding
GEN	genitive case
HPSG	Head-driven Phrase Structure Grammar
HTML	Hypertext Markup Language
IMPF	imperfective aspect
IPA	International Phonetic Alphabet
ISO	International Organization for Standardization

KPML	KOMET-Penman Multilingual
LFG	Lexical Functional Grammar
LinGO	Linguistic Grammars Online
LKB	Linguistic Knowledge Builder
MASC	masculine gender
MRS	Minimal Recursion Semantics
N	neuter gender
N	noun
NLG	natural language generation
NOM	nominative case
NP	noun phrase
NPST	non-past tense
PCFG	probabilistic context-free grammar
PERF	perfective aspect
PFOC	patient focus
PL	plural number
PST	past tense
SFOC	subject focus
SG	singular number
TDL	Type Description Language
UG	Universal Grammar
UW	University of Washington
V	verb
VP	verb phrase
XLE	Xerox Linguistic Environment

Chapter 2
BACKGROUND

This dissertation deals with the intersection of linguistic typology, formal syntax, and computational linguistics. The software system it describes is capable of creating, in the HPSG framework, formal grammars of natural languages that model a wide range of linguistic phenomena. Before the range, analysis, and implementation of these phenomena can be discussed, it is first necessary to situate this research in historical context by providing background information on four areas: HPSG, linguistic typology, the broad field of multi-lingual grammar engineering, and within that field, the Grammar Matrix in particular.

2.1 HPSG

In this section I will provide a brief introduction to Head-driven Phrase Structure Grammar (HPSG) (Pollard and Sag 1994, Sag et al. 2003), and in particular to the version of HPSG used in the Grammar Matrix (Copestake et al. 2005, and see §2.4). I will illustrate this introduction using a series of examples from a grammar of a small fragment of English created using the customization system.

2.1.1 Types

HPSG models the grammatical properties of signs using **types**. Types have features, and the values of features are themselves types, leading to the possibility of nested structures. These structures are called **typed feature structures**, and they are expressed using a notation called an **attribute value matrix** (AVM).

AVMs are formatted as a series of feature names in SMALL CAPS on the left, with the values of the features on the right, and with both columns surrounded by [brackets]. The type of the feature structure is shown at the top of the AVM in *italics*. Example (1) below shows a simple AVM, which has a type of *type1* and two features, FEATURE1 and FEATURE2. FEATURE1's value is *value1*, while FEATURE2's value is another feature structure of type *type2* containing FEATURE3 with value *value3*.

$$(1)\quad \begin{bmatrix} \textit{type1} & \\ \text{FEATURE1} & \textit{value1} \\ \text{FEATURE2} & \begin{bmatrix} \textit{type2} & \\ \text{FEATURE3} & \textit{value3} \end{bmatrix} \end{bmatrix}$$

In addition to types and feature structures, the values of features may also be lists, which are surrounded by < angle brackets >, as in the following AVM, in which the feature LIST1's value is a list containing two items of types *item1* and *item2*:

$$(2)\quad \begin{bmatrix} \text{LIST1} & \left\langle \textit{item1},\ \textit{item2} \right\rangle \end{bmatrix}$$

It is often useful to constrain two feature values to be identical. The AVM notation for this is to mark the identical values with numbers in small boxes; note that two values are identical only if the numbers in the boxes are equal, allowing any number of identity relationships to be expressed in the same feature structure. In the following example, the first element on LIST1 is identified with the second element on LIST2, and vice-versa:

$$(3)\quad \begin{bmatrix} \text{LIST1} & \left\langle \boxed{1}, \boxed{2} \right\rangle \\ \text{LIST2} & \left\langle \boxed{2}, \boxed{1} \right\rangle \end{bmatrix}$$

Below is an example of a feature structure that uses all of these formal devices at once. It represents the word *chase* in our fragment of English, and contains a subset of the features of that lexical item selected to illustrate the AVM notation. Notice in particular that the STEM feature has the value < “chase” >, a list containing a string representing its spelling; that the PRED feature similarly has the value < “_chase_v_rel” >, a predicate representing the meaning of the lexical item; and finally that the first and only elements on the SUBJ and COMPS lists are identified with the two items on the ARG-ST list.

(4)
$$\begin{bmatrix} \textit{transitive-verb-lex} & \\ \text{STEM} & \left\langle \text{“chase”} \right\rangle \\ \text{SYNSEM} & \begin{bmatrix} \text{LOCAL} & \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{VAL} & \begin{bmatrix} \text{SUBJ} & \left\langle \boxed{1} \right\rangle \\ \text{COMPS} & \left\langle \boxed{2} \right\rangle \end{bmatrix} \end{bmatrix} \end{bmatrix} \\ \text{LKEYS} & \begin{bmatrix} \text{KEYREL} & \begin{bmatrix} \text{PRED} & \text{“\_chase\_v\_rel”} \end{bmatrix} \end{bmatrix} \end{bmatrix} \\ \text{ARG-ST} & \left\langle \boxed{1}, \boxed{2} \right\rangle \end{bmatrix}$$

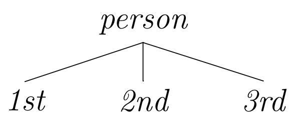
When it is desirable to address a feature several levels deep in the structure without constraining the intermediate levels, the AVM may be abbreviated by leaving out brackets or, if the feature names are unambiguous, by leaving out intermediate names entirely; for example, the above feature structure could also be written as:

(5)
$$\begin{bmatrix} \textit{transitive-verb-lex} & \\ \text{STEM} & \left\langle \text{“chase”} \right\rangle \\ \text{SYNSEM} & \begin{bmatrix} \text{LOCAL}|\text{CAT}|\text{VAL} & \begin{bmatrix} \text{SUBJ} & \left\langle \boxed{1} \right\rangle \\ \text{COMPS} & \left\langle \boxed{2} \right\rangle \end{bmatrix} \\ \text{LKEYS}|\text{KEYREL}|\text{PRED} & \text{“\_chase\_v\_rel”} \end{bmatrix} \\ \text{ARG-ST} & \left\langle \boxed{1}, \boxed{2} \right\rangle \end{bmatrix}$$

Types in HPSG are arranged into a hierarchy via the mechanism of **inheritance**. When one type inherits from another, the inheriting type, also known as the **subtype**, inherits all the features and values specified on its **supertype**, to which other feature specifications may be added. Subtypes are therefore more specific than (or more precisely, at least as specific as) their supertypes. Subtypes can inherit from more than one supertype as long as the supertypes do not specify conflicting feature values; this is known as **multiple inheritance**. Groups of types and their inheritance relationships are often shown as trees (or directed graphs, in the case of multiple inheritance), a notation used throughout this dissertation.

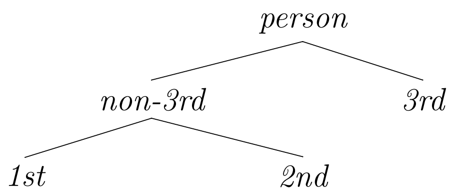
When modeling features whose value is simply one of a set, it is common to use types that have no features or **sorts**. For example, we might model grammatical person using the following type:

(6)



Suppose, however, that it were desirable to specify a sign as having a value of either *1st* or *2nd* (that is, a disjunction). This is not possible using the *person* type above. The usual solution to this problem is to create an additional type in the hierarchy from which the types in the disjunction inherit. In this case, we might add a *non-3rd* type and use it wherever the disjunction is required:

(7)



Unification is the operation by which HPSG combines grammatical entities represented by feature structures to make larger structures. Two feature structures can unify if, at every level of their structure, their values are compatible. Values are compatible if they have the same type, or if one of them is a subtype of the other.

2.1.2 Syntax

Syntax in HPSG is modeled using two different kinds of rules. **Lexical rules** operate on single lexical items, optionally changing their spelling and building up new signs over their inputs. It is possible to think of lexical rules as mapping from one set of lexical types to another, but, in fact, in our trees the signs built up by lexical rules are shown as non-branching nodes above their inputs. **Phrase-structure rules** license the combination of one or more inflected words into a single, larger phrase. These two kinds of rules are prevented from interfering with each other by the INFLECTED feature on signs and by constraints on the input type of lexical rules: a sign that is [INFLECTED −] cannot be a daughter of a phrase-structure rule, and the output of a phrase-structure rule (which is of type *phrase*) cannot be the input to a lexical rule.

HPSG grammars tend to model grammatical complexity using complexity in the lexicon rather than complexity in the syntactic rules. Grammars produced by the customization system may have only a handful of phrase-structure rules, but they will have a number of lexical rules proportional to the number of inflectional morphemes in the language. Our example grammar fragment of English, for example, has four lexical rules—for singular nouns, plural nouns, 3rd person singular verbs, and non-3rd person singular verbs—and four phrase-structure rules—a head-complement rule, a subject-head rule, a head-specifier rule, and a bare-NP rule for nouns without determiners.

The information contained in the feature structure of a sign includes both the syntax and the semantics. Within the syntactic information, we are primarily concerned with the features HEAD and VAL, which together represent information about the syntactic category. The HEAD encodes the part of speech of a sign, and takes values like *noun*, *verb*, *det*, etc (and their subtypes). All rules in HPSG are either **headed** or **non-headed**. In headed rules, one of the daughters is identified as the **head daughter**, and the HEAD feature of the phrase, located at SYNSEM|LOCAL|CAT|HEAD, is identified with the HEAD feature of that daughter. In non-headed rules (which the customization system uses for coordination, for example), the HEAD of the phrase is not identified with the HEAD of any of the daughters, and its value must be stipulated in the rule.

Precisely which words and phrases are allowed to combine to form larger phrases, and in what order, is controlled by **valence lists**, which are located in the feature VAL. Lexical items have a feature ARG-ST for modeling their argument structure. Verbs typically list subjects and complements specified on ARG-ST and nouns list their specifiers. The ARG-ST list of a lexical item may be manipulated by lexical rules; for example, a rule for turning active verbs into passives, which is typically accomplished using a lexical rule in HPSG. The ARG-ST list, however, does not directly participate in syntax. Instead, lexical types map the ARG-ST list, depending on the part of speech, onto several other lists: SUBJ for the subject, COMPS for complements, SPR for a specifier, and SPEC for the specified nominal (the SPR of a noun will typically be its determiner, and the determiner's SPEC will be its noun). In our example grammar fragment of English, lexical types constrain the valence lists as follows:

(8)
$$\begin{bmatrix} \textit{noun-lex} & \\ \text{VAL} & \begin{bmatrix} \text{SUBJ} & \langle\rangle \\ \text{COMPS} & \langle\rangle \\ \text{SPR} & \left\langle \boxed{1} \right\rangle \\ \text{SPEC} & \langle\rangle \end{bmatrix} \\ \text{ARG-ST} & \left\langle \boxed{1} \begin{bmatrix} \text{HEAD} & \textit{det} \end{bmatrix} \right\rangle \end{bmatrix}$$

(9) $$\begin{bmatrix} \textit{determiner-lex} & \\ \text{VAL} & \begin{bmatrix} \text{SUBJ} & \langle\rangle \\ \text{COMPS} & \langle\rangle \\ \text{SPR} & \langle\rangle \\ \text{SPEC} & \left\langle \begin{bmatrix} \text{HEAD} & \textit{noun} \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix}$$

(10) $$\begin{bmatrix} \textit{verb-lex} & \\ \text{VAL} & \begin{bmatrix} \text{SUBJ} & \left\langle \boxed{1} \right\rangle \\ \text{SPR} & \langle\rangle \\ \text{SPEC} & \langle\rangle \end{bmatrix} \\ \text{ARG-ST} & \left\langle \boxed{1} \begin{bmatrix} \text{HEAD} & \textit{noun} \end{bmatrix}, \ldots \right\rangle \end{bmatrix}$$

(11) $$\begin{bmatrix} \textit{intransitive-verb-lex} & \\ \text{VAL} \mid \text{COMPS} & \langle\rangle \\ \text{ARG-ST} & \left\langle [\ ] \right\rangle \end{bmatrix}$$

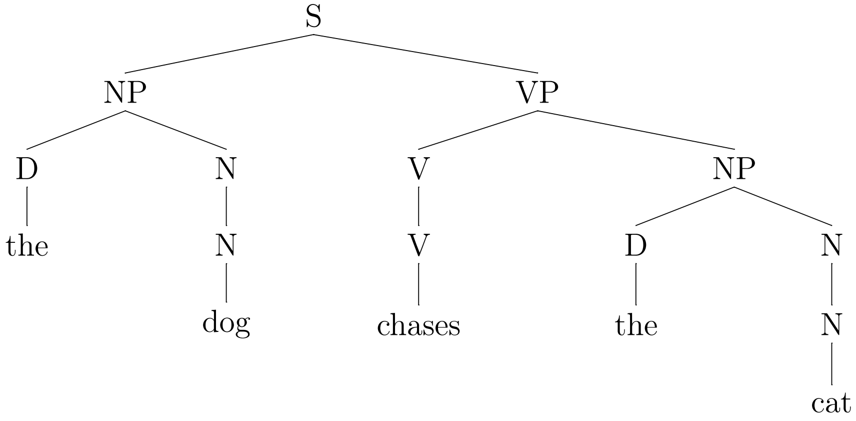
(12) $$\begin{bmatrix} \textit{transitive-verb-lex} & \\ \text{COMPS} & \left\langle \boxed{1} \right\rangle \\ \text{ARG-ST} & \left\langle [\ ],\ \boxed{1} \begin{bmatrix} \text{HEAD} & \textit{noun} \end{bmatrix} \right\rangle \end{bmatrix}$$

Note that the intransitive and transitive verb types are subtypes of *verb-lex*, so they do not need to re-specify the constraint on the subject. Note also that, in addition to the above constraints, the nominal arguments of verbs are constrained to have empty SPR and COMPS lists, which ensures that verbs only combine with full noun phrases (that is, noun phrases whose arguments have been satisfied as described below).

The valence lists on inflected words constrain which rules will apply. Those rules, in turn, "cancel off" the items on the valence lists until all the arguments are satisfied. For example, the head-complement rule will combine any two phrases where the first item on the COMPS list of the head daughter can unify with the non-head daughter, producing a larger phrase whose COMPS list is one item shorter. Similarly, the head-subject rule combines a head daughter with a non-head daughter that can unify with the first (and only) item on its SUBJ list. The head-specifier rule cancels off arguments from both of its daughters: the head-daughter's SPR must unify with the non-head daughter, and the non-head daughter's SPEC must unify with the head daughter.

To see how all these mechanisms license sentences, consider the syntactic analysis produced by our example grammar fragment for the sentence *the dog chases the cat*. First, the parse tree, showing only parts of speech:[1]

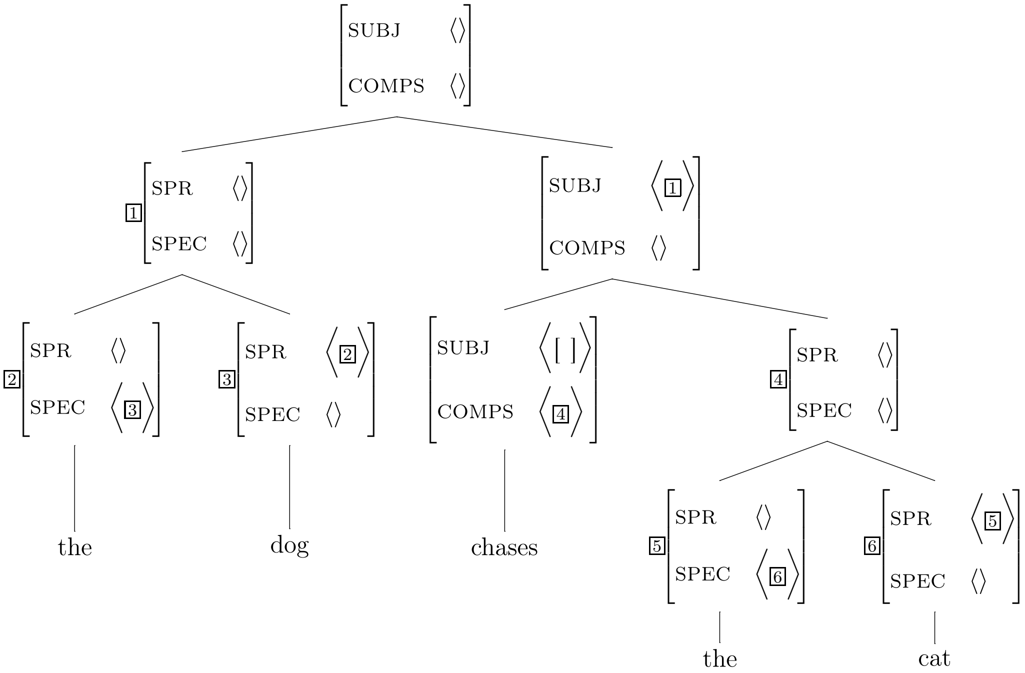
(13)



Notice the non-branching nodes above *dog*, *chases*, and *cat*—they represent the lexical rules that inflect those words for person and/or number. Expanding the feature structures of each node (and omitting the nodes associated with lexical rules to save space) shows how the phrase-structure rules and valence lists interact to license this sentence:

[1] In HPSG, the part of speech labels are nothing more than abbreviations for feature structures. For example, an N is a sign with [HEAD *noun*] and a non-empty SPR list (i.e., it still needs a specifier); an NP, in contrast, has an empty SPR list.

(14)



Each of the branching nodes in (14) is licensed by a phrase-structure rule: the noun phrases *the dog* and *the cat* by the head-specifier rule; the verb phrase *chases the cat* by the head-complement rule; and the whole sentence by the head-subject rule.

2.1.3 Semantics

HPSG also models the semantics of sentences. The grammars produced by the customization system, including our English fragment, use semantic representations compatible with Minimal Recursion Semantics (MRS) (Copestake et al. 2005).

The semantics of a sentence is built up from the semantics of its parts using the feature SYNSEM|LOCAL|CONT|RELS, a list of elementary predications or **relations**. Each relation has the following features: PRED, a string identifying the predicate (e.g., the verb *chase* might have "_chase_v_rel"); LBL, a handle used in modeling scope; and any number of arguments ARG0, ARG1, etc. The only possible values for the ARG0 feature are **variables**, either **individuals** (written as `x1`, `x2`, etc) or **events** (written as `e1`, `e2`, etc). The other arguments may be the variable of another relation—for example, a verb may take the variable of a noun phrase as an argument—or a handle to indicate a scoping relationship. Note that the argument features are simply numbered rather than being given descriptive labels (e.g., AGENT, GOAL).

The identification of the arguments of a verb's elementary predication with the variables of noun phrases is accomplished by a combination of phrase structure rules and lexical entries. For example, the variable of a subject noun is identified with the ARG1 of its verb by the following chain of identities: the noun lexical entry identifies its variable, KEYREL|ARG0, with the feature HOOK|INDEX; the head-specifier rule that builds noun phrases identifies the HOOK of the NP with the HOOK of the noun; the head-subject rule identifies the SYNSEM (which includes the HOOK) of the subject NP with the SUBJ of the VP; the head-complement rule that builds the VP identifies its SUBJ with the verb's SUBJ; and finally the verb lexical item identifies the HOOK|INDEX on its SUBJ with its ARG1.

The Matrix provides types that implement the principle of **semantic compositionality** (also known as Frege's Principle). This is done in two parts. First, the RELS list of every phrase is made up of the concatenated RELS lists of all its daughters, plus any additional semantic contribution made by the phrase. The result, at the root of the tree, is a representation of the meaning of the entire expression. Second, constraints on lexical entries and phrase structure rules interact to identify the relevant argument positions across predications. For example, the variable of a subject noun is identified with the ARG1 of its verb by the following chain of identities: the noun lexical entry identifies its variable, KEYREL|ARG0, with the feature HOOK|INDEX; the head-specifier rule that builds noun phrases identifies the HOOK of the NP with the HOOK of the noun; the head-subject rule identifies the SYNSEM (which includes the HOOK) of the subject NP with the SUBJ of the VP; the head-complement rule that builds the VP identifies its SUBJ with the verb's SUBJ; and finally the verb lexical item identifies the HOOK|INDEX on its SUBJ with its ARG1. (See Flickinger and Bender (2003) for more detail about the Matrix and compositionality.)

Each lexical item specifies the value of its main relation through the feature SYNSEM|LKEYS|KEYREL, whose value is the relation the item exposes for semantic composition. In addition to allowing lexical items to specify their PRED value, this also makes it possible for lexical items to properly "hook up" their arguments. For example, the Matrix type `basic-determiner-lex`, from which language-specific determiner types derive, is specified in such a way that it contributes a quantifier relation to the semantics whose argument (ARG0) is identified with the INDEX in the HOOK of the first item on its SPEC list—that is, with the variable of the noun it quantifies.

Below is the semantic representation of the sentence *the dog chases the cat*:

```
[ LTOP: h1
  INDEX: e2 [ e SF: PROP-OR-QUES ... ]
  RELS <
    [ "_def_q_rel"
      LBL: h3
      ARG0: x4 [ x PNG.PERNUM 3SG ]
      RSTR: h5
      BODY: h6 ]
    [ "_dog_n_rel"
      LBL: h7
      ARG0: x4 ]
    [ "_chase_v_rel"
      LBL: h1
      ARG0: e2
      ARG1: x4
      ARG2: x8 [ x PNG.PERNUM: 3SG ]]
    [ "_def_q_rel"
      LBL: h9
      ARG0: x8
      RSTR: h10
      BODY: h11 ]
    [ "_cat_n_rel"
      LBL: h12
      ARG0: x8 ]>
  HCONS < h5 qeq h7 h10 qeq h12 > ]
```

This representation consists of five relations. There are two quantifier relations associated with the determiners in the noun phrases; notice that each takes the identifier of a noun (`x4` or `x8`) as its sole argument. Each of the two noun relations also takes one of those two identifiers as its argument. The verb relation takes three arguments: `e2` is the event associated with the verb, `x4` is the agent, and `x8` is the patient. Notice that the features on each index (i.e., identifier or event) are shown next to the first occurrence of the index; in this case, the chasing event has the SF (sentential force) of a proposition or question[2], and the two nominal indices have a PERNUM (person/number) value of third person singular.

The LTOP, LABEL, RSTR, BODY, and HCONS features take values that include labels like `h1`, `h2`, etc. These are **handles** and they are used to model scoping. The MRS representation was designed to model the ambiguity often found in sentences with quantification (e.g., *every man loves some woman*) using underspecification of scope relations, but a discussion of how MRS represents scope is beyond the scope of this dissertation.

The analyses presented in chapters 3, 4, and 5 will affect the MRS representations of sentences in two ways. The analyses of case and direct-inverse languages will serve to identify the proper arguments of verbs, licensing just those that are allowed by the grammar of the target language, and thus causing the semantic arguments of verbs to be associated with the proper noun phrases. This results in correct MRS representations without decorating them with syntactic features (i.e., features of HEAD) like CASE. The same is true for syntactic features in chapter 5; in contrast, the analyses of person, number, and gender do result in features appearing directly on event variables in the semantic representation.

[2] The sentential force is left underspecified because we expect that any declarative sentence can also be intoned in such a way to make it a question.

2.1.4 Summary

In this section I have given a brief introduction to HPSG. It must be stated that this was a very simplified treatment of a formalism that has, after all, had several book-length works devoted to it. However, the purpose of this introduction was to give enough of a foundation that the analyses presented later in this dissertation might be understandable to a non-specialist reader.

2.2 Typology

The research described in this dissertation draws heavily on results from the field of **linguistic typology**. I take linguistic typology to be the comparative study of languages and linguistic phenomena, with the aim of discovering how languages are similar, how they differ, and in what ways. In order for languages to be compared, it is necessary that researchers develop ways of characterizing them consistently; that is, ways of categorizing languages as belonging to particular **types**. Typology cannot simply consist of collecting descriptions of languages; instead, the grammars of those languages must be analyzed to discover what grammatical phenomena they display, allowing those phenomena to be compared cross-linguistically. Typology is therefore the comparison of linguistic analyses, rather than of languages themselves.

For several decades, much of the research into linguistic typology has been divided into two strains with differing goals and methodologies, which I will refer to as the Greenbergian and the Chomskyan strains. Below, I give a brief introduction to the historical origins and differences between them.

2.2.1 Greenberg

The modern study of linguistic typology owes much to the work of Joseph Greenberg. Beginning with Greenberg 1963, he pioneered an approach to typology that involved searching a genetically diverse sample of languages for examples of a phenomenon, generally semantically defined, in order to find cross-linguistic patterns. The universals he found were not simple statements about phenomena that never occurred or always occurred; rather, they were **implicational universals** that took the form "given *x* in a particular language, we always find *y*" (Greenberg 1963:73). Greenberg believed that such implicational universals were in fact the only kind of non-vacuous universals that could be found, writing, "what seem to be non-implicational universals about language are in fact tacitly implicational since they are implied by the definitional characteristics of language" (Greenberg 1963:73).

Greenberg demonstrated empirically that the distribution of languages with respect to linguistic phenomena need not be uniform. In his early work, for example, he found in a sample of 30 languages that of the six logically possible basic orders of subject (S), object (O), and verb (V) in declarative sentences, SOV and SVO were about equally common (occurring 11 and 13 times, respectively), VSO was rather less common (occurring 6 times), and the other three were unattested. These results may seem to suggest a true universal of the kind Greenberg denies, namely the statement, "In simple declarative sentences, the subject always precedes the object." However, later results have shown that while VOS, OVS, and OSV languages are rare, they do exist. A survey of 1228 languages by Dryer (2008) included in the World Atlas of Language Structures found the distribution of basic word orders shown in Table 2.1.

Table 2.1: Basic word orders across languages

Word order	**count**
SOV	497
SVO	436
VSO	85
VOS	26
OVS	9
OSV	4

Greenberg's methods produce two kinds of statements about language: implicational universals ("if x then y") and statistical tendencies ("x is nearly always true"). Producing statements of this kind requires large number of languages to be examined, and so Greenbergian typologists are interested in the problem of **language sampling**: given the vast variety and complex interrelationships, both genetic and areal, between the world's languages, how can a sample of languages be selected that is, as far as possible, unbiased and representative of human language as a whole.

The Greenbergian approach to typology, then, is data-driven, shallow, and broad: data-driven because it is based on empirical samples of languages; shallow because the phenomena compared are generally quite "surfacy" for practical reasons (namely the large number of languages sampled and the great effort that would be required to develop a detailed analysis of each language); and broad because it allows the surveying of large samples of languages to probe for exceptions.

2.2.2 Chomsky

A focus on linguistic typology is an important feature of the work of another pioneering linguist, Noam Chomsky. One of Chomsky's ongoing research aims has been an attempt to characterize what he calls Universal Grammar (UG): the genetic endowment shared by all humans that shapes and constrains our language faculty. His work on syntax attempts to provide a formal framework for analyzing human languages that is informed by what is known about their range of variation.

Chomsky's early approach to language universals reached its fullest expression in his Government and Binding (GB) framework (Chomsky 1981). It was typological in its conception. He wrote that he expected to find, "a highly structured theory of UG based on a number of fundamental principles that sharply restrict the class of attainable grammars and narrowly constrain their form, but with parameters that have to be fixed by experience." (Chomsky 1981:3–4) In other words, UG was conceived of as consisting of two kinds of entities: **principles**, which are true universals that all natural languages must follow, and **parameters**, which are "switches" controlling the variation between languages. For this reason, the GB framework is sometimes also referred to by the alternate label Principles and Parameters (P&P). In this framework, UG was understood as a language-learning machine whose function is to help children to acquire language by setting the values of parameters. Chomsky further wrote:

> A valid observation that has frequently been made (and often, irrationally denied) is that a great deal can be learned about UG from the study of a single language, if such study achieves sufficient depth to put forth rules or principles that have explanatory force but are underdetermined by evidence available to the language learner.... Similarly, study of closely related languages that differ in some clustering of properties is particularly valuable for the opportunities it affords to identify and clarify parameters of UG that permit a range of variation in the proposed principles. Work of the past several years on the Romance languages...has exploited these possibilities quite effectively. (Chomsky 1981:6)

Following these ideas, Chomskyan linguists strove to design a theory of language that accounted for the observed variation in human languages, including fine variation between closely related languages.

It should be noted that Chomsky's approach to typology has evolved over time. Since the mid-1990s, he has championed what he calls the Minimalist Program (Chomsky 1995). As part of this program, Chomsky has worked to pare away at the content of the theory of language, reducing it to its conceptually necessary core. This has led Chomsky and others to question how much of the cross-linguistic similarity between languages can actually be ascribed to UG. On the subject of the innate core of the language faculty (FLN, in their terminology), Chomsky et al. (2002:1571) wrote, "...we propose in this hypothesis that FLN comprises only the core computational mechanisms of recursion as they appear in narrow syntax and the mappings to the interfaces.” In other words, it may be that our genetic endowment for language consists only of recursion, with other features of the language faculty resulting from interfaces with other mental systems. More recently, Chomsky (2005:11–12) put this idea in explicitly Minimalist terms, writing that the evolutionary “Great Leap Forward” might have been “that the brain was rewired, perhaps by some slight mutation, to provide the operation Merge” (where Merge is the fundamental operation that allows recursion and movement). This drastic narrowing of the conception of UG has required a rethinking of the status of principles and parameters. Boeckx (2008), for example, has suggested a formulation of parameters in which they reside outside of narrow syntax and only affect the presence or behavior of features.

Nonetheless, the P&P approach to typology was influential for many years, and guided the methods and focus of typological research in the field of generative syntax. The Chomskyan approach to typology, compared with Greenberg’s approach, has been theoretically focused, narrow, and deep: theoretically focused because the primary aim has been, not directly to improve the understanding of language variation, but rather to refine the content of UG, and hence of the grammatical theory; narrow because the analyses required take relatively more time to develop, which restricts the number of languages that can be examined; and deep because it has relied on thorough, detailed analyses of linguistic phenomena rather than merely observing surface variation.

The Greenbergian and Chomskyan strains have often been seen as opposed. This has been true to some extent historically, but I would suggest that computational methods, and the Grammar Matrix and its customization system in particular, can provide a way to bridge the gulf between deep and shallow, between narrow and broad, and between theory-internal and theory-neutral. This notion is discussed in more detail in §7.1.

2.2.3 Surveys and Databases

The contemporary field of linguistic typology acknowledges the need to encompass the whole range of linguistic variation. Many researchers accomplish this by conducting Greenberg-style cross-linguistic surveys. In a survey, some phenomenon is chosen, defined, and then examined in a sample of languages chosen with an eye towards genetic and areal diversity. Broad typological surveys serve three purposes. First, their breadth prevents the field from focusing too narrowly on a few well-known languages and the phenomena they exemplify. Second, surveys can turn up unknown variants of phenomena, thus expanding their known range. Third, surveys allow typologists to characterize the distribution of phenomena across the world languages in a principled and statistically valid way.

The output of a typological survey is often organized and published as a database, allowing later researchers access to the results, and sometimes allowing complex queries on the survey results. Some examples of typological databases include the Stanford Phonology Archive (SPA) (Vihman 1977), the UCLA Phonological Segment Inventory Database (UPSID) (Maddieson 1980), the various surveys that make up the World Atlas of Language Structures (WALS) (Haspelmath et al. 2005), and the Universals Archive at the Konstanz University (an archive of typological generalizations culled from the typological literature) (Plank and Filimonova 2000).

2.2.4 Linguistic Questionnaires

The customization system is based on a questionnaire that prompts the user-linguist to describe a single natural language. Several other language questionnaires have been developed, although each has a rather different focus from the customization system questionnaire.

The Lingua Descriptive Studies Questionnaire (Comrie and Smith 1977) is an extensive questionnaire covering a wide range of linguistic phenomena. Like the customization system questionnaire, it was intended to prompt a linguist to provide answers about a natural language under study. The Lingua Questionnaire was intended primarily as a checklist of phenomena to be considered during the preparation of a written grammar of a language. Its questions, therefore, are open-ended and intended to be answered in prose, with no limit on length. The customization system, in contrast, must ask questions whose answers are simple enough to be interpretable by software. Nonetheless, the breadth and completeness of the Lingua Questionnaire serve as a (very ambitious) target for the customization system.

The AUTOTYP project (Bickel and Nichols 2002) is a project aimed at discovering the distribution and typological range of various linguistic phenomena. As part of its methodology, researchers fill out questionnaires about languages under study, including questionnaires about the phonological and grammatical domains. The results of these questionnaires are collected into databases, although these are unfortunately not published. Like the Lingua Questionnaire, the AUTOTYP questionnaires consist of open-ended questions aimed at eliciting a detailed description of phenomena in a language. In contrast, the customization system questionnaire attempts to provide as full as possible a range of choices for certain phenomena, rather than prompting the user-linguist to describe new possibilities.

The Expedition Project (Nirenburg 1998) was an ambitious effort aimed at quickly producing machine translation systems for low-density languages. The project as described would have prompted non-linguists for information about their language via the Boas system, and from their answers produced a detailed morphological analysis and a characterization of the language by a number of "parameters" such as its gender and number categories. This project apparently stalled; the available publications describe the system as it stood in 1999 and 2000, and the project's online demo page still says "Coming Soon!". Of the systems described here, Expedition/Boas is probably closest to the customization system, although as mentioned above it was aimed at non-experts rather than trained linguists.

The AVENUE project (Probst et al. 2002) was another project intended to produce machine translation systems for languages with few resources for natural language processing. The data required to train a statistical machine translation system is typically a word-aligned bilingual corpus. AVENUE developed an elicitation tool that could speed the creation of such a corpus for a language for which none was available. The tool was aimed at a user who was bilingual and literate, but not necessarily experienced with linguistics. Unlike Expedition, the AVENUE project reached its goals, producing a Spanish-Mapudungun translation system.[3] AVENUE's broad goals were similar to those of the Matrix project, aiming to bring computational linguistic resources to bear on low-resource languages; however, the focus of the project was on corpus construction rather than on linguistic description, so the AVENUE elicitation tool does not provide much insight into the construction of a typological questionnaire.

In summary, although there have been previous efforts aimed at creating typological questionnaires about human languages, none of those efforts has provided much guidance toward the design of the questionnaire needed by the Matrix customization system, because these questionnaires were either too open-ended (the Lingua Questionnaire), aimed at a different audience (Expedition), or intended to collect a different kind of data (AVENUE).

2.2.5 Summary

In this section, I have briefly introduced the major trends in the field of linguistic typology, and described cross-linguistic surveys and typological questionnaires, both of which, as we shall see, are important concepts underlying the Matrix and the customization system.

[3] http://amauta.lti.cs.cmu.edu/mapudungun/index.html

2.3 Multilingual Grammar Engineering

Grammar engineering is the implementation in software of grammars of natural language, regardless of the particular formalism or development environment. The system described in this dissertation is a component of the Grammar Matrix, which is an instance of multilingual grammar engineering, but there have been other research projects directed at similar and related aims. In the sections that follow, I describe some notable examples, paying special attention to how they manage multilingual generalizations.

2.3.1 ParGram

The ParGram project (Butt et al. 2002, King et al. 2005) consists of a set of grammars, each developed by a separate group of researchers, all in an LFG framework and implemented using the XLE platform. Languages covered include English, French, German, Japanese, Norwegian, and Urdu. LFG analyses have two levels of representation: the constituent structure or c-structure and the function structure or f-structure. C-structures mark the grouping of words into phrases and sentences. F-structures "reflect a more language-independent analysis" (King et al. 2005:142–3), encoding "grammatical functions, syntactic features, and predicate-argument (dependency) relations conveyed by the sentence" (King et al. 2005:140). In the ParGram project, the researchers meet twice yearly in order to synchronize the features they use in their f-structures. New features and new values for existing features are proposed and the details of their implementation hashed out, and when changes are propagated, the existing grammars are brought up to date with those changes.

ParGram, therefore, has an explicit process whereby language-independent generalizations are made and revised in the light of newly implemented phenomena. These generalizations can then be used to give a head start to new grammars.

2.3.2 Natural Language Generation

Bateman et al. (2005) describe work in the field of natural language generation (NLG). The task of NLG is the generation of text that is appropriate to a particular domain, such as instructional text, weather reports, or news reports. One particular system they describe, known as Agile, was capable of generating Bulgarian, Czech, and Russian. Later work extended this to cover German, Dutch, Spanish, French, Chinese, with smaller experiments in Greek and Japanese. These systems are aware of various registers and styles. For example, when generating headings and subheadings in an instructional text, the system will produce the proper form for the target language, as shown in Table 2.2.[4]

Table 2.2: Headings generated by Agile

Language	"To draw a polyline"		
Bulgarian	*Chertane*	*na polilinija*	
	drawing.NOM	of polyline	
Czech	*Nakreslenií*	*křivky*	
	drawing.NOM	polyline.GEN	
Russian	*Chtoby*	*narisovatj*	*poliliniju*
	in.order.to	draw.INFIN	polyline.ACC

This NLG work was implemented in the KPML grammar development environment. The project started with an analysis of one language (English), then added more. Where languages used similar structures (e.g., a noun phrase) for similar meanings, constraints could be stated once for all languages; where structures differed, additional constraints conditioned on the output language were used. Over time, then, as languages were added, the system became increasingly complex, but at any point, an apparently language-independent core could be discerned, namely the constraints that were not conditioned on a particular language.

[4] The glosses in Table 2.2 have been slightly modified to conform to the Leipzig glossing rules.

2.3.3 Referential Properties of Nominals

Borthen and Haugereid (2005) describe an HPSG analysis of several properties of nominals, namely cognitive status, specificity, partitivity, and whether the nominal has universal interpretation. Their analysis is based on an examination of how these properties are expressed, and what values they can take, in Norwegian, English, Dutch, and Turkish. They noted a lack of a standardized implementation of these properties in existing HPSG grammars of various languages, and proceeded to design a general cross-linguistic solution, which involved the proposal of four new features: COGN-ST, PART, SPECI, and UNIV.

This approach to multilingual grammar engineering, where a phenomenon is studied in a range of languages and then a general analysis is designed, is closely related to the approach in the LinGO Grammar Matrix and the work described in this dissertation. In fact, the analysis of cognitive status was later adapted from this work and included in the Matrix (Bender and Goss-Grubbs 2008).

2.3.4 Automatic Acquisition of Grammars

Cahill et al. (2005) describe a system for automatically acquiring LFG and PCFG grammars from treebanks of natural languages. The algorithms they describe have been applied to English (using the Penn-II Treebank) and to German (using the Tiger Treebank), resulting in broad-coverage grammars for those languages capable of handling unseen newspaper text with f-scores comparable to the best hand-crafted grammars. The same techniques were also applied to Chinese (using the Penn Chinese Treebank) and Spanish (using the CAST3LB Treebank).

This work is certainly related to multilingual grammar engineering in that it is capable of producing grammars of various languages, but it differs from the other projects described here in the part of the system that is reusable cross-linguistically. In the other projects, the multilingual generalizations were of a linguistic nature:

features, types, structures, etc. In Cahill et al.'s work, the components that are shared cross-linguistically are the algorithms and tools for lexical and rule acquisition. This makes their work different in focus and in kind from the work described in this dissertation.

2.3.5 Porting a Grammar of a Closely-Related Language

Both Smrž (2005) and Kim et al. (2003) describe efforts to take an existing grammar in one language and modify it as necessary to handle a second, closely related language. In Smrž's work, a grammar of Czech written in a metagrammar format that was compiled to produce the actual grammar was used as a basis for the implementation of a grammar of Russian. Kim et al.'s project was the implementation of a grammar of Korean based on an LFG grammar of Japanese. In both cases, the idea was to reuse elements of the analysis that applied to both languages, while replacing elements that did not. The result of this process is a set of grammatical components that are shared between the two languages.

It is likely that other instances of this style of multilingual grammar engineering have taken place—a working, existing grammar is a natural starting point for the development of a second. In fact, one of the original rationales for the Grammar Matrix described below was the idea that many languages will share such a set of reusable components.

2.4 The LinGO Grammar Matrix

The LinGO Grammar Matrix (Bender et al. 2002) is an attempt to provide a typologically-informed foundation for building grammars of natural languages in software. It includes a set of predefined types for lexical and syntactic rules, and a hierarchy of lexical types. It also provides a detailed syntax-semantics interface consistent with Head-driven Phrase Structure Grammar (HPSG) and Minimal Recursion Semantics (MRS) (Copestake et al. 2005). Grammars based on the Matrix are expressed in the Type Description Language (TDL) (Krieger and Schäfer 1994) as interpreted by the Linguistic Knowledge Building system (LKB) (Copestake 2002), a software tool for developing constraint-based grammars. The primary purpose of the Matrix is to allow the rapid creation of new grammars based on insights gained in the implementation of previous grammars.

This section will discuss the history of the Grammar Matrix project, the contents of the Matrix, the conceptual division of the Matrix into core and libraries, and the design of the customization system and its associated questionnaire.

2.4.1 History

The Matrix grew out of two projects: the LinGO project's English Resource Grammar (ERG) (Copestake and Flickinger 2000) and Jacy, a grammar of Japanese (Siegel and Bender 2002). Each of these was a large, broad-coverage, single-language grammar in HPSG and expressed in TDL. Both are ongoing projects. The ERG is a project at Stanford's CSLI LinGO Lab. It currently contains about 4,000 types and 30,000 lexical items, and its operation can be seen by visiting its online demo site, http://erg.emmtee.net/. The Jacy grammar is currently maintained by Francis Bond at NiCT in Japan, and it contains about 2,000 types and 50,000 lexical items. An online demo of Jacy is available at http://uakari.ling.washington.edu:8103/logon.

Although English and Japanese are genetically unrelated languages, the ERG and Jacy share some similarity in structure. To some extent, this is a result of their both being implemented in HPSG and compatible with the LKB, but the ERG was also used as a resource during the development of Jacy, especially for the semantic representation. Large grammar engineering projects require a sizable investment of time and resources; for example, Bender et al. (2002) note that the large HPSG grammars of English, German, and Japanese "represent between 5 and 15 person years of research efforts, and comprise 35–70,000 lines of code." Furthermore, since each grammar is typically developed in isolation, it can be difficult to extract, or even to understand, the complex interacting analyses they contain. These facts make it challenging to create a grammar of a new language based on either the ERG or Jacy. To address this, Bender et al. (2002) embarked on the development of the Grammar Matrix, a common foundation for grammars that drew on insights gained during the development of the ERG and Jacy.

The Matrix was intended to benefit implementers of natural language grammars. By providing them with a set of types that might be expected to be useful in any natural language grammar, the Matrix helps grammar-writers avoid re-inventing the grammar engineering wheel. This benefit has been put into practice. Several research projects have built grammars using the Matrix as a foundation, including grammars of Norwegian (Hellan and Haugereid 2003), Modern Greek (Kordoni and Neu 2005), Spanish (Marimon et al. 2007), and the aforementioned Jacy grammar of Japanese, into which the Matrix was eventually incorporated. In addition, the Matrix has served as the primary resource for students in grammar engineering classes taught annually since 2004 at the University of Washington and other institutions (Bender 2007, http://courses.washington.edu/ling567/). In these classes, students choose natural languages (often ones with which they are not familiar), seek out printed sources (and native-speaker informants, when available) for those languages, then build software grammars for them. The grammars include support for a moderate number of phenomena, typically including basic word order, case, agreement, modification, argument optionality, discourse status, different clause types and illocutionary force, negation, raising and control, and tense and aspect. Based on the Grammar Matrix, students in the UW class have built grammars of 61 genetically diverse languages, including Ainu [ain], American Sign Language [ase], Moroccan Vernacular Arabic [ary], Armenian [hye], Bangla [ben], Basque [eus], Cantonese [yue], Classical Japanese [jpn], Classical Nahuatl [nci], Czech [ces], Dutch [nld], Old English [ang], Esperanto [epo], Farsi [pes], Finnish [fin], French [fra], Greek [ell], Haida [hdn], Haitian Creole [hat], Hausa [hau], Hawaiian [haw], Hebrew [heb], Hindi [hin], Hixkaryána [hix], Huallaga Quechua [qub], Hungarian [hun], Icelandic [isl], Indonesian [ind], Inupiaq [esi], Irish Gaelic [gle], Italian [ita], Japanese [jpn], Kannada [kan], Korean [kor], Latin [lat], Lillooet [lil], Malayalam [mal], Mandarin [cmn], Mandinka [mnk], Modern Standard Arabic [arb], Mongolian [khk], Navajo [nav], Norwegian [nor], Polish [pol], Portuguese [por], Romanian [ron], Russian [rus], Sanskrit [san], Spanish [spa], Swahili [swh], Swedish [swe], Taiwanese [nan], Tagalog [tgl], Tamil [tam], Thai [tha], Tok-Pisin [tpi], Turkish [tur], Uzbek [uzn], Western Sisaala [ssl], Welsh [cym], and Zulu [zul].

The process of building multiple grammars on a common foundation has several benefits. Obviously, from the point of view of the grammar-writers, not having to re-create a facsimile of the contents of the Matrix from scratch saves time. There are benefits from the point of view of the Matrix project as well. Every time someone implements a grammar based on the Matrix, it tests the correctness of the Matrix, which is, after all, a software system and therefore subject to false assumptions and bugs.

Furthermore, over time the project accumulates a collection of grammars that are all implemented similarly, at least to the extent supported by the Matrix. These grammars form a source of potential new types in the Matrix. If the Matrix lacks support for some phenomenon that all the students' grammars seem to require, that argues for the inclusion of a suitably cross-linguistic analysis of that phenomenon in the Matrix. Their common foundation means the grammars are also interoperable with each other in a practical way, as demonstrated by an event known as the "Machine Translation Extravaganza". Each time the UW grammar engineering course is complete, on the last day the grammars created by the students are hooked up pairwise, with one as the source and the other as the target, and, as long as the rules for producing harmonized semantic representations have been followed, sentences in the source language can be translated into sentences in the target language via semantic transfer (Lønning et al. 2004).[5]

2.4.2 Core and Libraries

In its conception, the Matrix is intended to provide a foundation for building grammars of all natural languages. However, building such a foundation could mean two different things. In the first, the Matrix would contain *only* those rules and mechanisms that are needed by *every* natural language. Under this interpretation, the contents of the Matrix should be restricted to types implementing phenomena that are known to be truly universal.

However, taking this tack would negate much of the intended utility of the Matrix. The great majority of linguistic phenomena that the Matrix ought to support are widespread, but not universal; that is, there are many phenomena that occur in a non-trivial fraction of the world's languages, but that also do not occur in a non-trivial fraction of languages. Examples are plentiful, and include case, agreement, person,[6] number, and gender. Leaving such phenomena unsupported by the Matrix would make it a far less broad and far less useful foundation for building grammars.

The desire to support non-universal phenomena led to a second, broader organizing principle of the Matrix: that it should contain support for non-universal phenomena as well as universal, and that the non-universal parts of the Matrix should be organized into **libraries**,[7] one for each phenomenon, as described by Bender and Flickinger (2005). The separation of types implementing universal and non-universal phenomena divides the Matrix into two parts: the **core Matrix** and the **Matrix libraries**. The core Matrix, represented by the contents of the file `matrix.tdl` (and a few other files,

[5] Note that such translation relies on the simplifying assumption of the existence of a lexical interlingua, which in this case is guaranteed by having the students agree to use the same set of (English) predicate names in their semantic representations.

[6] Although it seems likely that all languages have a way of referring to discourse participants, there are a few in which the existence of person markers has been questioned, including Thai, Burmese, Vietnamese, and Japanese (Siewierska 2004:8).

[7] Originally called **modules**.

most of which support interaction with the LKB), contains types that are expected to be used in every language, while libraries contain types used in implementing non-universal phenomena.

Before moving on to a discussion of Matrix libraries, there is one last question to be resolved concerning the division between core and libraries. Suppose we have a group of closely-related types that support a phenomenon that all languages are expected to have, but not all of those types are expected to be used in every grammar. For example, consider the types `head-initial` and `head-final` in the Matrix, which implement word-order alternations: phrases where the head comes before the complement are subtypes of `head-initial`, phrases where the head follows the complement are subtypes of `head-final`, and phrases where head-complement ordering is unrestricted are modeled by a pair of rules, one deriving from `head-initial` and one from `head-final`. It seems likely that all complement-taking heads in all languages must choose one of these options when implementing word-order,[8] but it is possible that there exist languages that are either consistently head-initial or head-final for all types of phrases. If such a language exists, one of the types `head-initial` and `head-final` will go unused in a grammar that models it. It could be argued on this basis that these types are not truly universal, and so do not belong in the core Matrix, but rather in a library. To avoid paring the core Matrix down too far, we have chosen instead to retain in the core types that may not be required for every Matrix-based grammar, but that exist in paradigms of closely related types of which *at least one* is expected to be used in every grammar.

It should be mentioned that these principles for dividing the Matrix into core and libraries are a rather late development, first described in this dissertation, and that the actual contents of the core in `matrix.tdl` at the time of writing include some types that are clearly not required in all grammars and should be removed

[8] But see Bender (2008b) for discussion of a Matrix-based grammar that approaches the combination of heads and complements in a radically different way.

to libraries. The types associated with the current implementation of coordination (Drellishak and Bender 2005), for example, are not expected to be used in all grammars, and yet they currently reside in `matrix.tdl`. The separation of core and libraries is therefore ongoing work in the Matrix project.

2.4.3 Matrix Customization System

As described above, there are linguistic phenomena that are widespread, but not universal, but for which we intend to provide support in the Matrix via libraries. The initial effort at adding libraries to the Matrix, described by Bender and Flickinger (2005), included several phenomena: varying word orders, matrix yes-no questions, and sentential negation. However, these initial libraries were not based on thorough surveys of the typological literature about the phenomena to be covered. Later, a Matrix library based on such a survey was added, a library for coordination (Drellishak and Bender 2005).

A Matrix library is more than just a collection of types. Most linguistic phenomena are not simply present or absent; they vary in their expressions and in the complexity of the analyses required. It is not enough, for example, simply to know that the target language has coordination. It is necessary also to know, among other things, what types of phrases can be coordinated, how those phrases are marked, and what patterns of marking appear in the language. Supporting a linguistic phenomenon, therefore, requires eliciting the answers to such questions from the grammar-writer.

The component of the Matrix system that elicits these answers and, based on them, creates a grammar that models the language described is called the **customization system**. It is a software system that presents the grammar-writer with a detailed, World Wide Web-based typological questionnaire, interprets the answers without human intervention, and produces a grammar in the format expected by the LKB. This dissertation does not describe in great detail the internal architecture of the customization system,[9] but some aspects of the system and its terminology are relevant to readers of this dissertation. Foremost among these are the web-based questionnaire and choices files, which are described in the next section.

2.4.4 The Questionnaire

Before designing a questionnaire, it is critical to decide who will be filling it out. The intended audience of the Matrix customization system is linguists who want to create computational grammars of natural languages. The questionnaire, therefore, is aimed at a user who is familiar with technical linguistic terminology and comfortable with the idea of building a formal grammar that models a natural language. It freely uses technical linguistic terminology, but avoids, when possible, mentioning the internals of the grammar that will be produced, although a user who intends to modify the grammar will of course need to become familiar with HPSG and TDL before doing so.

The questionnaire[10] is presented to the user-linguist as a series of connected web pages. The first page the user sees (the "main page") contains some introductory text and directs the user to the sections of the questionnaire ("subpages") via a set of hyperlinks. Each subpage contains a set of related questions that (with some exceptions) covers the range of a single Matrix library. The main page of the questionnaire at the time of writing, including the list of subpages, is shown in Figure 2.1.

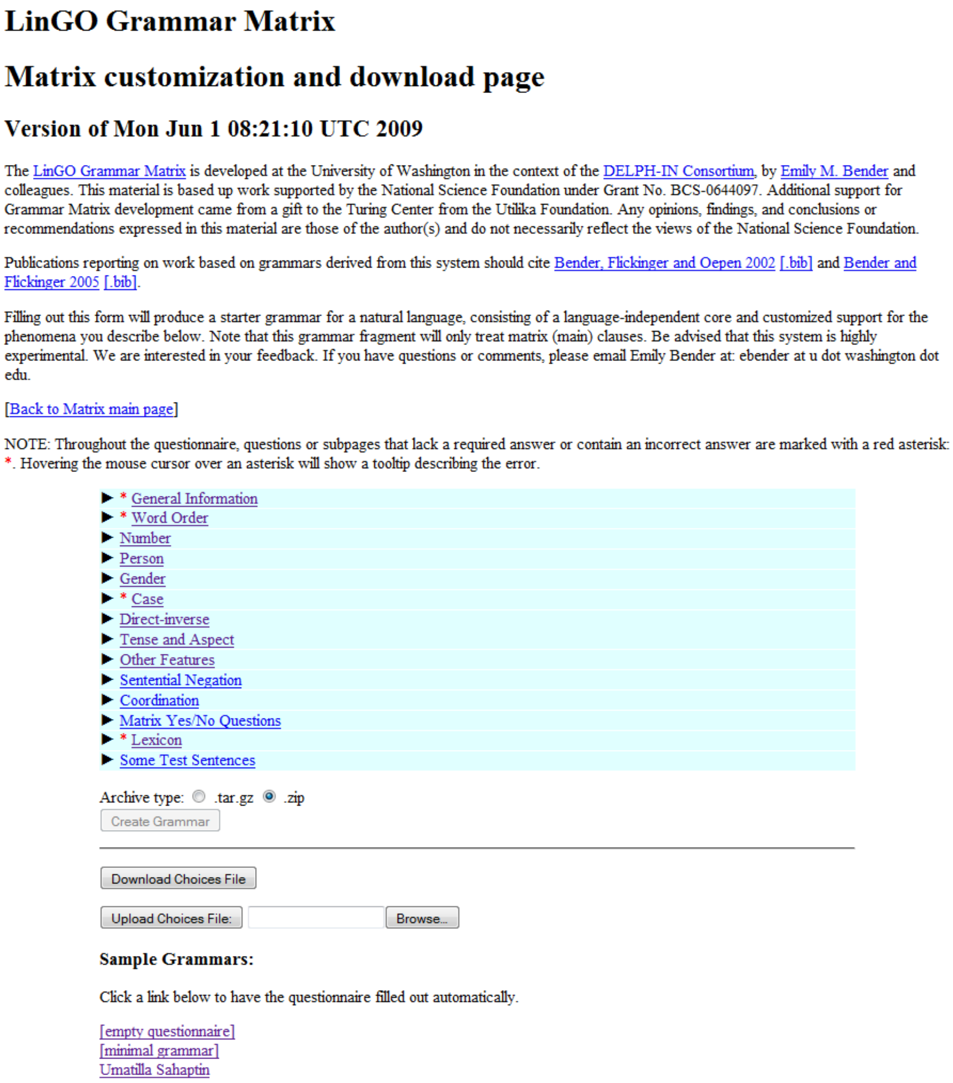
The actual questions in the questionnaire are represented by HTML (Hypertext Markup Language) form fields. The customization system uses five different kinds of form fields:

1. **Text fields**, which may contain arbitrary Unicode text

2. **Check boxes**, which represent binary, yes/no choices

[9] For such details, see the source code, available at https://lemur.ling.washington.edu/trac/matrix, and the documentation at http://wiki.delph-in.net/moin/MatrixDevTop

[10] http://www.delph-in.net/matrix/sfddiss/



LinGO Grammar Matrix

Matrix customization and download page

Version of Mon Jun 1 08:21:10 UTC 2009

The LinGO Grammar Matrix is developed at the University of Washington in the context of the DELPH-IN Consortium, by Emily M. Bender and colleagues. This material is based up work supported by the National Science Foundation under Grant No. BCS-0644097. Additional support for Grammar Matrix development came from a gift to the Turing Center from the Utilika Foundation. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Publications reporting on work based on grammars derived from this system should cite Bender, Flickinger and Oepen 2002 [.bib] and Bender and Flickinger 2005 [.bib].

Filling out this form will produce a starter grammar for a natural language, consisting of a language-independent core and customized support for the phenomena you describe below. Note that this grammar fragment will only treat matrix (main) clauses. Be advised that this system is highly experimental. We are interested in your feedback. If you have questions or comments, please email Emily Bender at: ebender at u dot washington dot edu.

[Back to Matrix main page]

NOTE: Throughout the questionnaire, questions or subpages that lack a required answer or contain an incorrect answer are marked with a red asterisk: *. Hovering the mouse cursor over an asterisk will show a tooltip describing the error.

Archive type: ○ .tar.gz ● .zip

Sample Grammars:

Click a link below to have the questionnaire filled out automatically.

[empty questionnaire]
[minimal grammar]
Umatilla Sahaptin

Figure 2.1: The customization system questionnaire main page

3. **Radio buttons**, which allow the selection of one item from a group of alternatives

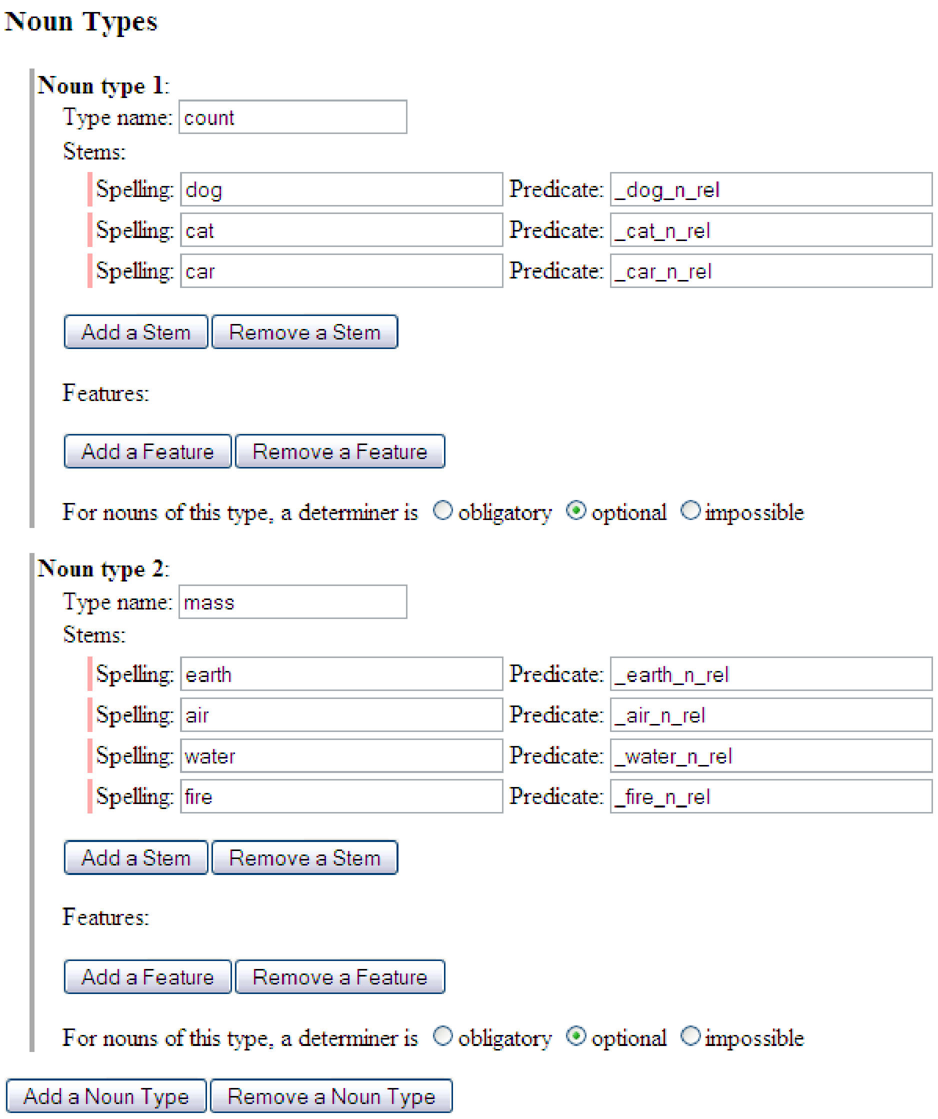
4. **Drop-downs**, which also allow the selection of one item from a group of alternatives, but are more visually compact than radio buttons

5. **Multi-select drop-downs**, which are similar to drop-downs but which allow the user to select more than one item from a group of alternatives

In earlier versions of the customization system (Bender and Flickinger 2005, Drellishak and Bender 2005), the number of form fields in the questionnaire, and therefore the number of questions, was fixed. As part of my work, I have improved the questionnaire to include repeatable sections called **iterators** that allow an arbitrarily large number of similar questions to be answered by the user-linguist. To see how iterators work, consider the excerpt from the Lexicon subpage shown in Figure 2.2. This excerpt contains questions that allow the description of noun classes in the lexicon, and is filled out with answers that would be appropriate for (a fragment of) English. Note the two nearly-identical sections labeled “Noun type 1” and “Noun type 2”. When the Lexicon subpage first loads, only one such section appears; if the user-linguist wishes to add additional noun classes, the “Add a Noun Type” button can be pressed to do so. The system places no limit on the number of lexical classes that can be defined in this way, allowing the description of a lexicon of unbounded size.

Note further the “Stems” section inside each noun type, which allows the description of any number of stems for each noun type—for example, having defined a lexical type for common nouns, the user-linguist may then include stems for *house*, *dog*, and *cat* in that type. The spelling of these stems (and of inflectional morphemes) in the questionnaire can include any character in the Unicode character set, including symbols in the International Phonetic Alphabet. For each stem, the user-linguist must also provide a predicate, which is a string used to represent the meaning of the associated lexical item in the semantic representation of a sentence.

Another result of my work is that the Lexicon subpage now allows the description of inflectional morphology on the various lexical types. Inflection is modeled as a set of **slots** that contain **morphemes**. A slot is defined to take one or more **inputs** that



Noun Types

Noun type 1:
Type name: count
Stems:
Spelling: dog Predicate: _dog_n_rel
Spelling: cat Predicate: _cat_n_rel
Spelling: car Predicate: _car_n_rel

Features:

For nouns of this type, a determiner is ○ obligatory ● optional ○ impossible

Noun type 2:
Type name: mass
Stems:
Spelling: earth Predicate: _earth_n_rel
Spelling: air Predicate: _air_n_rel
Spelling: water Predicate: _water_n_rel
Spelling: fire Predicate: _fire_n_rel

Features:

For nouns of this type, a determiner is ○ obligatory ● optional ○ impossible

Figure 2.2: Noun types in the customization system questionnaire

it attaches to; each input is either a lexical type (for inflection that attaches directly to a stem) or another slot (for inflection that attaches further from the stem). Within each slot in the questionnaire, the user-linguist can define any number of morphemes that appear in the slot, including the spelling of the morpheme and the values of any features it specifies.

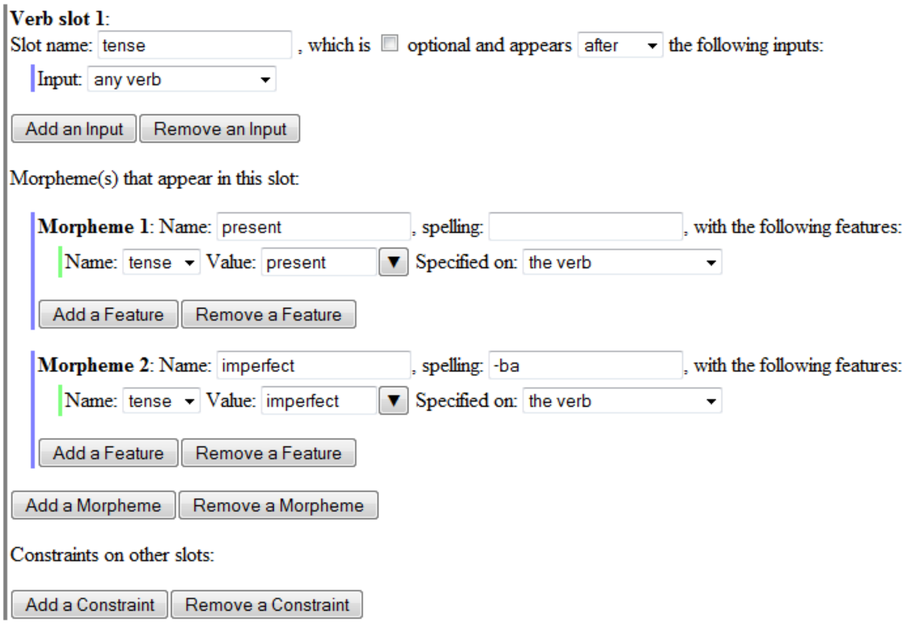
The HPSG implementation of inflectional slots and morphemes is based on work by O'Hara (2008). In this framework, each slot is a lexical rule that specifies the type of its input using the DTR feature—this mechanism determines where the slot may appear within the inflected word. Every morpheme that appears in a slot is a lexical rule that derives from the slot's rule. This ensures that all morphemes in a slot appear in the same place, but each morpheme can specify its own spelling and features.

Table 2.3: Paradigm of the Latin verb *amare*

Number	**Person**	**Present Tense**	**Imperfect Tense**
singular	1st	*amo*	*ama-ba-m*
	2nd	*ama-s*	*ama-ba-s*
	3rd	*ama-t*	*ama-ba-t*
plural	1st	*ama-mus*	*ama-ba-mus*
	2nd	*ama-tis*	*ama-ba-tis*
	3rd	*ama-nt*	*ama-ba-nt*

To illustrate how slots and morphemes are described in the questionnaire, consider the paradigm of the Latin verb *amare* 'to love' shown in Table 2.3. If we make a simplifying assumption and regularize the first person singular as *ama-m*, then this paradigm can be modeled using two slots. The first slot, a suffix, takes a verb stem as its input and marks it for tense, either with *-ba* for the imperfect or with ∅ for the present. The second slot, also a suffix, takes the first slot as its input (so it attaches further from the stem) and contains one of the six person/number suffixes. Portions of the Lexicon page filled out to represent these slots are shown in Figures 2.3 and 2.4. Figure 2.3 shows the section describing the tense slot and its morphemes. Figure 2.4 shows the person/number slot; notice also that it shows one of the multi-select drop-downs in the "open" state, displaying all of the person/number choices.

The customization system also makes available three additional constraints that may be placed on inflectional slots: slot A may require that slot B also appears in



Verb slot 1:
Slot name: tense , which is □ optional and appears after the following inputs:
Input: any verb

Morpheme(s) that appear in this slot:

Morpheme 1: Name: present , spelling: , with the following features:
Name: tense Value: present Specified on: the verb

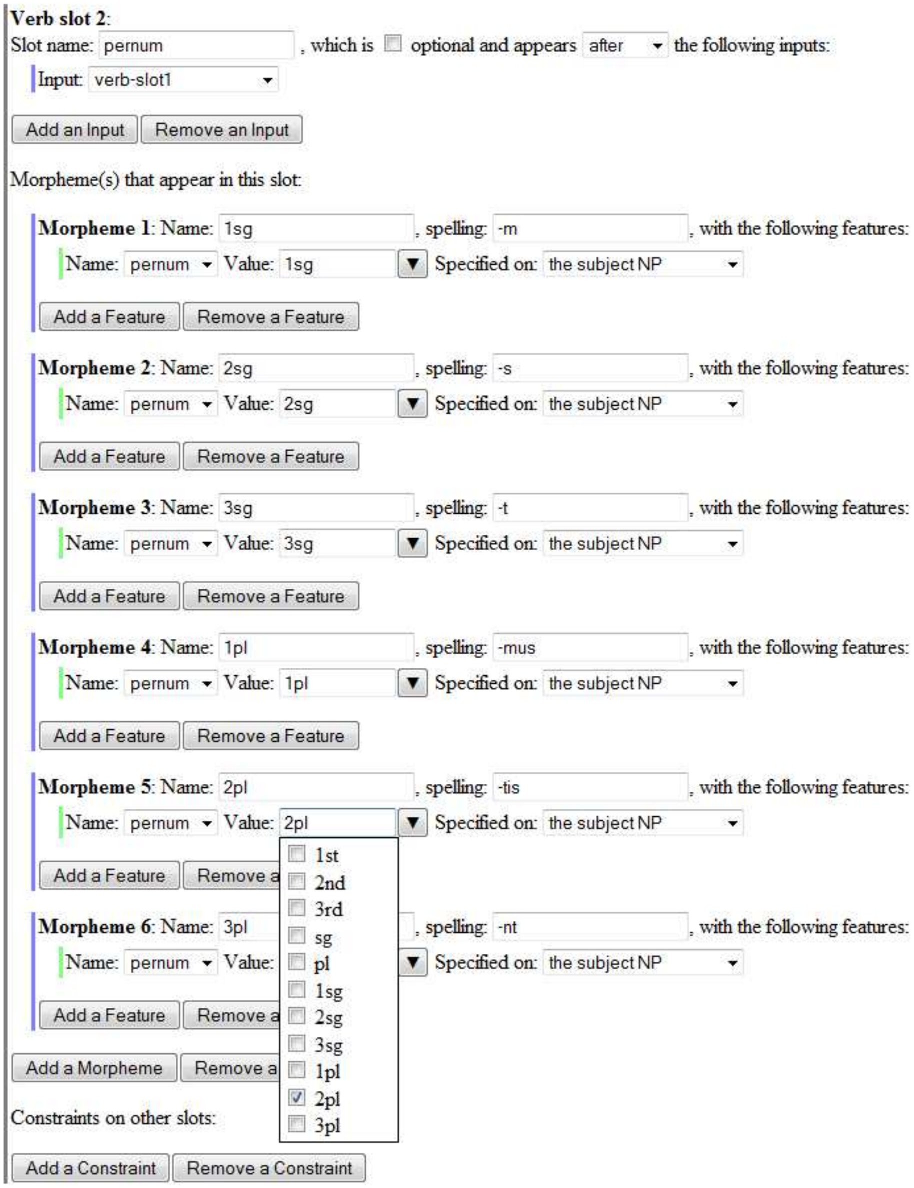
Morpheme 2: Name: imperfect , spelling: -ba , with the following features:
Name: tense Value: imperfect Specified on: the verb

Constraints on other slots:

Figure 2.3: Latin tense slot in the questionnaire

the inflected word; slot A may require that slot B *not* appear in the inflected word; and slot A may require that slot B appear *closer to the root* (i.e., lower in the phrase structure tree) in the inflected word. These additional constraints allow the modeling of even more complex inflectional patterns; see O'Hara (2008) for details.

It is important to note that the current version of the customization system supports only purely concatenative morphophonology; it focuses on the modeling of complex morphosyntatic phenomena rather than complex morphophonological phenomena. The recommended procedure for modeling languages that have more elaborate morphophonology (e.g., ablaut plurals in German, root-pattern morphology in Semitic languages) is not to model sentences in the surface orthography of the language, but rather to model morpheme-by-morpheme glosses of those sentences (Bender and Good 2005). There are well-understood solutions for modeling the mapping between these morpheme-by-morpheme glosses and surface forms (see e.g.,



Verb slot 2:
Slot name: pernum, which is □ optional and appears after the following inputs:
Input: verb-slot1

Morpheme(s) that appear in this slot:

Morpheme 1: Name: 1sg, spelling: -m, with the following features:
Name: pernum Value: 1sg Specified on: the subject NP

Morpheme 2: Name: 2sg, spelling: -s, with the following features:
Name: pernum Value: 2sg Specified on: the subject NP

Morpheme 3: Name: 3sg, spelling: -t, with the following features:
Name: pernum Value: 3sg Specified on: the subject NP

Morpheme 4: Name: 1pl, spelling: -mus, with the following features:
Name: pernum Value: 1pl Specified on: the subject NP

Morpheme 5: Name: 2pl, spelling: -tis, with the following features:
Name: pernum Value: 2pl Specified on: the subject NP

□ 1st
□ 2nd
□ 3rd
□ sg
□ pl
□ 1sg
□ 2sg
□ 3sg
□ 1pl
☑ 2pl
□ 3pl

Morpheme 6: Name: 3pl, spelling: -nt, with the following features:
Name: pernum Value: Specified on: the subject NP

Constraints on other slots:

Figure 2.4: Latin person/number slot in the questionnaire

Beesley and Karttunen (2003) for the description of a flexible system based on finite state transducers), but they are beyond the scope of this dissertation.

The addition of iterators to the customization system enables much more detailed linguistic descriptions to be elicited from the user-linguist. The ability to have iterators inside of iterators (in the above example, any number of lexical types, each containing any number of stems) allows the questionnaire to contain richly structured information. Since the system places no limit on the depth or complexity of iterator nesting, this structure is also of arbitrary complexity.

When the user-linguist is first presented with the questionnaire, it is empty. It makes no sense to attempt to create a consistent grammar from an empty questionnaire, an incomplete questionnaire, or a questionnaire containing contradictory answers, so the customization system will not accept a questionnaire that has not passed through **validation**. The validation component places a set of arbitrarily complex constraints on the answers provided. The system insists, for example, that every language have a name, that it contain at least one noun and two verb lexical entries, and that the user-linguist not state the language contains no determiners but then provide one in the Lexicon subpage. When a question fails validation, it is marked with a red asterisk ("*") in the questionnaire. If a subpage contains any red asterisks, its link on the main page is shown with an asterisk as well, as can be seen in Figure 2.1, in order to lead the user-linguist to answers that must be provided or corrected.

Table 2.4: Portion of a choices file corresponding to Figure 2.2

```
noun1_name=count              noun2_name=mass
noun1_stem1_orth=dog          noun2_stem1_orth=earth
noun1_stem1_pred=_dog_n_rel   noun2_stem1_pred=_earth_n_rel
noun1_stem2_orth=cat          noun2_stem2_orth=air
noun1_stem2_pred=_cat_n_rel   noun2_stem2_pred=_air_n_rel
noun1_stem3_orth=car          noun2_stem3_orth=water
noun1_stem3_pred=_car_n_rel   noun2_stem3_pred=_water_n_rel
noun1_det=opt                 noun2_stem4_orth=fire
                              noun2_stem4_pred=_fire_n_rel
                              noun2_det=opt
```

As the user-linguist fills out the questionnaire, the customization system keeps track of his or her answers in a special Unicode text file called the **choices file**. Table 2.4 contains an excerpt from the choices file representing the answers shown in Figure 2.2. The choices file consists of variable names and variable values, separated by the equals sign ("="). Every text field, check box, group of radio buttons, or drop-down in the questionnaire is associated with a particular unique variable name. Iterators are represented by variable names containing numbers and underscores as follows: if we have an iterator for the variable name `noun`, and each iteration can have a `name`, the choices file will contain values for variables with names like `noun1_name`, `noun2_name`, etc. Iterators can be nested, as with the iterator for stems inside of the iterator for noun types in Figure 2.2; this is reflected in choices variable names like `noun1_stem1_orth`, `noun1_stem2_orth`, `noun2_stem1_orth`, `noun2_stem2_orth`, and so forth.

The choices file is not merely a format for storing the state of the questionnaire inside the system. Filling out the questionnaire may take considerable time, especially if the user-linguist adds dozens, hundreds, or even thousands of lexical entries, so the system allows the choices file to be downloaded and stored on the user's computer, then uploaded again at some future point to continue from where he or she left off. The choices file at the time of grammar customization is also included with the grammar when the user downloads it. It is possible that a user might save a choices file from one version of the customization system, keep it long enough for a new version of the system to be developed, then upload it to the system again. The system anticipates this possibility by including a version number in each choices file and having the capability to translate every old choices file version to the current version.

When the questionnaire is complete and validated, the user-linguist can press the "Create Grammar" button on the main page. This causes the customization system to create an LKB-compatible grammar that includes all the types in the core Matrix, along with the types from each library, tailored appropriately, that are called for by the specific answers provided for the language described in the questionnaire.

2.4.5 New Libraries

This dissertation describes the new libraries that I added to the Matrix customization system, which support the following linguistic phenomena:

1. The marking of mandatory verbal arguments via case

2. Verb and verbal argument marking that is sensitive to a grammatical scale, as found in so-called direct-inverse languages

3. Agreement between verbs and their arguments in person, number, and gender, and between determiners and nouns in case, number, and gender.

I chose to implement most of these particular phenomena for a simple reason: when students attempted to use the system in the UW grammar engineering class, and when we demoed the system to linguists by asking them to describe a language they knew, we immediately ran up against the lack of case, number, person, and gender. Furthermore, while case could be implemented in relative isolation, it did not make sense to add person, number, and gender to the system without adding support for inflectional morphology and agreement. The remaining phenomenon, scale-sensitive marking in direct-inverse languages, was added because it turned up in my typological research on case and fit into the larger category of mandatory verbal argument marking, not to mention being an interesting challenge since there had been no attempt to describe such languages in the HPSG literature.

Development of new libraries involves three steps. First, the typological range of the phenomenon to be covered must be determined. Second, HPSG analyses must be developed for each of the possible expressions of each phenomenon. Finally, these analyses must be "factored" into a set of sub-analyses that the customization system can "snap together" in response to a user-linguist's answers to the questionnaire, producing a consistent grammar. This dissertation will discuss all three steps.

In addition, the correct functioning of each library will be demonstrated by the inclusion of grammars of small language fragments that contain the phenomenon covered by each library. Some of these test grammars are based on natural languages, but others model what I refer to as **pseudo-languages**, small artificial languages designed specifically to contain a particular variant of the phenomenon being tested. Use of these pseudo-languages allows the creation of test grammars that efficiently test the entire supported typological range of each phenomenon.

A frozen version of the customization system as described in this dissertation can be found at http://www.delph-in.net/matrix/sfddiss/. The most current version of the system, including any enhancements that followed this dissertation, can be found at http://www.delph-in.net/matrix/customize/.

2.5 Summary

This chapter has provided a brief introduction to HPSG, linguistic typology, multilingual grammar engineering, and the Grammar Matrix. Having presented this background material, I now turn to the description of new libraries for the Matrix.

Chapter 3

CASE[1]

Blake (2001) defines **case** as "a system of marking dependent nouns for the type of relationship they bear to their heads." Until the work described below, the Matrix customization system only had very narrow support for the marking of dependent NPs: it was possible to describe languages that marked the grammatical subject and object using case-marking adpositions (e.g., Japanese). In languages with case, however, it is often a very prominent feature, appearing in all or nearly all sentences, even very short ones. Case was therefore one of the linguistic phenomena whose lack was most often noticed by users of the customization system, and consequently one of the highest-priority phenomena to implement.

Blake's definition includes an extremely broad range of phenomena, including the marking of possessives, vocatives, locatives, adverbial adjuncts, and adpositional complements. Implementing all of these phenomena would require an enormous expansion of the customization system. In order to narrow the range of case phenomena to a dissertation-sized project, it was necessary to choose a subset of the full range of case phenomena. The Matrix case library presented here, therefore, covers only case-marking of up to two mandatory arguments of verbs.

3.1 Typology

Even within the narrowed typological range to be covered, there exists considerable variation cross-linguistically. Below, I will describe the attested patterns in case-

[1] This chapter and the following one describe in greater detail work that was originally presented in Drellishak 2008.

marking of verbal arguments.

Of course, when languages are studied in sufficient depth, we find even more complexity in the patterns of argument-marking case. For example, there are nominative-accusative languages, such as English and German, in which the nominative case marks the subject only of finite verbs. Adding support to the customization system for such fine interactions between case-marking and verb form (or between case and any other part of the grammar) is left as an area for future work, with one exception. Some languages have mandatory verbal arguments marked by additional cases beyond those marking intransitive subjects, agents, and patients, a phenomenon sometimes referred to as **quirky case** (Levin and Simpson 1981). The Matrix customization questionnaire supports the description of an arbitrary number of additional case labels, which can then be used when describing the case requirements of lexical items. In this dissertation, however, the bulk of the description and analysis will be focused on cases marking intransitive subjects, transitive agents, and transitive objects.

3.1.1 Morphosyntactic Alignment

The most important distinction among languages in their case-marking is the issue of **morphosyntactic alignment**, which concerns the pattern of marking in intransitive and transitive clauses. Following Dixon (1994), I refer to the central grammatical roles of arguments as S (intransitive subject), A (transitive agent), and O (transitive patient or object).[2]

Some languages mark S and A with the same case, and O with another case; this is called the **nominative-accusative** (or sometimes simply the **accusative**) pattern. This pattern is illustrated by the following sentences of Latin (Indo-European):

[2] Some of the literature refers to the patient role as P. I chose to use O in the customization system because, although the commonly-used abbreviations of S, O, and V are too oversimplified to use here, I felt that linguists accustomed to seeing languages categorized using abbreviations such as “SOV” or “VSO” would find “AOV” or “VAO” more transparent than “APV” or “VAP”.

(15) *domin-us veni-t*
master-NOM come-3SG
'The master comes' [lat]

(16) *serv-us veni-t*
slave-NOM come-3SG
'The slave comes' [lat]

(17) *domin-us serv-um audi-t*
master-NOM slave-ACC hear-3SG
'The master hears the slave' [lat]

(18) *serv-us domin-um audi-t*
slave-NOM master-ACC hear-3SG
'The slave hears the master' [lat] (Dixon 1994:9, glosses mine)

Other languages mark S and O the same, with A different; this is called the **ergative-absolutive** (or sometimes simply the **ergative**) pattern. The ergative pattern appears in the Australian language Dyirbal [dbl] (Pama-Nyungan):[3]

(19) *ŋuma banaga-n^{y}u*
father.ABS return-NONFUT
'Father returned' [dbl]

(20) *yabu banaga-n^{y}u*
mother.ABS return-NONFUT
'Mother returned' [dbl]

[3] The Dyirbal language as a whole, however, is not purely ergative-absolutive, showing a split conditioned on the type of nominal; see §3.1.2.2 and §3.4.2.2 for details.

(21) *ŋuma* *yabu-ŋgu* *bura-n*
father.ABS mother-ERG see-NONFUT
‘Mother saw father’ [dbl]

(22) *yabu* *ŋuma-ŋgu* *bura-n*
mother.ABS father-ERG see-NONFUT
‘Father saw mother’ [dbl] (Dixon 1994:10)

Finally, some few languages mark all three roles differently; these are called **tripartite** languages. This pattern appears throughout the nominal forms of Wangkumara [nbx] (Pama-Nyungan):

(23) *yura* *muṛuba-ŋa*
you.PL.NOM finish-PAST/P
‘You are all finished’ [nbx] (McDonald and Wurm 1979:61)

(24) *yundru* *nad̦a-gala* *d̯aldra-an̯a*
you.SG.ERG see-PRES kangaroo-ACC
‘You see the kangaroo’ [nbx] (McDonald and Wurm 1979:64)

3.1.2 Split Ergativity

Many languages are neither consistently ergative nor consistently accusative. Such languages are said to display **split ergativity**. In order to support split case-marking, the Matrix customization system must be able to create grammars in which multiple kinds of marking, commonly the ergative and accusative patterns, co-exist.

Dixon (1994:70) divides split ergative languages into four categories based on how the split is conditioned:

1. Semantic nature of the main verb

2. Semantic nature of the core NPs[4]

3. Tense, aspect, or mood of the clause

4. Grammatical status of the clause

These different types of split are described in turn below.

3.1.2.1 Nature of Main Verb

The first type of split occurs in two subtypes. In the first, called **split-S**, the intransitive verbs are divided into two classes: those that take A-like marking on their single arguments and those that take O-like marking. This pattern is found in Mandan [mhq] (Siouan). The following are examples are from Mixco (1997), and involve verbs that can be marked for the case and person of both agents and patients. Mixco uses the abbreviation "A" to refer to "active" marking, which appears on the agents of transitives, and "S" to refer to "stative" marking, which marks patients of transitives.

(25) *rı̨-∅-hœ-oʔš*
S2-A3-see-INDma
'He sees you' [mhq]

(26) *wı̨-wa:-xwœ:-oʔš*
S1.SG-UNSP-hide-INDma
'I'm hiding' [mhq]

(27) *wa:-ptœ:h-oʔš*
A1.SG-run-INDma
'I'm running' [mhq] (Mixco 1997:16–17)

[4] Dixon uses the term "semantic" here, but he includes, for example, the difference between common nouns and pronouns as a kind of semantic distinction. In HPSG, such a distinction would likely be modeled using different lexical types or different features—a primarily syntactic distinction rather than a semantic one, though there would likely be semantic differences as well.

The other subtype is called **fluid-S**. Fluid-S languages have, in addition to the two classes of verbs described above for split-S languages, an additional intransitive verb class in which the single argument can be marked like A or like O, depending on whether the subject controls the action or not: when a speaker marks an intransitive subject like A, this emphasizes the agency of the subject; when the subject is marked like O, this implies a lack of volition on the part of the subject.

A language that displays the fluid-S pattern is Bats [bbl] (North Caucasian) (also known as Tsova-Tush). Holisky (1987) performed an experiment to check 303 verbs with native speakers to see whether A-like marking, S-like marking, or both were acceptable. She found 31 verbs that were acceptable only with O-like marking and 78 verbs that were acceptable only with A-like marking; the remaining 194 were acceptable with either (Holisky 1987, cited in Dixon 1994:79–80).

3.1.2.2 Nature of NP*s*

The second type of ergativity split is conditioned on the nature of the nominal arguments. In such languages, certain kinds of NPs (e.g., pronouns) are marked in a nominative-accusative pattern while others (e.g., common nouns) are marked in an ergative-absolutive pattern. Dyirbal, mentioned above as an example of a language with the ergative-absolutive pattern, actually has a split between different noun classes. First and second person pronouns take nominative-accusative marking: a zero-marked nominative and a marked accusative. Third person pronouns, proper names, and common nouns take ergative-absolutive marking: a zero-marked absolutive and a marked ergative (Dixon 1994:86). See §3.4.2.2 for a detailed description of this phenomenon in Dyirbal.

3.1.2.3 Clausal Splits

The third and fourth types of split are similar, both types being conditioned on clausal features. The third type is conditioned on the tense, aspect, or mood of the clause.

In many Iranian languages, for example, clauses in the past tense are marked in an ergative-absolutive pattern, while clauses in other tenses take nominative-accusative marking (Dixon 1994:100). Gujarati [guj] (Indo-Iranian) also has the third type of split, with the nominative-accusative pattern in the imperfective aspect and the ergative-absolutive pattern in the perfective:

(28) *Ramesh* *pen* *khərid-t-o* *hə-t-o*
Ramesh.MASC pen.F buy-IMPF-MASC AUX-IMPF-MASC
'Ramesh was buying the pen.' [guj]

(29) *Ramesh-e* *pen* *khərid-y-i*
Ramesh.MASC-ERG pen.F buy-PERF-F
'Ramesh bought the pen.' [guj] (Croft 1990:139)

The fourth type of split is conditioned on the grammatical status of the clause; for example, whether it is a main or subordinate clause. An example of a language like this is Päri [lkr] (Nilo-Saharan), in which "S is generally treated like O but in purposive clauses..., S is instead treated like A." (Dixon 1994:103)

Analyses of each of these four types of ergativity split will be described in §3.2 below.

3.1.3 Focus-case Systems

Some Austronesian languages display an interesting variant of verbal argument marking (Comrie 1989:120). In Tagalog [tgl] (Austronesian, Philippines), a language of this type, noun phrase arguments must be marked by one of several case-marking prepositions, one of which marks an NP as the **focus** (Comrie 1989:121). The focus is marked by *ang*, while agent and patient are marked by *ng*. Every clause must have one argument marked as the focus. In intransitive clauses, this will be the sole argument. In transitive clauses, the verb is marked by one of a set of affixes that tell how the focus-marked NP should be interpreted, including among others agent-focus and patient-focus affixes. This pattern can be seen in the following examples:

(30) *Bumili* *ang* *babae* *ng* *baro*
bought-AGENT.FOCUS FOCUS woman PATIENT dress
'The woman bought a dress' [tgl]

(31) *Bimili* *ng* *babae* *ang* *baro*
bought-PATIENT.FOCUS AGENT woman FOCUS dress
'A/the woman bought the dress' [tgl] (Comrie 1989:121)[5]

A more detailed description of this phenomenon in Tagalog can be found in §3.4.2.4 below.

3.1.4 Summary

In this section, I have described the typology of argument-marking case, including morphosyntactic alignment and ergativity splits of various kinds. This typology will form the basis of my analysis of case and of the design of the case section of the questionnaire.

3.2 Analysis

In this section, I provide an HPSG analysis of argument-marking case, including marking strategies, types for lexical items, and case hierarchies for each type of marking. The features I use in my analysis are generally compatible with standard approaches to case in HPSG. Pollard and Sag (1994) described a version of HPSG in which CASE was a feature of nominal HEADs, and the case of verbal arguments, both subjects

[5] Comrie actually uses the terms *actor* and *undergoer*, but I use *agent* and *patient* here for consistency. Note that, although a single case-marker *ng* is used to mark both agents and patients in Tagalog, my analysis distinguishes between agent and patient, allowing it to model languages where they are marked differently.

and complements, was specified in the lexical entries for verbs. This can be seen in their lexical entry (Pollard and Sag 1994:29) for the English verb *sees*, which selects a nominative, third person singular subject and an accusative object:

(32) *sees*
$$\begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{HEAD} & \textit{verb}[\textit{fin}] \\ \text{SUBCAT} & \left\langle \text{NP}[\textit{nom}]_{\boxed{1}[3rd,sing]},\ \text{NP}[\textit{acc}]_{\boxed{2}} \right\rangle \end{bmatrix} \\ \text{CONTENT} & \begin{bmatrix} \text{RELN} & \textit{see} \\ \text{SEER} & \boxed{1} \\ \text{SEEN} & \boxed{2} \end{bmatrix} \end{bmatrix}$$

Apart from changes in feature geometry—for example, the feature SUBCAT has been replaced by ARG-ST in the version of HPSG used in the Matrix—and minor differences in the formatting of feature structures, this closely resembles the analysis I will describe below, which also uses a CASE feature on HEAD and verb lexical types that specify the case of their arguments. However, my analysis differs in one important respect: as described in the next section, I also use a CASE feature on the HEAD of adpositional phrases, since the case library supports case-marking adpositions. My analysis also makes use of more articulated hierarchies for the value of the CASE feature, where Pollard and Sag use only the values *nom* and *acc*.

3.2.1 Marking Strategies

The analysis of case in the Grammar Matrix case library provides, in the lexicon section of the questionnaire, several strategies for the marking of case on the NP arguments: marking of whole NPs via case-marking adpositions, or marking morphologically on nouns, determiners, or both.

Morphological marking is accomplished using lexical rules. These non-branching rules take a lexical item as their input, apply some spelling change, constrain the values of one or more features (in these rules, the CASE feature), and copy up all other information to form a new sign.[6]

For languages that only have morphological marking, the analysis of case is simple. Nominal heads have a CASE feature whose value is specified either via lexical rules corresponding to case inflection or directly on a lexical type (e.g., for case-marked pronouns). Verbs take arguments that are specified to be noun phrases, and the case on those noun phrases is specified in one of the patterns described below.

For languages that have adpositional marking of case, the analysis is slightly different, but still fairly simple. The CASE feature appears on both nominal and adpositional heads. Case-marking adpositions are lexical items that take nominal complements and specify a particular value of CASE on both that complement and on the resulting adpositional phrase. Verbs specify adpositional phrases rather than noun phrases as their arguments; this prevents bare noun phrases from appearing as arguments.

3.2.1.1 Mixed Marking

A more complex treatment is required for languages with mixed case marking—that is, languages with both morphological and adpositional case-marking. Tagalog is an example of such a language, with adpositional marking of common nouns but not of pronouns (Schachter and Otanes 1972:88). As in pure-adpositional languages, the CASE feature appears on both NPs and adpositional phrases. Depending on how the various cases are marked in the target language, verbs specify their arguments to be of head type *noun*, *adp*, or *+np* (a type defined in the Matrix that is a supertype of *noun* and *adp*).

However, these constraints alone are not sufficient to model a language with mixed

[6] Lexical rules are given in the format of Sag et al. (2003) (with INPUT and OUTPUT features) for conciseness—in the format used in grammars compatible with the LKB, lexical rules are actually described in two different files, one giving the type definition and the other the lexical rule instances, including the spelling change specifications.

case-marking. Consider a language with adpositional marking of the nominative and morphological marking of the accusative. The morphological accusative is modeled via a mandatory lexical rule applying to nouns—in the questionnaire, a slot in which a single morpheme can appear. The adpositional nominative should be formed by an adposition taking a noun as its complement and marking the resulting phrase as nominative; however, since the lexical rule marking the accusative is mandatory, there is no way to get a bare, non-case-marked NP to use as the adposition's complement.

In grammars produced by the customization system, the solution to this is to synthesize an additional, non-spelling-changing lexical rule that applies to nouns and marks the appropriate value of case, one rule for each case-marking adposition in order to ensure that every adpositionally-marked case is covered. In the present example, the system will "sneak in" an additional null morpheme marking the nominative case into the case-marking slot on nouns.

A further complication can arise in languages with mixed marking: double marking of case. Suppose the target language has two ways of marking the nominative case, either via an adposition or via inflection. Both a noun phrase produced by the inflectional rule and an adpositional phrase with the nominative adposition as its head are marked [CASE *nom*], and can thus serve as the subject, as expected. However, the case-marking adposition could also take a noun that is already morphologically case-marked as its complement, producing erroneous redundant marking.

The solution to this is the addition of a new feature to nominal heads in grammars with mixed marking: CASE-MARKED, which takes the values + and −. It is unconstrained on lexical items. Any case-marking morpheme specified by the user also marks the noun [CASE-MARKED +], but the non-spelling-changing morpheme that is "snuck in" leaves CASE-MARKING unspecified. Case-marking adpositions, on the other hand, require NP complements that are compatible with [CASE-MARKED −]. This allows unmarked NPs as the complement of a case-marking adposition, but excludes any NPs that are already case-marked. Finally, verbs specifying case on their arguments also require those arguments to be [CASE-MARKED +], ensuring that bare NPs marked by the "snuck-in" rules are excluded.

3.2.1.2 Case-marked Determiners

Before the work I describe in this dissertation, the customization system had support for determiners in the lexicon. My development of libraries for case and agreement suggested the addition of support for another phenomenon: case-marking on determiners. A language exhibiting this phenomenon is German [deu] (Germanic). Notice that the German determiners in the following sentences are marked for case, but the nouns *Mann* and *Mädchen* are not. Notice further than the determiners agree with the nouns they attach to in gender:

(33) *Das Mädchen sieht den Mann*
the.N.NOM girl.N.SG see.3SG the.M.ACC man.M.SG
'The girl sees the man.' [deu]

(34) *Der Mann hilft dem Mädchen*
the.M.NOM man.M.SG help.3SG the.N.DAT girl.N.SG
'The man helps the girl.' [deu]

The valence features in the Matrix make this marking pattern easy to analyze. The VAL feature of signs contains, in addition to the SUBJ and COMPS lists, two similar features called SPR and SPEC. The SPR list is optionally specified for nouns, and requires the appearance of a specifier attaching to the noun phrase that must, much like the subject and complement requirements on verbs, be "canceled off" in order to construct a grammatical sentence. The specifier, often of head type *det* (determiner), is usually attached by a head-specifier rule. The SPEC list is the mirror image of the SPR list: it is specified on the determiner, and its first element is identified by the head-specifier rule with whatever noun the determiner attaches to. By specifying features on the SPEC list, therefore, determiners can constrain the features of nouns they attach to, including their CASE.

To analyze the German determiners above, the customization system uses lexical types such as the following:

(35)
$$\begin{bmatrix} \textit{neut-nom-determiner-lex} & \\ \text{VAL} \mid \text{SPEC} & \left\langle \begin{bmatrix} \text{HEAD} \mid \text{CASE} & \textit{nom} \\ \text{INDEX} \mid \text{GEND} & \textit{neut} \end{bmatrix} \right\rangle \end{bmatrix}$$

$$\begin{bmatrix} \textit{masc-acc-determiner-lex} & \\ \text{VAL} \mid \text{SPEC} & \left\langle \begin{bmatrix} \text{HEAD} \mid \text{CASE} & \textit{acc} \\ \text{INDEX} \mid \text{GEND} & \textit{masc} \end{bmatrix} \right\rangle \end{bmatrix}$$

$$\begin{bmatrix} \textit{masc-nom-determiner-lex} & \\ \text{VAL} \mid \text{SPEC} & \left\langle \begin{bmatrix} \text{HEAD} \mid \text{CASE} & \textit{nom} \\ \text{INDEX} \mid \text{GEND} & \textit{masc} \end{bmatrix} \right\rangle \end{bmatrix}$$

$$\begin{bmatrix} \textit{neut-dat-determiner-lex} & \\ \text{VAL} \mid \text{SPEC} & \left\langle \begin{bmatrix} \text{HEAD} \mid \text{CASE} & \textit{dat} \\ \text{INDEX} \mid \text{GEND} & \textit{neut} \end{bmatrix} \right\rangle \end{bmatrix}$$

From these lexical types, the lexical items for the German determiners can be derived: *das* from *neut-nom-determiner-lex*, *den* from *masc-acc-determiner-lex*, *der* from *masc-nom-determiner-lex*, and *dem* from *neut-dat-determiner-lex*. For an example of this marking pattern in action, see §3.4.2.1 below.

3.2.1.3 Summary

In this section, I have described the various marking strategies that are available for case in the customization system. These include morphological marking of nouns and determiners and the marking of whole noun phrases via case-marking adpositions.

3.2.2 Simple Case-Marking

Nominative-accusative, ergative-absolutive, and tripartite NP case-marking can be specified on verb lexical types as shown below using the ARG-ST feature, based on the ARG-S feature of Manning and Sag (1995), to constrain the argument structure, with the lexical types also providing the mapping between ARG-ST and the SUBJ and COMPS lists. The current version of the system always treats S or A as the subject and O as an object by placing them on the SUBJ and COMPS lists, respectively, but in fact, this is not an adequate analysis cross-linguistically. Some languages show inter-clausal or **syntactic ergativity**, in which S and O pattern together in constructions including coordination and relative clauses (Dixon 1979:127). Manning (1996) describes an analysis of the variation between morphological and syntactic ergativity; however, the current version of the Matrix questionnaire includes almost no multi-clausal phenomena (the exception being clausal coordination), so support for syntactic ergativity has been left for future work.

(36) Nominative-Accusative

$$\begin{bmatrix} \textit{intransitive-verb-lex} & \\ \text{VAL} \mid \text{SUBJ} & \left\langle \boxed{1} \right\rangle \\ \text{ARG-ST} & \left\langle \boxed{1} \begin{bmatrix} \text{HEAD} \mid \text{CASE} & \textit{nom} \end{bmatrix} \right\rangle \end{bmatrix}$$

$$\begin{bmatrix} \textit{transitive-verb-lex} & \\ \text{VAL} & \begin{bmatrix} \text{SUBJ} & \left\langle \boxed{1} \right\rangle \\ \text{COMPS} & \left\langle \boxed{2} \right\rangle \end{bmatrix} \\ \text{ARG-ST} & \left\langle \boxed{1}\begin{bmatrix}\text{HEAD}\,|\,\text{CASE} & \textit{nom}\end{bmatrix}, \boxed{2}\begin{bmatrix}\text{HEAD}\,|\,\text{CASE} & \textit{acc}\end{bmatrix} \right\rangle \end{bmatrix}$$

(37) Ergative-Absolutive

$$\begin{bmatrix} \textit{intransitive-verb-lex} & \\ \text{VAL}\,|\,\text{SUBJ} & \left\langle \boxed{1} \right\rangle \\ \text{ARG-ST} & \left\langle \boxed{1}\begin{bmatrix}\text{HEAD}\,|\,\text{CASE} & \textit{abs}\end{bmatrix} \right\rangle \end{bmatrix}$$

$$\begin{bmatrix} \textit{transitive-verb-lex} & \\ \text{VAL} & \begin{bmatrix} \text{SUBJ} & \left\langle \boxed{1} \right\rangle \\ \text{COMPS} & \left\langle \boxed{2} \right\rangle \end{bmatrix} \\ \text{ARG-ST} & \left\langle \boxed{1}\begin{bmatrix}\text{HEAD}\,|\,\text{CASE} & \textit{erg}\end{bmatrix}, \boxed{2}\begin{bmatrix}\text{HEAD}\,|\,\text{CASE} & \textit{abs}\end{bmatrix} \right\rangle \end{bmatrix}$$

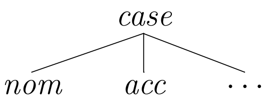
(38) Tripartite

$$\begin{bmatrix} \textit{intransitive-verb-lex} & \\ \text{VAL}\,|\,\text{SUBJ} & \left\langle \boxed{1} \right\rangle \\ \text{ARG-ST} & \left\langle \boxed{1}\begin{bmatrix}\text{HEAD}\,|\,\text{CASE} & \textit{s}\end{bmatrix} \right\rangle \end{bmatrix}$$

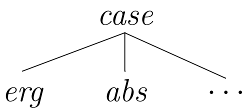
$$\begin{bmatrix} \textit{transitive-verb-lex} & \\ \text{VAL} & \begin{bmatrix} \text{SUBJ} & \left\langle \boxed{1} \right\rangle \\ \text{COMPS} & \left\langle \boxed{2} \right\rangle \end{bmatrix} \\ \text{ARG-ST} & \left\langle \boxed{1}\begin{bmatrix}\text{HEAD}\,|\,\text{CASE} & \textit{a}\end{bmatrix}, \boxed{2}\begin{bmatrix}\text{HEAD}\,|\,\text{CASE} & \textit{o}\end{bmatrix} \right\rangle \end{bmatrix}$$

In all grammars produced by the customization system, the feature CASE on HEAD has a value of type *case*, but the shape of the hierarchy under *case* varies depending on the marking pattern. The case hierarchies for nominative-accusative, ergative-absolutive, and tripartite need only to distinguish a set of exclusive alternatives (e.g., nominative is not compatible with accusative), so the corresponding hierarchies are flat:[7]

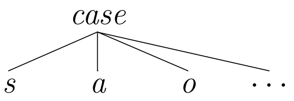
(39) Nominative-Accusative Case



(40) Ergative-Absolutive Case



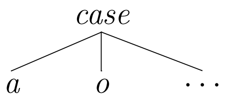
(41) Tripartite Case



3.2.3 Split-S

I analyze split-S languages as having the following simple case hierarchy:

(42)



Based on this case type, split-S grammars have a single transitive verb class with A- and O-marked arguments, but two intransitive verb classes:

[7] In representations of case hierarchies, the location where any additional quirky cases defined by the user will appear is represented by $\cdots$.

(43)
$$\begin{bmatrix} \textit{a-intrans-verb-lex} & \\ \textsc{arg-st} & \left\langle \begin{bmatrix} \textsc{head}\,|\,\textsc{case} & a \end{bmatrix} \right\rangle \end{bmatrix}$$

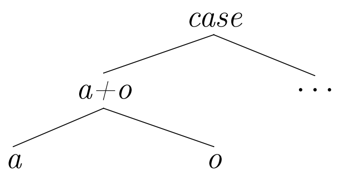
$$\begin{bmatrix} \textit{o-intrans-verb-lex} & \\ \textsc{arg-st} & \left\langle \begin{bmatrix} \textsc{head}\,|\,\textsc{case} & o \end{bmatrix} \right\rangle \end{bmatrix}$$

The questionnaire allows the user-linguist to define verb lexical entries by defining any number of verb classes, each of which contains any number of stems. For each user-defined verb class, the user-linguist can choose which of the three lexical types above it derives from.

3.2.4 Fluid-S

Fluid-S languages have a class of verbs that mark intransitive subjects as more or less agent-like. The semantic representation in grammars produced by the customization system does not presently have any way to show such a distinction; however, it is possible to model the three intransitive verb classes. I analyze fluid-S languages with a slightly more articulated case hierarchy:

(44)



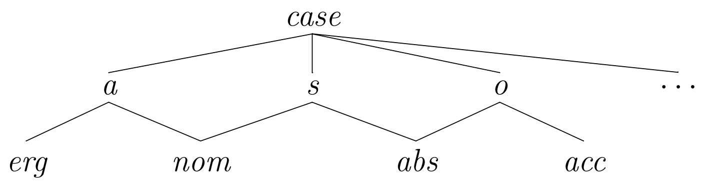
Fluid-S grammars include, in addition to the two lexical types above in (43), a lexical type for the fluid-marking verb class. This type simply specifies that the case of intransitive subjects is a supertype of both A and O, which means that NPs marked with either A or O will be compatible with, and therefore allowed as, the subjects of intransitives.

(45) $$\begin{bmatrix} \textit{a+o-intrans-verb-lex} & \\ \textsc{arg-st} & \left\langle \begin{bmatrix} \textsc{head} \mid \textsc{case} & \textit{a+o} \end{bmatrix} \right\rangle \end{bmatrix}$$

3.2.5 Split-N

I analyze split-N languages with a rather more articulated case hierarchy:

(46)

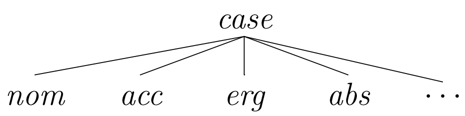


For this type of language, the customization system will produce the same verb lexical types, shown in (38), that it would for a tripartite language. That is, an intransitive verb's sole argument is specified as [CASE *s*], and a transitive verb's agent and patient arguments as [CASE *a*] and [CASE *o*], respectively. Then, when creating noun classes in the lexicon section of the questionnaire, the user-linguist will be prompted to specify for each class whether it is marked for *nom* (which unifies with *s* and *a*) and *acc* (which unifies just with *o*), or for *erg* (which unifies just with *a*) and *abs* (which unifies with *s* and *o*). This analysis puts the complexity in the right place in the lexicon for languages where the split is conditioned on the noun: verbs are not split, instead deriving from either the single intransitive or the single transitive type, while nouns are divided into classes based on whether they take the nominative-accusative or the ergative-absolutive pattern.

3.2.6 Split-V

Splits conditioned on tense, aspect, mood, or the grammatical status of the clause all receive a similar analysis. The case hierarchy is flat, and has at least four values:

(47)

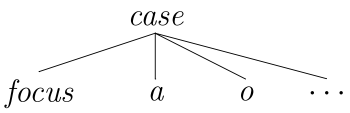


Verb lexical items have no case specified on their arguments; instead, a set of mandatory lexical rules is used to constrain the CASE values on their ARG-ST lists. For tense/aspect/mood splits, the lexical rule that marks the conditioning feature (e.g., the past-tense morpheme) will constrain the CASE value of the arguments. For clausal splits, two non-spelling-changing lexical rules can be used, along with the Matrix's MC (main clause) feature, to achieve the proper analysis: one rule marks the clause as [MC +] and constrains the cases on ARG-ST to one pattern, while the second rule marks the clause as [MC −] and constrains the cases on ARG-ST to the other pattern. However, the customization system has no support for any phenomena involving a subordinate clause, so there is no way to describe languages with a split based on clausal type using the questionnaire. Furthermore, at the time the case library was implemented, the Matrix customization system available at http://www.delph-in.net/matrix/sfddiss/ had only limited support for tense, aspect, and mood, but see Poulson (forthcoming) for the details of a library for tense and aspect.

3.2.7 Focus-case

This pattern of argument marking is neither accusative nor ergative, instead constituting a distinct pattern. I analyze it as follows. The case hierarchy is:

(48)



NPs are marked for agent, patient, or focus case, either directly in the lexicon, via lexical rules, or via case-marking adpositions. The sole argument on the ARG-ST of the intransitive verb lexical type is specified to have focus case. The lexical type of transitive verbs has an ARG-ST that is unspecified for case. If a class of verbs is focus-marked, then in the lexicon section of the questionnaire, the user must describe a set of morphemes marking the appropriate type of focus (including agent and patient focus), which will produce in the output grammar a set of lexical rules that both apply the appropriate spelling change and constrain the cases of the arguments on ARG-ST. The rules for agent- and patient-focus marking are:

(49)
$$\begin{bmatrix} \textit{agent-focus-verb-lex-rule} & \\ \text{INPUT} & \left\langle \boxed{1}, \textit{transitive-verb-lex} \right\rangle \\ \text{OUTPUT} & \left\langle \begin{array}{l} F_{af}(\boxed{1}), \\ \begin{bmatrix} \text{ARG-ST} & \left\langle \begin{bmatrix}\text{HEAD}\,|\,\text{CASE} & \textit{focus}\end{bmatrix}, \begin{bmatrix}\text{HEAD}\,|\,\text{CASE} & \textit{o}\end{bmatrix} \right\rangle \end{bmatrix} \end{array} \right\rangle \end{bmatrix}$$

$$\begin{bmatrix} \textit{patient-focus-verb-lex-rule} & \\ \text{INPUT} & \left\langle \boxed{1}, \textit{transitive-verb-lex} \right\rangle \\ \text{OUTPUT} & \left\langle \begin{array}{l} F_{pf}(\boxed{1}), \\ \begin{bmatrix} \text{ARG-ST} & \left\langle \begin{bmatrix}\text{HEAD}\,|\,\text{CASE} & \textit{a}\end{bmatrix}, \begin{bmatrix}\text{HEAD}\,|\,\text{CASE} & \textit{focus}\end{bmatrix} \right\rangle \end{bmatrix} \end{array} \right\rangle \end{bmatrix}$$

3.2.8 Summary

In this section, I have described an analysis for each of the various types of argument-marking case. This analysis and the division between the types inform the structure and wording of the section of the questionnaire related to case.

3.3 Questionnaire

The section of the customization system questionnaire devoted to case has a structure based closely on the typology described in §3.1. The user is first presented with the following introductory text:

> Blake (2001) defines **case** as "a system of marking dependent nouns for the type of relationship they bear to their heads." For example, case commonly marks the subject or the direct object of a verb. The marking might consist of affixation or some other morphological process that marks words, or it might be a adposition that marks whole noun phrases.
>
> This questionnaire allows you to describe **core case marking** in your language; that is, the pattern of cases marking the mandatory arguments of transitive and intransitive verbs. Following Dixon (1968), we refer to the grammatical relations commonly expressed by case using the following abbreviations: A refers to the agent of a transitive verb; O refers to the patient (or object) of a transitive verb; and S refers to the lone argument (or subject) of intransitive verbs.
>
> What type of core case marking does your language exhibit?

After this, the user-linguist is asked to select one case pattern out of a set of options. These options (along with additional explanatory text, if present) include:

> **None**
>
> **Nominative-accusative**
>
> **Ergative-absolutive**
>
> **Tripartite**
>
> **Split-S**
>
> > "The S argument of some intransitive verbs is marked by the same case as the agent of transitives, while for other verbs the S argument is marked by the same case as the patient."
>
> **Fluid-S**
>
> > "The S argument of some intransitive verbs is marked by the same case as the agent of transitives, while for other verbs the S argument

is marked by the same case as the patient, and for still other verbs the S argument can be marked by either case, depending on pragmatic factors (e.g. whether the S is perceived as being in control of the action."

Split conditioned on features of the noun phrase arguments

"Some classes of noun phrases (e.g. pronouns) show a nominative-accusative pattern, while others (e.g. common nouns) show an ergative-absolutive pattern. You will have an opportunity to define these classes on the Lexicon page."

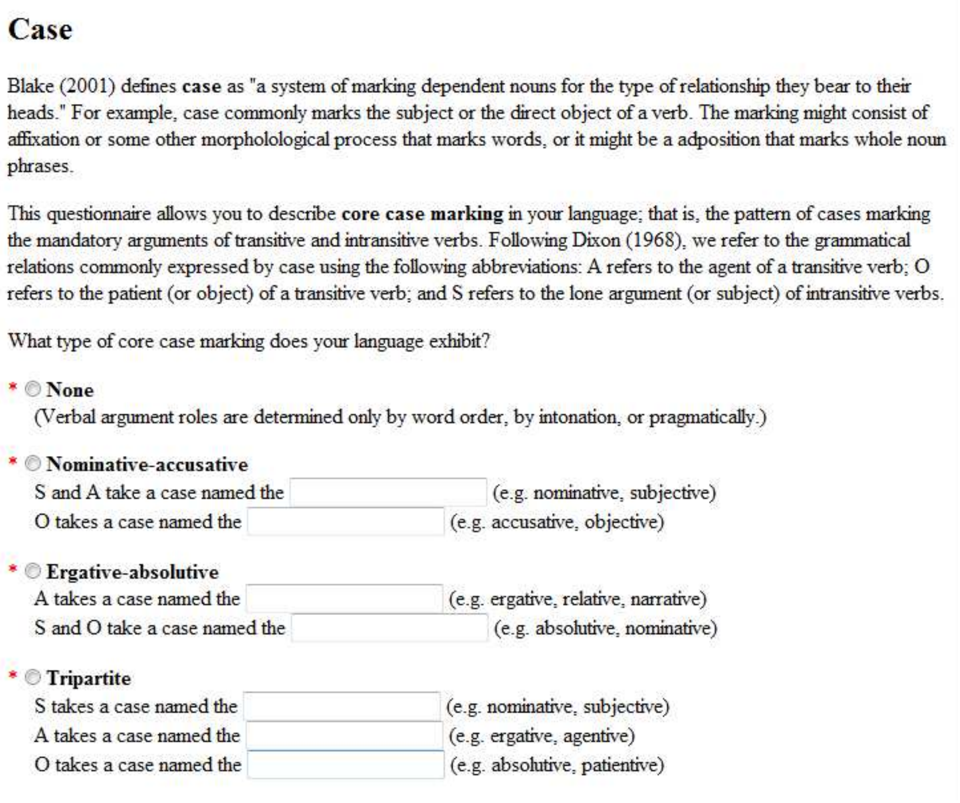
Split conditioned on features of the verb

"Depending on some feature of the verb (e.g. tense or aspect), the core arguments are sometimes marked in a nominative-accusative pattern and other times in an ergative-absolutive pattern. You will have an opportunity to define these features on the Lexicon page."

Focus-case

"A number of Austronesian languages, including several Philippine languages, have a system where A and O are marked by cases. An additional case, sometimes called the focus, is mandatory in every clause and has its grammatical role assigned by the morphology of the verb. You will have an opportunity to define this morphology on the Lexicon page."

For each of these options, the user-linguist is then asked to supply labels for each case defined by the chosen case pattern. For each case, suggestions are supplied for common names of the grammatical role associated with the case; these suggestions come largely from Haspelmath (2009). The beginning of the case section is shown in Figure 3.1.



Case

Blake (2001) defines **case** as "a system of marking dependent nouns for the type of relationship they bear to their heads." For example, case commonly marks the subject or the direct object of a verb. The marking might consist of affixation or some other morpholological process that marks words, or it might be a adposition that marks whole noun phrases.

This questionnaire allows you to describe **core case marking** in your language; that is, the pattern of cases marking the mandatory arguments of transitive and intransitive verbs. Following Dixon (1968), we refer to the grammatical relations commonly expressed by case using the following abbreviations: A refers to the agent of a transitive verb; O refers to the patient (or object) of a transitive verb; and S refers to the lone argument (or subject) of intransitive verbs.

What type of core case marking does your language exhibit?

* ○ **None**
(Verbal argument roles are determined only by word order, by intonation, or pragmatically.)

* ○ **Nominative-accusative**
S and A take a case named the (e.g. nominative, subjective)
O takes a case named the (e.g. accusative, objective)

* ○ **Ergative-absolutive**
A takes a case named the (e.g. ergative, relative, narrative)
S and O take a case named the (e.g. absolutive, nominative)

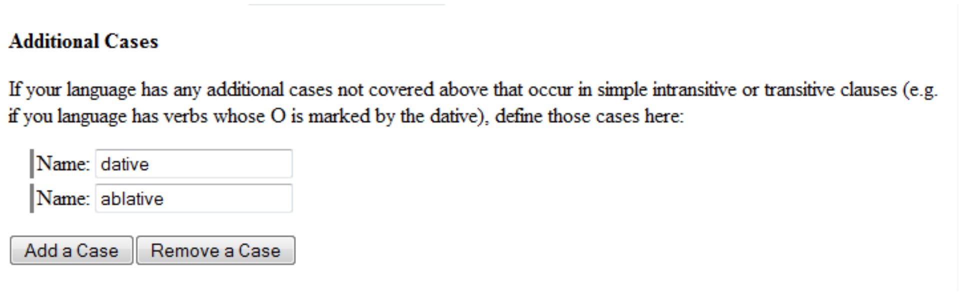
* ○ **Tripartite**
S takes a case named the (e.g. nominative, subjective)
A takes a case named the (e.g. ergative, agentive)
O takes a case named the (e.g. absolutive, patientive)

Figure 3.1: The beginning of the case section of the questionnaire

At the end of the case section, the user-linguist is optionally allowed to define any number of additional cases to be used for quirky-case verbs. This is accomplished with a simple iterated text field containing the name of each additional case. This section of the case section, filled out to represent a language with a dative and an ablative case, is shown in Figure 3.2.

3.4 Test Cases

In order to verify that the grammars produced by the case library are behaving as expected, I constructed a number of test cases. Each test case consists of two parts: first, a choices file describing a language produced by filling out the questionnaire and second, a set of test sentences. These sentences are fed into the grammar produced



Additional Cases

If your language has any additional cases not covered above that occur in simple intransitive or transitive clauses (e.g. if you language has verbs whose O is marked by the dative), define those cases here:

Name: dative
Name: ablative

Figure 3.2: The section of the questionnaire for defining quirky cases

by the customization system based on the choices file, and the resulting grammaticality judgments and semantic representations are verified. There are two kinds of test cases. The first set are very small pseudo-languages, based on no particular natural language, that have restricted vocabularies and are intended to test only a particular, narrow part of the Matrix's implementation of case-marking. The second set consists of four test cases based on fragments natural languages: German [deu] (Germanic), Dyirbal [dbl] (Pama-Nyungan), Hindi [hin] (Indo-Iranian), and Tagalog [tgl] (Austronesian). These two sets of tests serve complementary purposes: the first is intended to thoroughly, but shallowly, test the function of all supported variations of case-marking, while the second set is intended to demonstrate that the system is capable of handling the more complex case-marking patterns that appear in real natural languages.

3.4.1 Pseudo-Languages

The test cases that are not based on a natural language all have similar lexicons. Each includes two nouns spelled *n1* and *n2*. In pseudo-languages with inflectional case-marking, cases are marked by suffixes whose spellings indicates the cases they mark, including -*NOM*, -*ACC*, -*ERG*, -*ABS*, -*S*, -*A*, -*O*, and -*FOC*. Each lexicon also includes at least two verbs, usually an intransitive verb *iv* and a transitive verb *tv*. Some test cases have additional lexical items or inflection; these will be described below. The behavior of each test case is illustrated by a set of sentences, some grammatical and some ungrammatical, that are accepted and rejected by each grammar.

The first nine test cases represent pseudo-languages for each of the possible choices for case-marking in the questionnaire: none, nominative-accusative, ergative-absolutive, tripartite, split-S, fluid-S, split-N, split-V, or focus-case. All of these grammars are, arbitrarily, SVO. The behavior of these nine test cases is illustrated in Tables 3.1 through 3.9. Two of these pseudo-languages require a bit of additional explanation. The split-V test case in Table 3.8 has an additional feature, marked by the suffix -*PAST*, that is used to distinguish past-tense clauses, which take ergative-absolutive marking, from non-past-tense clauses, which take nominative-accusative marking. The focus-case pseudo-language in Table 3.9 has three suffixes on the verb, -*SFOC*, -*AFOC*, and -*PFOC*, that cause the noun in focus case to be interpreted as the S, A, or O argument, respectively.

Table 3.1: Test sentences: None

Grammatical	**Ungrammatical**
n1 iv	*iv n1
n1 tv n2	*n1 n2 tv
	*tv n1 n2

Table 3.2: Test sentences: Nominative-accusative

Grammatical	**Ungrammatical**	
n1-NOM iv	*n1 iv	*n1 tv n2-NOM
n1-NOM tv n2-ACC	*n1-ACC iv	*n1-NOM tv n2-NOM
	*n1 tv n2	*n1-ACC tv n2-NOM
	*n1-NOM tv n2	*n1 tv n2-ACC
	*n1-ACC tv n2	*n1-ACC tv n2-ACC

The remaining four test cases in Tables 3.10 through 3.13 illustrate case-marking adpositions. All are nominative-accusative, and all have SVO word order. Table 3.10

Table 3.3: Test sentences: Ergative-absolutive

Grammatical	Ungrammatical	
n1-ABS iv	*n1 iv	*n1 tv n2-ABS
n1-ERG tv n2-ABS	*n1-ERG iv	*n1-ABS tv n2-ABS
	*n1 tv n2	*n1 tv n2-ERG
	*n1-ABS tv n2	*n1-ABS tv n2-ERG
	*n1-ERG tv n2	*n1-ERG tv n2-ERG

Table 3.4: Test sentences: Tripartite

Grammatical	Ungrammatical	
n1-S iv	*n1 iv	*n1-A tv n2-S
n1-A tv n2-O	*n1-A iv	*n1-O tv n2-S
	*n1-O iv	*n1 tv n2-A
	*n1 tv n2	*n1-S tv n2-A
	*n1-S tv n2	*n1-A tv n2-A
	*n1-A tv n2	*n1-O tv n2-A
	*n1-O tv n2	*n1 tv n2-O
	*n1 tv n2-S	*n1-S tv n2-O
	*n1-S tv n2-S	*n1-O tv n2-O

Table 3.5: Test sentences: Split-S

Grammatical	Ungrammatical	
n1-A iv-a	*n1 iv-a	*n1-O tv n2
n1-O iv-o	*n1-O iv-a	*n1 tv n2-A
n1-A tv n2-O	*n1 iv-o	*n1-A tv n2-A
	*n1-A iv-o	*n1-O tv n2-A
	*n1 tv n2	*n1 tv n2-O
	*n1-A tv n2	*n1-O tv n2-O

corresponds to a pseudo-language with case-marking postpositions. Table 3.11 corresponds to a pseudo-language with case marked both by adpositions (a nominative preposition) and morphologically (an accusative suffix). Table 3.12 corresponds to a pseudo-language with purely prepositional case-marking, but in which those prepositions are optional. Finally, Table 3.13 corresponds to a pseudo-language with optional

Table 3.6: Test sentences: Fluid-S

Grammatical	**Ungrammatical**	
n1-O iv-o	*n1 iv-o	*n1-O tv n2
n1-A iv-a	*n1-A iv-o	*n1 tv n2-A
n1-O iv	*n1 iv-a	*n1-A tv n2-A
n1-A iv	*n1-O iv-a	*n1-O tv n2-A
n1-A tv n2-O	*n1 tv n2	*n1 tv n2-O
	*n1-A tv n2	*n1-O tv n2-O

prepositional case-marking of nominatives and mandatory morphological marking of accusatives.

Table 3.7: Test sentences: Split-N

Grammatical	Ungrammatical	
n1-NOM iv	*n1 iv	*n1 tv n2-ABS
n2-ABS iv	*n1-ACC iv	*n1-ACC tv n2-ABS
n1-NOM tv n1-ACC	*n1-ERG iv	*n1-ERG tv n2-ABS
n1-NOM tv n2-ABS	*n1-ABS iv	*n1-ABS tv n2-ABS
n2-ERG tv n1-ACC	*n2 iv	*n2 tv n1
n2-ERG tv n2-ABS	*n2-NOM iv	*n2-NOM tv n1
	*n2-ACC iv	*n2-ACC tv n1
	*n2-ERG iv	*n2-ERG tv n1
	*n1 tv n2	*n2-ABS tv n1
	*n1-NOM tv n2	*n2 tv n1-NOM
	*n1-ACC tv n2	*n2-NOM tv n1-NOM
	*n1-ERG tv n2	*n2-ACC tv n1-NOM
	*n1-ABS tv n2	*n2-ERG tv n1-NOM
	*n1 tv n2-NOM	*n2-ABS tv n1-NOM
	*n1-NOM tv n2-NOM	*n2 tv n1-ACC
	*n1-ACC tv n2-NOM	*n2-NOM tv n1-ACC
	*n1-ERG tv n2-NOM	*n2-ACC tv n1-ACC
	*n1-ABS tv n2-NOM	*n2-ABS tv n1-ACC
	*n1 tv n2-ACC	*n2 tv n1-ERG
	*n1-NOM tv n2-ACC	*n2-NOM tv n1-ERG
	*n1-ACC tv n2-ACC	*n2-ACC tv n1-ERG
	*n1-ERG tv n2-ACC	*n2-ERG tv n1-ERG
	*n1-ABS tv n2-ACC	*n2-ABS tv n1-ERG
	*n1 tv n2-ERG	*n2 tv n1-ABS
	*n1-NOM tv n2-ERG	*n2-NOM tv n1-ABS
	*n1-ACC tv n2-ERG	*n2-ACC tv n1-ABS
	*n1-ERG tv n2-ERG	*n2-ERG tv n1-ABS
	*n1-ABS tv n2-ERG	*n2-ABS tv n1-ABS

Table 3.8: Test sentences: Split-V

Grammatical	**Ungrammatical**	
n1-NOM iv	*n1 iv	*n1 tv n2-ACC
n1-ABS iv-PAST	*n1-ACC iv	*n1-ACC tv n2-ACC
n1-NOM tv n2-ACC	*n1-ERG iv	*n1-ERG tv n2-ACC
n1-ERG tv-PAST n2-ABS	*n1-ABS iv	*n1-ABS tv n2-ACC
	*n1 iv-PAST	*n1 tv-PAST n2-ACC
	*n1-NOM iv-PAST	*n1-NOM tv-PAST n2-ACC
	*n1-ACC iv-PAST	*n1-ACC tv-PAST n2-ACC
	*n1-ERG iv-PAST	*n1-ERG tv-PAST n2-ACC
	*n1 tv n2	*n1-ABS tv-PAST n2-ACC
	*n1-NOM tv n2	*n1 tv n2-ERG
	*n1-ACC tv n2	*n1-NOM tv n2-ERG
	*n1-ERG tv n2	*n1-ACC tv n2-ERG
	*n1-ABS tv n2	*n1-ERG tv n2-ERG
	*n1 tv-PAST n2	*n1-ABS tv n2-ERG
	*n1-NOM tv-PAST n2	*n1 tv-PAST n2-ERG
	*n1-ACC tv-PAST n2	*n1-NOM tv-PAST n2-ERG
	*n1-ERG tv-PAST n2	*n1-ACC tv-PAST n2-ERG
	*n1-ABS tv-PAST n2	*n1-ERG tv-PAST n2-ERG
	*n1 tv n2-NOM	*n1-ABS tv-PAST n2-ERG
	*n1-NOM tv n2-NOM	*n1 tv n2-ABS
	*n1-ACC tv n2-NOM	*n1-NOM tv n2-ABS
	*n1-ERG tv n2-NOM	*n1-ACC tv n2-ABS
	*n1-ABS tv n2-NOM	*n1-ERG tv n2-ABS
	*n1 tv-PAST n2-NOM	*n1-ABS tv n2-ABS
	*n1-NOM tv-PAST n2-NOM	*n1 tv-PAST n2-ABS
	*n1-ACC tv-PAST n2-NOM	*n1-NOM tv-PAST n2-ABS
	*n1-ERG tv-PAST n2-NOM	*n1-ACC tv-PAST n2-ABS
	*n1-ABS tv-PAST n2-NOM	*n1-ABS tv-PAST n2-ABS

Table 3.9: Test sentences: Focus-case

Grammatical	Ungrammatical	
n1-FOC iv-SFOC	*n1 iv	*n1 tv-PFOC n2-A
n1-FOC tv-AFOC n2-O	*n1-A iv	*n1-A tv-PFOC n2-A
n1-A tv-PFOC n2-FOC	*n1-O iv	*n1-O tv-PFOC n2-A
	*n1-FOC iv	*n1-FOC tv-PFOC n2-A
	*n1 iv-SFOC	*n1 tv n2-O
	*n1-A iv-SFOC	*n1-A tv n2-O
	*n1-O iv-SFOC	*n1-O tv n2-O
	*n1 tv n2	*n1-FOC tv n2-O
	*n1-A tv n2	*n1 tv-AFOC n2-O
	*n1-O tv n2	*n1-A tv-AFOC n2-O
	*n1-FOC tv n2	*n1-O tv-AFOC n2-O
	*n1 tv-AFOC n2	*n1 tv-PFOC n2-O
	*n1-A tv-AFOC n2	*n1-A tv-PFOC n2-O
	*n1-O tv-AFOC n2	*n1-O tv-PFOC n2-O
	*n1-FOC tv-AFOC n2	*n1-FOC tv-PFOC n2-O
	*n1 tv-PFOC n2	*n1 tv n2-FOC
	*n1-A tv-PFOC n2	*n1-A tv n2-FOC
	*n1-O tv-PFOC n2	*n1-O tv n2-FOC
	*n1-FOC tv-PFOC n2	*n1-FOC tv n2-FOC
	*n1 tv n2-A	*n1 tv-AFOC n2-FOC
	*n1-A tv n2-A	*n1-A tv-AFOC n2-FOC
	*n1-O tv n2-A	*n1-O tv-AFOC n2-FOC
	*n1-FOC tv n2-A	*n1-FOC tv-AFOC n2-FOC
	*n1 tv-AFOC n2-A	*n1 tv-PFOC n2-FOC
	*n1-A tv-AFOC n2-A	*n1-O tv-PFOC n2-FOC
	*n1-O tv-AFOC n2-A	*n1-FOC tv-PFOC n2-FOC
	*n1-FOC tv-AFOC n2-A	

Table 3.10: Test sentences: Adpositions

Grammatical	Ungrammatical	
n1 NOM iv	*n1 iv	*n1 tv n2 NOM
n1 NOM tv n2 ACC	*n1 ACC iv	*n1 NOM tv n2 NOM
	*NOM n1 iv	*n1 ACC tv n2 NOM
	*n1 tv n2	*n1 tv n2 ACC
	*n1 NOM tv n2	*n1 ACC tv n2 ACC
	*n1 ACC tv n2	

Table 3.11: Test sentences: Mixed adpositions and inflection

Grammatical	**Ungrammatical**	
NOM n1 iv	*n1 iv	*n1 tv n2-ACC
NOM n1 tv n2-ACC	*n1-ACC iv	*NOM n1 tv n2
	*NOM n1-ACC iv	*NOM n1 tv NOM n2
	*NOM NOM n1 iv	*n1-ACC tv n2
	*n1 tv n2	*n1-ACC tv NOM n2
	*n1 tv NOM n2	*n1-ACC tv n2-ACC

Table 3.12: Test sentences: Optional adposition

Grammatical	**Ungrammatical**
n1 iv	*ACC n1 iv
NOM n1 iv	*NOM ACC n1 iv
n1 tv n2	*NOM NOM n1 iv
n1 tv ACC n2	*n1 tv NOM n2
NOM n1 tv n2	*NOM n1 tv NOM n2
NOM n1 tv ACC n2	*ACC n1 tv n2
	*ACC n1 tv NOM n2
	*ACC n1 tv ACC n2

Table 3.13: Test sentences: Mixed optional adpositions and inflection

Grammatical	**Ungrammatical**	
n1 iv	*n1-ACC iv	*NOM n1 tv n2
NOM n1 iv	*NOM n1-ACC iv	*NOM n1 tv NOM n2
n1 tv n2-ACC	*NOM NOM n1 iv	*n1-ACC tv n2
NOM n1 tv n2-ACC	*n1 tv n2	*n1-ACC tv NOM n2
	*n1 tv NOM n2	*n1-ACC tv n2-ACC

3.4.2 Natural Languages

Each of the remaining four test cases in this chapter covers a limited fragment of a natural language. Both the languages and the fragments have been selected to test the function of the case library. The test sentences for each grammar are all grammatical sentences of the target language; however, since each of the grammars lacks many linguistic phenomena, each covers far less than all the grammatical strings in the language it models.

In addition, three of the grammars, namely German, Dyirbal, and Hindi, differ from the natural languages they are based on in an important respect: I have filled out the questionnaire for a more fixed word order than is actually found in those languages. I did this for two reasons. First, freer word orders inevitably mean that the set of grammatical sentences will be larger, but not in a way that tests the case library—it merely makes the lists of sentences that must be presented here longer and more unwieldy. Second, freer word orders make it harder to construct ungrammatical examples, which are just as important as grammatical sentences in probing the behavior of a test grammar.

3.4.2.1 German

German was chosen as a test language because, although most German NPs are marked for case, most nouns are not declined for case; rather, most case distinctions are visible only on determiners. German also has quirky case verbs. Making a grammar for German tests the support for both of these phenomena in the customization system. I have based my German grammar on the description of German by Donato et al. (2004).

In the questionnaire, I described a fragment German as follows: it is SVO (but note that the full language is actually V2), it is nominative-accusative, and additionally has a dative case; it has three genders: masculine, feminine, and neuter; and it has determiners that appear before the noun, and those determiners are declined for case. The test grammar's lexicon includes a masculine noun (*Mann* 'man'), a feminine noun (*Frau* 'woman'), and a neuter noun (*Mädchen* 'girl'), none of which are declined for case. It also includes what is known as a weak masculine noun (*Mensch* 'person'), which has a distinct form for the accusative and dative, and an inflectional morpheme that marks this form. It includes three verbs, all in the third person singular present tense form: *schläft* 'sleep', which takes a single nominative argument; *sieht* 'see', which takes a nominative subject and an accusative object, and *hilft* 'help', which takes a nominative subject and a dative object. I also described a set of variants of the definite article, each of which constrains the case and gender of the noun it attaches to; these are shown in Table 3.14

Table 3.14: German determiners

	masculine	**feminine**	**neuter**
nominative	*der*	*die*	*das*
accusative	*den*	*die*	*das*
dative	*dem*	*der*	*dem*

The precise details of the fragment of German can be found in the choices file, which is in Appendix A. The set of test sentences includes all possible grammatical sentences, intransitive and transitive, as well as a number of ungrammatical variants. These test sentences, along with the grammaticality judgments assigned by the test grammar, are shown in Table 3.15.

3.4.2.2 Dyirbal

Dyirbal is an ergative-absolutive language through most of its grammar, but the first and second person pronouns follow the nominative-accusative pattern. It therefore falls into the category referred to here as split-N, and serves as a test of that part of

Table 3.15: Test sentences: German

Grammatical	Ungrammatical	
der Mann schläft	**(missing determiner)**	**(wrong det on dat obj)**
der Mensch schläft	*Mann schläft	*die Frau hilft der Mann
die Frau schläft	*Mensch schläft	*die Frau hilft die Mann
das Mädchen schläft	*Frau schläft	*die Frau hilft das Mann
der Mann sieht den Mann	*Mädchen schläft	*die Frau hilft den Mann
der Mann sieht den Menschen		*die Frau hilft der Menschen
der Mann sieht die Frau	**(wrong det on subj)**	*die Frau hilft die Menschen
der Mann sieht das Mädchen	*die Mann schläft	*die Frau hilft das Menschen
der Mensch sieht den Mann	*das Mann schläft	*die Frau hilft den Menschen
der Mensch sieht den Menschen	*die Mensch schläft	*die Frau hilft die Frau
der Mensch sieht die Frau	*das Mensch schläft	*die Frau hilft das Frau
der Mensch sieht das Mädchen	*der Frau schläft	*die Frau hilft den Frau
die Frau sieht den Mann	*das Frau schläft	*die Frau hilft dem Frau
die Frau sieht den Menschen	*der Mädchen schläft	*die Frau hilft der Mädchen
die Frau sieht die Frau	*die Mädchen schläft	*die Frau hilft die Mädchen
die Frau sieht das Mädchen		*die Frau hilft das Mädchen
das Mädchen sieht den Mann	**(wrong weak masculine)**	*die Frau hilft den Mädchen
das Mädchen sieht den Menschen	*der Mann sieht den Mensch	
das Mädchen sieht die Frau	*der Menschen sieht die Frau	
das Mädchen sieht das Mädchen		
der Mann hilft dem Mann	**(wrong det on obj)**	
der Mann hilft dem Menschen	*der Mann sieht der Mann	
der Mann hilft der Frau	*der Mann sieht die Mann	
der Mann hilft dem Mädchen	*der Mann sieht das Mann	
der Mensch hilft dem Mann	*der Mann sieht dem Mann	
der Mensch hilft dem Menschen	*der Mann sieht der Menschen	
der Mensch hilft der Frau	*der Mann sieht die Menschen	
der Mensch hilft dem Mädchen	*der Mann sieht das Menschen	
die Frau hilft dem Mann	*der Mann sieht dem Menschen	
die Frau hilft dem Menschen	*der Mann sieht der Frau	
die Frau hilft der Frau	*der Mann sieht das Frau	
die Frau hilft dem Mädchen	*der Mann sieht den Frau	
das Mädchen hilft dem Mann	*der Mann sieht dem Frau	
das Mädchen hilft dem Menschen	*der Mann sieht der Mädchen	
das Mädchen hilft der Frau	*der Mann sieht die Mädchen	
das Mädchen hilft dem Mädchen	*der Mann sieht den Mädchen	
	*der Mann sieht dem Mädchen	

the case library. I have based this grammar on the description of Dyirbal by Dixon (1972).

Dyirbal has very free word order; furthermore, even the preferred order of constituents is beyond what can be described in the customization system. Dixon (1972:291) states that agent pronouns tend to precede other NPs, that "nominative" (i.e. absolutive) NPs tend to precede ergative NPs, and that ergative NPs tend to precede the verb. Notice that agent pronouns and ergative NPs do not pattern together—this word order cannot be described in the customization system questionnaire. Furthermore, the system provides no way to describe word order tendencies, only strict patterns. Instead, I have described Dyirbal in the questionnaire as an SOV language and created test sentences accordingly; these sentences are still grammatical, of course, because of Dyirbal's free word order.

I further described a fragment of Dyirbal in the questionnaire as follows: it has mandatory determiners (which Dixon calls "noun markers") that precede the noun; it distinguishes first, second, and third person; it has two genders, I and II (in fact, the full language also has III and IV); and its case-marking is split-N, with the four cases labeled *nom*, *acc*, *erg*, and *abs*.[8] In the lexicon, I defined two nouns, two determiners, and two pronouns, each of which has two case forms. These are shown in Table 3.16.

Table 3.16: Dyirbal nominals and determiners

Nouns	*abs*	*erg*
'man', Class I	*yaɽa*	*yaɽaŋgu*
'woman', Class II	*ɖugumbil*	*ɖugumbiɽu*
Determiners	*abs*	*erg*
Class I	*bayi*	*baŋgul*
Class II	*balan*	*baŋgun*
Pronouns	*nom*	*acc*
'I/me'	*ŋaɖa*	*ŋayguna*
'you'	*ŋinda*	*ŋinuna*

I also included two verbs in the lexicon, both in the unmarked (non-past) tense: intransitive *baniɲu* 'is coming' and transitive *balgan* 'is hitting'. This lexicon can be used to construct simple sentences like:

[8] Note that Dixon refers to both the form of pronouns marking S and A and the form of nouns marking S and O as "nominative", whereas I describe the latter as "absolutive" in the questionnaire.

(50) *bayi yaɽa baniɲu*
I.ABS man.I.ABS come.NPST
‘man is coming’ [dbl]

(51) *balan ɖugumbil baniɲu*
II.ABS woman.II.ABS come.NPST
‘woman is coming’ [dbl]

(52) *balan ɖugumbil baŋgul yaɽaŋgu balgan*
II.ABS woman.II.ABS I.ERG man.I.ERG hit.NPST
‘man is hitting woman’ [dbl]

(53) *bayi yaɽa baŋgun ɖugumbiɽu balgan*
I.ABS man.I.ABS II.ERG woman.II.ERG hit.NPST
‘woman is hitting man’ [dbl] (Dixon 1972:59, glosses mine)

The precise details of the fragment of Dyirbal can be found in the choices file, which is in Appendix B. The set of test sentences includes all possible grammatical sentences with the exception of those with the same agent and patient, since these would call for a reflexive form, as well as a number of ungrammatical variants. These test sentences, along with the grammaticality judgments assigned by the test grammar, are shown in Table 3.17.

3.4.2.3 Hindi

Hindi shows an ergativity split of a different kind, namely a split between the marking of NP arguments conditioned on the aspect of the verb. However, there are two significant respects in which Hindi is not straightforwardly a split-V language. First, there is a case missing from the pattern. Transitives in the perfective aspect take subjects and objects in the ergative and accusative, respectively; transitives in non-perfective aspects take the nominative and accusative. There are therefore only three

Table 3.17: Test sentences: Dyirbal

Grammatical	**Ungrammatical**
bayi yaɽa baniɲu	**(wrong case on S)**
balan ɖugumbil baniɲu	*baŋgul yaɽaŋgu baniɲu
ŋaɖa baniɲu	*baŋgun ɖugumbiɽu baniɲu
ŋinda baniɲu	*ŋayguna baniɲu
baŋgul yaɽaŋgu balan ɖugumbil balgan	*ŋinuna baniɲu
baŋgul yaɽaŋgu ŋayguna balgan	
baŋgul yaɽaŋgu ŋinuna balgan	**(wrong case on A)**
baŋgun ɖugumbiɽu bayi yaɽa balgan	*bayi yaɽa balan ɖugumbil balgan
baŋgun ɖugumbiɽu ŋayguna balgan	*bayi yaɽa ŋayguna balgan
baŋgun ɖugumbiɽu ŋayguna balgan	*bayi yaɽa ŋinuna balgan
ŋaɖa bayi yaɽa balgan	*balan ɖugumbil bayi yaɽa balgan
ŋaɖa balan ɖugumbil balgan	*balan ɖugumbil ŋayguna balgan
ŋaɖa ŋinuna balgan	*balan ɖugumbil ŋayguna balgan
ŋinda bayi yaɽa balgan	*ŋayguna bayi yaɽa balgan
ŋinda balan ɖugumbil balgan	*ŋayguna balan ɖugumbil balgan
ŋinda balan ɖugumbil balgan	*ŋayguna ŋinuna balgan
	*ŋinuna bayi yaɽa balgan
	*ŋinuna balan ɖugumbil balgan
	*ŋinuna balan ɖugumbil balgan
	(wrong case on O)
	*baŋgul yaɽaŋgu baŋgun ɖugumbiɽu balgan
	*baŋgul yaɽaŋgu ŋaɖa balgan
	*baŋgul yaɽaŋgu ŋinda balgan
	*baŋgun ɖugumbiɽu baŋgul yaɽaŋgu balgan
	*baŋgun ɖugumbiɽu ŋaɖa balgan
	*baŋgun ɖugumbiɽu ŋaɖa balgan
	*ŋaɖa baŋgul yaɽaŋgu balgan
	*ŋaɖa baŋgun ɖugumbiɽu balgan
	*ŋaɖa ŋinda balgan
	*ŋinda baŋgul yaɽaŋgu balgan
	*ŋinda baŋgun ɖugumbiɽu balgan
	*ŋinda baŋgun ɖugumbiɽu balgan

of the possible four cases in play, with the absolutive missing. Second, the behavior of verbs in the intransitive is not what we might expect of a split-V language. Instead, Hindi intransitives display the fluid-S pattern: some take the nominative, some the ergative, and some either. Nonetheless, it is possible to describe this pattern of argument marking in the questionnaire and get full coverage of a set of representative test sentences. I have based this grammar on the description of Hindi by Mohanan (1994).

Hindi has free word order; however, there is one word order that Mohanan (1994:11) refers to as "canonical", namely SOV. I have therefore described it as such in the questionnaire. I have further described a fragment of Hindi as follows: it has split-V case-marking, with four cases labeled *nom*, *acc*, *erg*, and *abs* (with *abs* going unused); and it has a two-way distinction between future and perfective aspect (since the imperfective involves an auxiliary verb, I have omitted it for simplicity).

In the lexicon, I defined two nouns, both proper names, *raam* 'Ram' and *ravii* 'Ravi', along with inflection for three cases: *-ne* for the ergative, *-ko* for the accusative, and a null morpheme for the nominative. I also defined four verbs. Three were intransitive: *gir* 'fall', which specifies a nominative subject; *c^hĩĩk* 'sneeze', which specifies an ergative subject; and *naac* 'dance', which can take either a nominative or ergative subject. The sole transitive verb was *piiṭ* 'beat', and it specified that its object is accusative. Aspect was marked on the verb by suffixes: *-aa* for the perfective and *-egaa* for the future. In fact, I defined both of these affixes twice. One variant attached to intransitives and did not specify any case on the arguments. The other attached to transitives and specified the appropriate case on the subject: *erg* for perfective *-aa*, *nom* for future *-egaa*.

Using these lexical items and inflection, it is possible to construct sentences like:

(54) *raam-ne ravii-ko piiṭ-aa*
Ram-ERG Ravi-ACC beat-PERF
'Ram beat Ravi.' [hin]

(55) *raam ravii-ko piiṭ-egaa*
Ram.NOM Ravi-ACC beat-FUT
'Ram will beat Ravi.' [hin] (Mohanan 1994:70)

The precise details of the fragment of Hindi can be found in the choices file, which is in Appendix C. The set of test sentences includes all grammatical patterns, though sentences with the same agent and patient have again been avoided since there is no way to describe a reflexive. The test set also includes a variety of ungrammatical variants. These test sentences, along with the grammaticality judgments assigned by the test grammar, are shown in Table 3.18.

Table 3.18: Test sentences: Hindi

Grammatical	**Ungrammatical**
raam giraa	**(intrans, wrong case)**
raam giregaa	*raamne giraa
raamne c^{h}ĩĩkaa	*raamko giraa
raamne c^{h}ĩĩkegaa	*raamne giregaa
raam naacaa	*raamko giregaa
raamne naacaa	*raam c^{h}ĩĩkaa
raam naacegaa	*raamko c^{h}ĩĩkaa
raamne naacegaa	*raam c^{h}ĩĩkegaa
raamne raviiko piiṭaa	*raamko c^{h}ĩĩkegaa
raam raviiko piiṭegaa	*raamko naacaa
	*raamko naacegaa
	(trans, wrong case)
	*raam ravii piiṭaa
	*raam raviine piiṭaa
	*raam raviiko piiṭaa
	*raamko ravii piiṭaa
	*raamko raviine piiṭaa
	*raamko raviiko piiṭaa
	*raamne ravii piiṭegaa
	*raamne raviine piiṭegaa
	*raamne raviiko piiṭegaa
	*raamko ravii piiṭegaa
	*raamko raviine piiṭegaa
	*raamko raviiko piiṭegaa

3.4.2.4 Tagalog

As described above in §3.1.3, the Tagalog language marks verbal arguments according to the focus-case pattern. I have based this grammar on descriptions of Tagalog by Schachter and Otanes (1972) and Kroeger (1993).

Tagalog marks cases using prepositions: *ang* marks the focus NP argument, while agents and patients are marked by *ng*.[9] The interpretation of the focus case depends on the inflection of the verb, with possible inflections including subject-focus, agent-focus, and patient-focus. The word order of sentences is strictly verb-initial, and I have described it as such in the questionnaire. I have further described a fragment of Tagalog as a focus-case language with three cases labeled *foc*, *a*, and *o*.

The verbal morphology of Tagalog is quite complex; in particular, there are many inflectional classes of verbs, and the same feature can be marked by a prefix, a suffix, a circumfix, or an infix, depending on the inflectional class. To avoid this complexity (which is beyond the capability of the LKB), I have carefully chosen verbs that are inflected solely by prefixes; therefore, the lexical items below and the test sentences created using them are grammatical.

In the lexicon, I defined two nouns: *babae* 'woman' and *lalaki* 'man'. I defined one intransitive verb, *tulog* 'sleep', which can take a single prefix, *ma-*, marking the sole argument as the focus. I defined one transitive verb, *kita* 'see', which can be inflected either with *ma-* for agent-focus or with *maka-* for patient-focus. Finally, I defined the two case-marking prepositions described above: *ang* for focus case and *ng* for the non-focus cases.

Using these lexical items and inflection, it is possible to construct sentences like:

[9] There are other prepositions for other roles (e.g., recipients and beneficiaries), but the discussion here is restricted to agents and patients.

(56) *ma-tulog ang babae*
SFOC-sleep FOC woman
'The woman sleeps.' [tgl]

(57) *ma-kita ang babae ng lalaki*
AFOC-see FOCUS woman PATIENT man
'The woman sees the man.' [tgl]

(58) *maka-kita ng babae ang lalaki*
PFOC-see AGENT woman *focus* man
'The woman sees the man.' [tgl]

The precise details of the fragment of Tagalog can be found in the choices file, which is in Appendix D. The set of test sentences includes all possible grammatical sentences, though sentences with the same agent and patient have been avoided, as well as a number of ungrammatical variants. These test sentences, along with the grammaticality judgments assigned by the test grammar, are shown in Table 3.19.

3.5 Summary

In this chapter I have described the implementation of a library for core case-marking in the Grammar Matrix customization system. This implementation involved choosing a typology of case-marking, developing an HPSG analysis of each of its variants, and designing a questionnaire that allows a language's case-marking to be described. Furthermore, I have demonstrated that the grammars produced by the customization system have the intended coverage by creating test cases that probe the system's behavior for all the supported case-marking types.

Table 3.19: Test sentences: Tagalog

Grammatical	**Ungrammatical**
matulog ang babae	**(wrong prefix)**
matulog ang lalaki	*tulog ang babae
makita ang babae ng lalaki	*tulog ng babae
makakita ng babae ang lalaki	*makatulog ang babae
makita ng babae ang lalaki	*makatulog ng babae
makakita ang babae ng lalaki	*kita ang babae ng lalaki
	*kita ng babae ng lalaki
	*kita ang babae ang lalaki
	(wrong case)
	*matulog babae
	*matulog ng babae
	*makita babae lalaki
	*makita babae ang lalaki
	*makita babae ng lalaki
	*makita ang babae lalaki
	*makita ang babae ang lalaki
	*makita ng babae lalaki
	*makita ng babae ng lalaki
	*makakita babae lalaki
	*makakita babae ang lalaki
	*makakita babae ng lalaki
	*makakita ang babae lalaki
	*makakita ang babae ang lalaki
	*makakita ng babae lalaki
	*makakita ng babae ng lalaki

Chapter 4

DIRECT-INVERSE LANGUAGES[1]

In languages with **direct-inverse** marking, the marking of verbal arguments is sensitive to a grammatical hierarchy or scale. In each sentence, if the agent is ranked more highly than the patient, the clause is said to be **direct**; if the patient is higher, the clause is said to be **inverse**. Different languages have different direct-inverse hierarchies, and the marking of direct and inverse clauses varies as well.

For an example of direct-inverse marking, let us consider the Algonquian languages, where the argument-marking scale is primarily sensitive to person:

(59) 2nd > 1st > 3rd proximate > 3rd obviative

The entity in (59) is often referred to in the literature as a *hierarchy*, but it differs markedly from the sort of multiply-inheriting type hierarchies found in HPSG. The hierarchy in (59) only implies one-dimensional precedence relationships among the positions on the hierarchy; in contrast, HPSG-style type hierarchies involve arbitrary pairwise inheritance relationships among the types they contain. To avoid confusion, I will hereafter consistently refer to grammatical hierarchies like (59) as **scales**.[2]

4.1 Typology

The direct-inverse pattern occurs in a number of languages and language families. In his survey of the phenomenon, Givón (1994) divides these into two categories:

[1] This chapter and the preceding one describe in greater detail work that was originally presented in Drellishak 2008.

[2] The usage of *hierarchy* to refer to such scales, it should be noted, has quite a long history in linguistics, and includes such well-known examples as the Noun Phrase Accessibility Hierarchy of Keenan and Comrie (1977).

pronominal or morphological inverses, which will be the focus of this chapter, and word order inverses, which involve an interaction with word order that is beyond the current capability of the customization system. Among the pronominal inverses, Givón includes the Algonquian languages (1994:16), Koyukon [koy] (Na-Dene) (1994:17), Sahaptin [sah] (Penutian) (1994:20), Squamish [squ] (Salishan) (1994:21), Kimbundu [kmb] and Dzamba [bni] (Niger-Congo) (1994:26–28), and Maasai [mas] (Nilo-Saharan) (1994:29).

In this section, I describe in detail how the direct-inverse argument marking pattern operates in the Algonquian language family of North America and in Fore, a language of Papua New Guinea that I will argue can also be analyzed as direct-inverse.

4.1.1 Algonquian

As mentioned above, all the Algonquian languages have argument marking that is primarily sensitive to person. In addition, when a transitive clause in an Algonquian language contains two non-coreferential third-person arguments, one of them will be marked as proximate and the other as obviative to prevent ambiguity. The Algonquian proximate NP, according to Dahlstrom (1991:91), is usually "the topic of the discourse" or "the focus of the speaker's empathy". The proximate NP is generally unmarked, while the obviative noun is marked by a suffix.

Recall the Algonquian person scale mentioned above in (59) above. The following examples from Fox [sac] (Algonquian) illustrate how this scale controls argument marking:

(60) *ne -waapam-aa -wa*
1SG see-DIRECT 3
'I see him.' [sac]

(61) *ne -waapam-ek -wa*
1SG see-INVERSE 3
'He sees me.' [sac] (Comrie 1989:129)

When the agent is first person and the patient is third person, the agent outranks the patient on the scale, so the verb is marked direct. When the agent is third person and the patient is first, on the other hand, the patient outranks the agent, and so the verb is marked inverse.

4.1.2 Fore

Grammatical scales can also control the verbal argument marking patterns in languages that lack direct or inverse marking on the verb. One such language is Fore [for] (Trans-New Guinea), where the relative position of agent and patient on a scale correlates with the presence or absence of a marker on the agent NP. The scale governing argument marking in Fore is:

(62) pronoun, name, kin term > human > animate > inanimate

The operation of this hierarchy can be seen in the following examples (Scott 1978:116, Blake 2001:122):

(63) *yaga: wá aegúye*
pig man 3SG.hit.3SG
'The man kills the pig' [for]

(64) *yaga:-wama wá aegúye*
pig-DLN man 3SG.hit.3SG
'The pig kills the man' [for]

(65) *wa* *yága:-wama* *aegúye*
man pig-DLN 3SG.hit.3SG
'The pig kills the man' [for]

An extra suffix *-wama* (which Scott (1978) describes as a "delineator") appears on the agent when it is lower on the hierarchy than the patient. Scott describes these facts of Fore without referring to it as a direct-inverse language; however, I will show that this marking pattern can be analyzed by treating Fore as direct-inverse language where, instead of marking on the verb, it is the marking of case on NPs that is sensitive to direct or inverse clauses.

4.2 Analysis

Analyzing the direct-inverse pattern is challenging in the version of HPSG used in the Matrix (which, recall, is expressed in TDL and interpreted by the LKB system). For transitive verbs, it is necessary to constrain the verb's arguments differently for direct and inverse clauses. It would be convenient when modeling this aspect of direct-inverse languages (via lexical rules, say) if there were a formal mechanism for stating scale constraints compactly, perhaps something like:

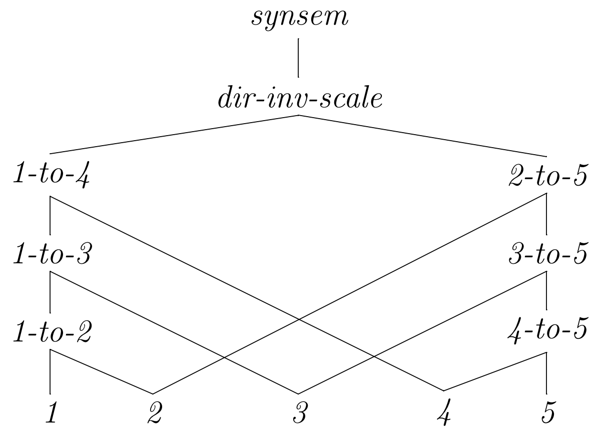
(66)
$$\begin{bmatrix} \textit{direct-verb-lex-rule} & \\ \text{INPUT} & \left\langle \boxed{1}, \ldots \right\rangle \\ \text{OUTPUT} & \left\langle F_{dv}(\boxed{1}), \begin{bmatrix} \text{ARG-ST} \left\langle \boxed{2}, \boxed{3} \right\rangle \end{bmatrix} \right\rangle \end{bmatrix} \& \ \boxed{2} >> \boxed{3}$$

$$\begin{bmatrix} \textit{inverse-verb-lex-rule} & \\ \text{INPUT} & \left\langle \boxed{1}, \ldots \right\rangle \\ \text{OUTPUT} & \left\langle F_{iv}(\boxed{1}), \begin{bmatrix} \text{ARG-ST} \left\langle \boxed{2}, \boxed{3} \right\rangle \end{bmatrix} \right\rangle \end{bmatrix} \& \ \boxed{2} << \boxed{3}$$

However, no such mechanism is available to us, so another method of analyzing scales is required.[3] It would be possible, of course, to simply create a lexical rule for each possible pair of positions on the scale, but this would mean having on the order of n^2 lexical rules for an n-position scale. It would be better to somehow model the scale with a type hierarchy.

Perhaps, noticing that it is necessary to address ranges of the scale that start at the left or the right end, we might try to model the scale using a type hierarchy like (67) (labeling the positions on the scale from 1 through 5), which is then used to constrain the series of lexical rules in (68) (which all derive from a single rule that applies the direct morphology to the verb):[4]

(67)



[3] Note, however, that other systems for implementing HPSG grammars are more powerful. In particular, the TRALE system (Meurers et al. 2002) can state constraints like those in (66) using its complex antecedent feature (Stefan Müller, personal communication, October 2008).

[4] This analysis models scales using subtypes of *synsem*, anticipating that the features involved may be syntactic or semantic. It is possible that a more specific feature structure would do (e.g., *local* or something within *cat* or *cont*), in some or all languages. This is left for future work.

(68)

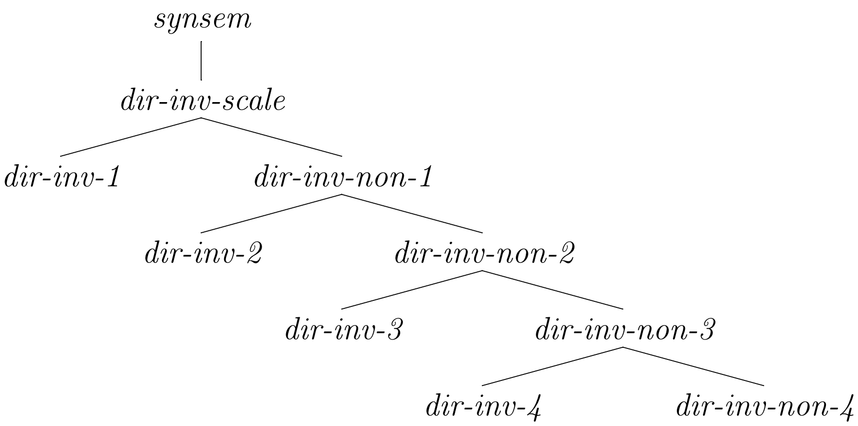
$$\begin{bmatrix} \textit{direct-verb-lex-rule-1} & \\ \text{ARG-ST} & \left\langle \textit{1},\ \textit{2-to-5} \right\rangle \end{bmatrix} \qquad \begin{bmatrix} \textit{direct-verb-lex-rule-2} & \\ \text{ARG-ST} & \left\langle \textit{1-to-2},\ \textit{3-to-5} \right\rangle \end{bmatrix}$$

$$\begin{bmatrix} \textit{direct-verb-lex-rule-3} & \\ \text{ARG-ST} & \left\langle \textit{1-to-3},\ \textit{4-to-5} \right\rangle \end{bmatrix} \qquad \begin{bmatrix} \textit{direct-verb-lex-rule-4} & \\ \text{ARG-ST} & \left\langle \textit{1-to-4},\ \textit{5} \right\rangle \end{bmatrix}$$

Unfortunately, when I created an experimental grammar with this set of rules, it turned out that they produced spurious ambiguity when applied to some sentences—that is, sentences with only one reading incorrectly received more than one analysis. While a sentence with, say, a subject from position 1 and an object from position 2 would parse just once with *direct-verb-lex-rule-1* having applied to the verb, a sentence with a subject from position 1 and an object from position 5 would parse four times, once for each of the above rules.

I addressed this problem by revising the *dir-inv-scale* hierarchy. Rather than having ranges that extend from both ends, the revised hierarchy consists of pairs of types, one covering a single position in the scale and the other the rest of the scale to the right, arranged into a right-branching tree:

(69)



To prevent spurious parses, the type hierarchy must constrain the appropriate syntactic features on both the leaves and the non-terminal nodes of the tree. For example, a sentence with a verb in the direct form and an agent that is compatible with *dir-inv-1* will parse just once, as long as the patient is compatible with *dir-inv-non-1*, with *direct-verb-lex-rule-1* having applied to the verb. Constraining all the types in this way sometimes necessitates the insertion of additional types into existing type hierarchies; for example, if the first position specified a noun phrase with [PERSON *1st*], then the type *dir-inv-non-1* would need to be constrained to have any value of PERSON other than *1st*. If no appropriate type for stating this constraint exists in the *person* hierarchy, it must be inserted by the customization system during grammar creation. This is accomplished by **hierarchy augmentation**, a process described below in §5.4.3.

4.2.1 SC-ARGS

An additional mechanism is necessary to model agreement in some direct-inverse languages. It is common in the world's languages for verbs to agree with their arguments—more usually the subject, but not uncommonly the object—in one or more features. HPSG includes a mechanism for modeling such agreement: the ARG-ST list, a feature on signs, whose first element is identified with the grammatical subject (on the SUBJ list) and whose other members are identified with the object(s) (on the COMPS list).[5] Constraints placed on the members of this list, such as the specification of feature values, will constrain the grammatical subject and object (see §3.2.2 for some examples of the use of ARG-ST).

However, in some direct-inverse languages, the clearest way of describing agreement is not "agrees with the subject" or "agrees with the object", but rather "agrees with the higher-ranked argument" or "agrees with the lower-ranked argument", where ranking is according to the same grammatical scale that controls argument marking.

[5] At least, this is true for syntactically accusative languages. In syntactically ergative languages, the subject and object are reversed on the ARG-ST list (see Manning and Sag 1995 for details). See also §3.2.2 for more about syntactic ergativity and the customization system.

The customization system provides an analysis of languages of this type using an additional list feature, parallel to ARG-ST, called SC-ARGS. This list contains the arguments of the verb in order from highest-ranked to lowest-ranked—that is, if the verb is direct, the SC-ARGS list will contain first the subject, then the object; if the verb is inverse, SC-ARGS will contain first the object, then the subject. Using SC-ARGS, grammars can address the highest-ranked or lowest-ranked argument in addition to the grammatical subject and object.

The order of the elements on SC-ARGS is guaranteed by placing additional constraints on the lexical rules for transitive verbs described in (72) above. For direct and inverse verbs, respectively, these constraints are:

(70) Direct:

$$\begin{bmatrix} \text{SC-ARGS} & \left\langle \boxed{1}, \boxed{2} \right\rangle \\ \text{VAL} & \begin{bmatrix} \text{SUBJ} & \left\langle \boxed{1} \right\rangle \\ \text{COMPS} & \left\langle \boxed{2} \right\rangle \end{bmatrix} \end{bmatrix}$$

Inverse:

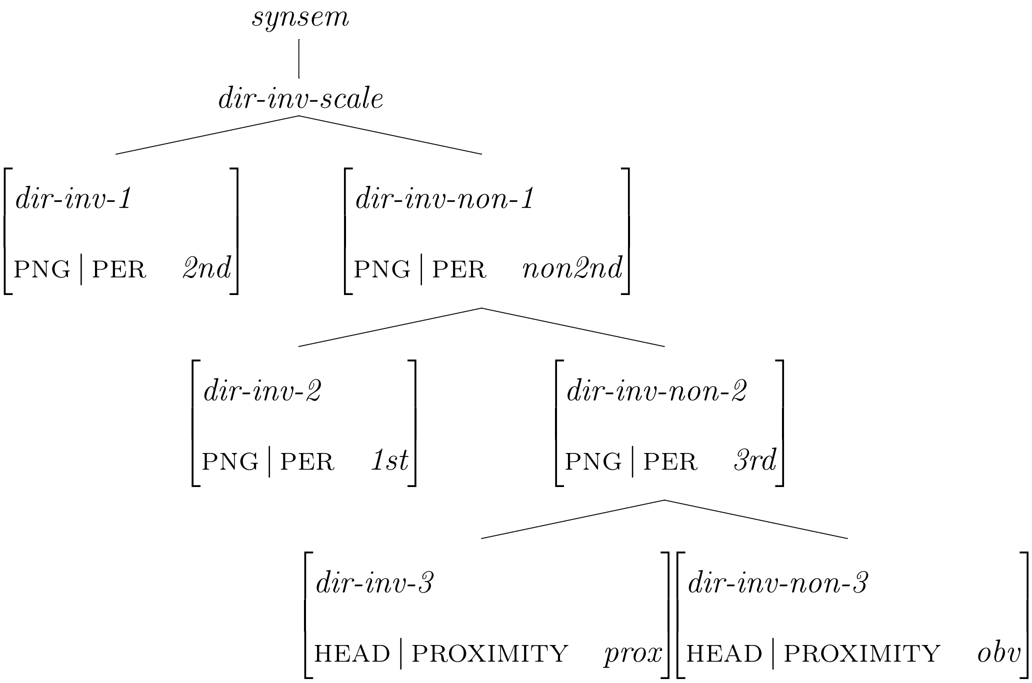
$$\begin{bmatrix} \text{SC-ARGS} & \left\langle \boxed{1}, \boxed{2} \right\rangle \\ \text{VAL} & \begin{bmatrix} \text{SUBJ} & \left\langle \boxed{2} \right\rangle \\ \text{COMPS} & \left\langle \boxed{1} \right\rangle \end{bmatrix} \end{bmatrix}$$

For detailed examples of the SC-ARGS feature in operation, see the test case for Cree in §4.4.1 below and the extended case study of Sahaptin that forms Chapter 6, in particular the three scale-sensitive enclitics in Table 6.10.

4.2.2 Algonquian

For a concrete example, below I have provided a type hierarchy (71) and lexical rules (72) that can be used to analyze an Algonquian language with the scale in (59):

(71)



(72)

$$\begin{bmatrix} \textit{direct-verb-lex-rule-1} \\ \text{HEAD} \mid \text{DIRECTION } \textit{dir} \\ \text{ARG-ST } \left\langle \textit{dir-inv-1},\ \textit{dir-inv-non-1} \right\rangle \end{bmatrix} \begin{bmatrix} \textit{inverse-verb-lex-rule-1} \\ \text{HEAD} \mid \text{DIRECTION } \textit{inv} \\ \text{ARG-ST } \left\langle \textit{dir-inv-non-1},\ \textit{dir-inv} \right\rangle \end{bmatrix}$$

$$\begin{bmatrix} \textit{direct-verb-lex-rule-2} \\ \text{HEAD} \mid \text{DIRECTION } \textit{dir} \\ \text{ARG-ST } \left\langle \textit{dir-inv-2},\ \textit{dir-inv-non-2} \right\rangle \end{bmatrix} \begin{bmatrix} \textit{inverse-verb-lex-rule-2} \\ \text{HEAD} \mid \text{DIRECTION } \textit{inv} \\ \text{ARG-ST } \left\langle \textit{dir-inv-non-2},\ \textit{dir-inv-2} \right\rangle \end{bmatrix}$$

$$\begin{bmatrix} \textit{direct-verb-lex-rule-3} \\ \text{HEAD} \mid \text{DIRECTION } \textit{dir} \\ \text{ARG-ST } \left\langle \textit{dir-inv-3},\ \textit{dir-inv-non-3} \right\rangle \end{bmatrix} \begin{bmatrix} \textit{inverse-verb-lex-rule-3} \\ \text{HEAD} \mid \text{DIRECTION } \textit{inv} \\ \text{ARG-ST } \left\langle \textit{dir-inv-non-3},\ \textit{dir-inv-3} \right\rangle \end{bmatrix}$$

A further set of lexical rules that are sensitive to the value of the DIRECTION feature are defined by the user-linguist in the lexicon section of the questionnaire.

These rules actually apply whatever spelling changes are associated with the direct and inverse forms of the verb; for example, handling the Fox examples in (60) and (61) would require a direct-marking rule for the suffix -*aa* and an inverse-marking rule for the suffix -*ek*. It would be possible in principle to merge the scale-constraining rules like those in (72) and the rules marking direct or inverse on the verb into a single paradigm of lexical rules; however, the questionnaire allows any number of morphological "slots" to be created that are sensitive to the DIRECTION feature, raising the question of which slot's rules should also specify the constraints in (72). To avoid this issue, the customization system always separates the scale-constraining rules from any lexical rules that implement user-defined verb morphology.

Note that this analysis does not allow the parsing of transitive sentences where both NP arguments occupy the same position on the scale. This is correct for at least some Algonquian languages including Nishnaabemwin [otw], where coreferential NP arguments require a reflexive form and two third person arguments can be distinguished using the obviative (Valentine 2001:273). Another possibility, languages where both NP arguments may occupy the same position on the scale, is analyzed below in §4.1.2.

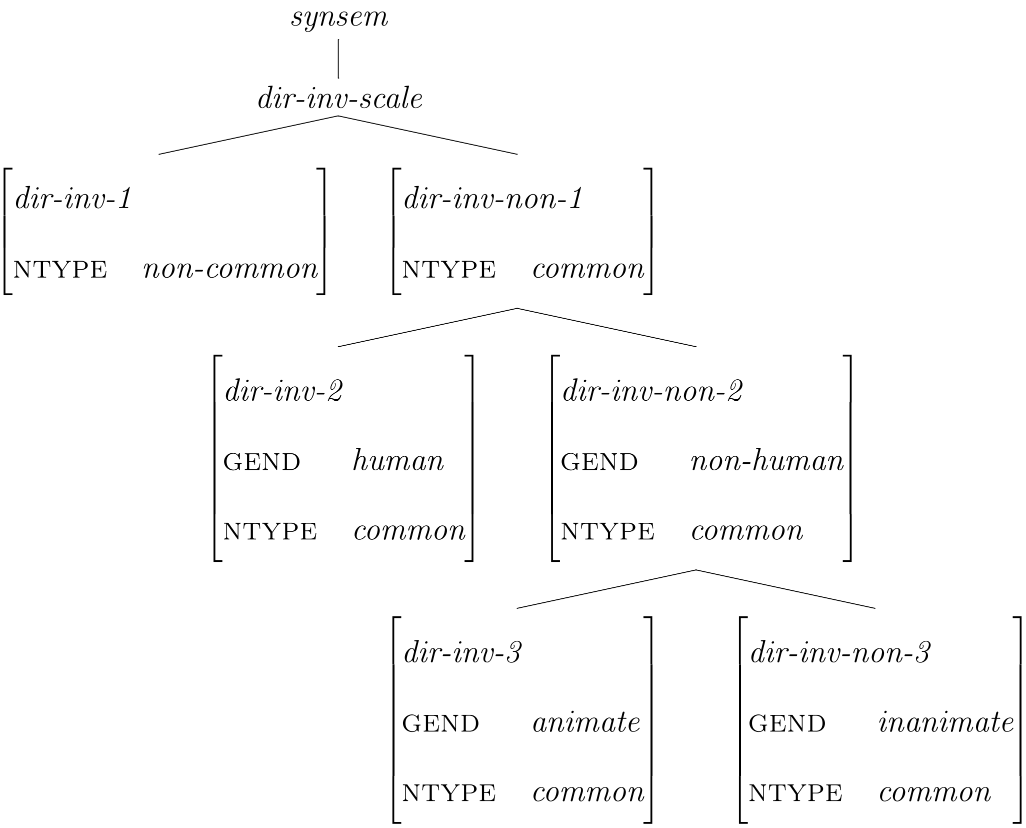
It is worth noting some drawbacks to this analysis. First, it requires, for a scale with n positions, $2(n-1)$ lexical rules. Furthermore, the type hierarchy in (71) is only arbitrarily right-branching. An analysis could just as easily have been built around a left-branching hierarchy. Having two equally-valid analyses with nothing to choose between them may seem like luxury, but it could also be argued that it results from the inability of the formalism being used to compactly and efficiently express the linguistic generalization being analyzed.

Finally, it should be noted that the leaf types in the *dir-inv-scale* hierarchy, which are certainly necessary because they encode the positions on the grammatical scale, need not be arranged in a single hierarchy in order to model the language. The leaves could all be independent subtypes of *synsem*, and the verb lexical rules could be stated in exactly the same way without a *dir-inv-scale* supertype. This is because the subtypes of *dir-inv-scale* operate by constraining features, not by using the inheritance relationships between types. However, there is a good reason to prefer a hierarchy to independent types. In (71), the features of the types *dir-inv-2* and *dir-inv-non-2* had better be compatible with those of *dir-inv-non-1*—otherwise, the latter type cannot be opposed with *dir-inv-1* in verb argument structures to distinguish NPs at the left of the scale from NPs at any position further down the scale. Since software systems can contain bugs, it is therefore valuable, as a "sanity check" on grammars produced by the customization system, to arrange the leaf types into a hierarchy. If the types are not compatible, loading the grammar with the LKB will produce an error rather than apparently succeeding but parsing and generating incorrectly. In other words, it ought to be possible to arrange the types encoding the grammatical scale into a hierarchy, and in fact, the grammar is seriously inconsistent if they cannot be so arranged, so to be safe, the customization system does so.

4.2.3 Fore

I analyze Fore as an ergative-absolutive language, where ergative case is marked by the suffix *-wama* (which Scott refers to as a "delineator"). Fore has a distinction between non-common and common nouns, and among the common nouns, there are three genders: human, animate, and inanimate. To model these distinctions, I use two features: an NTYPE feature on nominal heads that takes the values *common* and *non-common*, and a GEND feature on PNG under INDEX that takes the values *human*, *non-human*, *animate*, and *inanimate* (where the latter two are subtypes of *non-human*). The *dir-inv-scale* hierarchy in the grammar is:

(73)



The grammar also contains a set of constant verb lexical rules, one of which will apply to the verb in each transitive clause, constraining the items on its ARG-ST list:

(74)

$$\begin{bmatrix} \textit{direct-verb-lex-rule-1} \\ \textsc{head}\,|\,\textsc{direction}\ \textit{dir} \\ \textsc{arg-st}\ \left\langle \textit{dir-inv-1},\ \textit{dir-inv-scale} \right\rangle \end{bmatrix} \begin{bmatrix} \textit{inverse-verb-lex-rule-1} \\ \textsc{head}\,|\,\textsc{direction}\ \textit{inv} \\ \textsc{arg-st}\ \left\langle \textit{dir-inv-non-1},\ \textit{dir-inv-1} \right\rangle \end{bmatrix}$$

$$\begin{bmatrix} \textit{direct-verb-lex-rule-2} \\ \textsc{head}\,|\,\textsc{direction}\ \textit{dir} \\ \textsc{arg-st}\ \left\langle \textit{dir-inv-2},\ \textit{dir-inv-non-1} \right\rangle \end{bmatrix} \begin{bmatrix} \textit{inverse-verb-lex-rule-2} \\ \textsc{head}\,|\,\textsc{direction}\ \textit{inv} \\ \textsc{arg-st}\ \left\langle \textit{dir-inv-non-2},\ \textit{dir-inv-2} \right\rangle \end{bmatrix}$$

$$\begin{bmatrix} \textit{direct-verb-lex-rule-3} \\ \textsc{head}\,|\,\textsc{direction}\ \textit{dir} \\ \textsc{arg-st}\ \left\langle \textit{dir-inv-3},\ \textit{dir-inv-non-2} \right\rangle \end{bmatrix} \begin{bmatrix} \textit{inverse-verb-lex-rule-3} \\ \textsc{head}\,|\,\textsc{direction}\ \textit{inv} \\ \textsc{arg-st}\ \left\langle \textit{dir-inv-non-3},\ \textit{dir-inv-3} \right\rangle \end{bmatrix}$$

Compare the ARG-ST constraints in the rules in (74) with those in (72). The inverse rules are similar, but notice that the direct rules for Fore, rather than constraining agents and patients using types from the same level in the hierarchy, instead constrain patients to types that are the supertypes of their corresponding agents. For example, in *direct-verb-lex-rule-1*, *dir-inv-1* is opposed with *dir-inv-scale* rather than with *dir-inv-non-1*. This is necessary because Fore, unlike the Algonquian languages described in §4.2.2, allows clauses where both arguments occupy the same position on the scale (Scott 1978:115).[6] The customization system allows the user to describe either type of languages in its questionnaire (see §4.3 for details).

After one of the above rules has applied to a verb stem, another constant verb lexical rule from the set below applies. These rules are sensitive to the value of the DIRECTION feature and constrain the case of the verb's arguments appropriately.

[6] The delineator in Fore can also be used to make available dispreferred word orders with scale-equivalent arguments, but the current version of the customization system is not powerful enough to capture such an interaction between word order and argument marking. This grammatical fact must therefore be left for future work.

(75)
$$\begin{bmatrix} \textit{direct-lex-rule} & \\ \text{HEAD}\,|\,\text{DIRECTION} & \textit{dir} \\ \text{SUBJ} & \left\langle \begin{bmatrix}\text{HEAD}\,|\,\text{CASE}\ \textit{nom}\end{bmatrix} \right\rangle \\ \text{COMPS} & \left\langle \begin{bmatrix}\text{HEAD}\,|\,\text{CASE}\ \textit{nom}\end{bmatrix} \right\rangle \end{bmatrix}$$

$$\begin{bmatrix} \textit{inverse-lex-rule} & \\ \text{HEAD}\,|\,\text{DIRECTION} & \textit{inv} \\ \text{SUBJ} & \left\langle \begin{bmatrix}\text{HEAD}\,|\,\text{CASE}\ \textit{erg}\end{bmatrix} \right\rangle \\ \text{COMPS} & \left\langle \begin{bmatrix}\text{HEAD}\,|\,\text{CASE}\ \textit{nom}\end{bmatrix} \right\rangle \end{bmatrix}$$

Note that constraints on the rules in (74) and (75) could have been folded into a single paradigm of rules by having the direct rules derive from *direct-lex-rule* and the inverse rules from *inverse-lex-rule*. However, because this analysis of Fore treats it as a direct-inverse language, the structure of the lexical rule system produced by the customization system parallels that in §4.2.2 above, with separate two sets of rules, one implementing scale constraints and the other marking clauses as direct or inverse (via verb morphology in Algonquian and via case-marking in Fore).

4.3 Questionnaire

The section of the questionnaire devoted to direct-inverse languages needs to elicit two pieces of information from the user-linguist. Once these are provided, the grammar produced by the customization system will contain all the structures necessary for a direct-inverse language, including a set of properly-constrained lexical rules for transitive verbs as in (69). Describing the target language as direct-inverse also causes some additional values to become available on the Lexicon page; in particular, when choosing the valence of a verb (transitive or intransitive), the user-linguist may further choose whether transitive verbs follow the direct-inverse pattern or not, and when marking features on verbs, the user-linguist may specify them as applying to the higher- or lower-ranked argument (as described in §4.2.1).

The first, and more complex, piece of information is the grammatical scale that controls argument marking in the target language. This is accomplished using an iterator, each iteration of which allows the definition of the features that characterize one position on the scale. The user is asked to describe the scale positions from highest (most agent-like) to lowest (most patient-like).

The second, and much simpler, piece of information is the behavior of the target language when the two verbal arguments occupy the same position on the scale. There are two possible answers: the verb is in direct form, or there is some other, special form of the verb when the arguments are scale-equal, such as a reflexive. (In the latter case, the user will have to define that form by hand later, since the customization system does not yet support reflexives.)

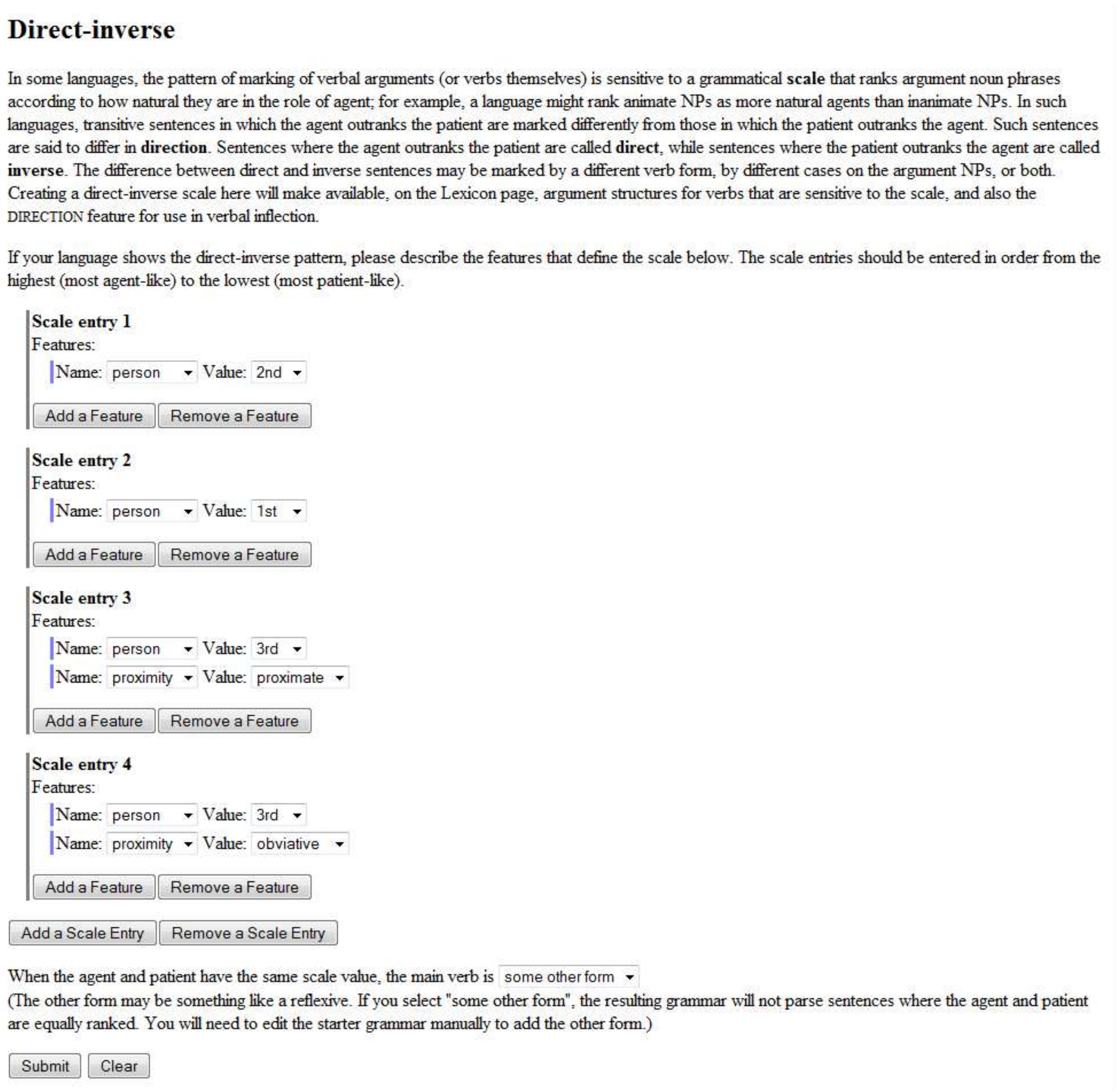
The direct-inverse section of the questionnaire, filled out for the Algonquian language Cree described in §4.4.1 below, is shown in Figure 4.1.

4.4 Test Cases

In order to test the direct-inverse section of the customization system, I have filled out the questionnaire and created two small grammars, one for a fragment of Cree, an Algonquian language, and the other for a fragment of Fore. Below, I show that both grammars have the expected coverage on a set of sentences designed to test the direct-inverse marking pattern. In addition, the extended case study in Chapter 6 is based on another language, Sahaptin, that displays the direct-inverse pattern.

4.4.1 Cree

Plains Cree [crk] (Algonquian), as described by Dahlstrom (1991), has a complex pattern of agreement and argument marking. Its argument marking is sensitive to the usual scale for Algonquian languages, repeated here for convenience:



Direct-inverse

In some languages, the pattern of marking of verbal arguments (or verbs themselves) is sensitive to a grammatical **scale** that ranks argument noun phrases according to how natural they are in the role of agent; for example, a language might rank animate NPs as more natural agents than inanimate NPs. In such languages, transitive sentences in which the agent outranks the patient are marked differently from those in which the patient outranks the agent. Such sentences are said to differ in **direction**. Sentences where the agent outranks the patient are called **direct**, while sentences where the patient outranks the agent are called **inverse**. The difference between direct and inverse sentences may be marked by a different verb form, by different cases on the argument NPs, or both. Creating a direct-inverse scale here will make available, on the Lexicon page, argument structures for verbs that are sensitive to the scale, and also the DIRECTION feature for use in verbal inflection.

If your language shows the direct-inverse pattern, please describe the features that define the scale below. The scale entries should be entered in order from the highest (most agent-like) to the lowest (most patient-like).

Scale entry 1
Features:
Name: person Value: 2nd

Scale entry 2
Features:
Name: person Value: 1st

Scale entry 3
Features:
Name: person Value: 3rd
Name: proximity Value: proximate

Scale entry 4
Features:
Name: person Value: 3rd
Name: proximity Value: obviative

When the agent and patient have the same scale value, the main verb is some other form
(The other form may be something like a reflexive. If you select "some other form", the resulting grammar will not parse sentences where the agent and patient are equally ranked. You will need to edit the starter grammar manually to add the other form.)

Figure 4.1: The direct-inverse section of the questionnaire

(76) 2nd > 1st > 3rd proximate > 3rd obviative

Cree verbs are marked for agreement with both the subject and the object by prefixes and suffixes, some of which are synthetic and some of which are themselves sensitive to whether the clause is direct or inverse. Nonetheless, with a few simplifying assumptions, I was able to successfully use the customization system to describe and model a fragment of Cree containing a significant part of its verbal agreement pattern.

Word order in Cree is quite free, but there are a complex set of interactions with topicality and obviation that are beyond the capabilities of the customization system. To avoid this in my fragment, I selected VSO as the word order. I further described the fragment of Cree as follows: it has a person feature that distinguishes first, second, and third (leaving out the inclusive/exclusive distinction in order to focus on the argument-marking scale); it has an additional syntactic feature PROXIMITY with the values *proximate* and *obviative*; and it has the direct-inverse scale shown in (76). I did not describe a number distinction in Cree, although it has one, because number is not relevant to the direct-inverse scale.

In the lexicon, I defined only one verb: *se·kih*, a transitive animate verb—that is, a verb whose object is required to be animate—that follows the direct-inverse pattern. Each of its various forms is marked by both a prefix and a suffix, both of which can agree with either the subject or the object. The portion of its agreement paradigm that I modeled is shown in Table 4.1. Based on this paradigm, I analyze Cree as having the inflectional morphemes shown in Tables 4.2 and 4.3. Table 4.2 shows the prefixes, all of which agree with the more highly-ranked argument on the direct-inverse scale (an example of the SC-ARGS feature, described in §4.2.1, in action). Table 4.3 shows the suffixes. Note that there are two variants of the *-ik* suffix with different features specified.

Table 4.1: Cree agreement paradigm for *se·kih* (Dahlstrom 1991:21–3)

	2nd person	**1st person**	**3rd proximate**	3rd **obviative**
2nd person	—	*ki-se·kih-in*	*ki-se·kih-a·w*	*ki-se·kih-ima·wa*
1st person	*ki-se·kih-itin*	—	*ni-se·kih-a·w*	*ni-se·kih-ima·wa*
3rd proximate	*ki-se·kih-ik*	*ni-se·kih-ik*	—	*se·kih-e·w*
3rd obviative	*ki-se·kih-ikoyiwa*	*ni-se·kih-ikoyiwa*	*se·kih-ik*	—

(vertical axis=subject, horizontal axis=object)

Table 4.2: Cree verbal prefixes

	Higher-ranked NP		**Lower-ranked** NP	
prefix	PERSON	PROXIMITY	PERSON	PROXIMITY
ki-	*2nd*			
ni-	*1st*			
∅	*3rd*	*proximate*	3RD	OBVIATE

Table 4.3: Cree verbal suffixes

		Subject	**Subject**	**Object**	**Object**
suffix	DIRECTION	PERSON	PROXIMITY	PERSON	PROXIMITY
-in	*direct*			*1st*	
-a·w	*direct*			*3rd*	*proximate*
-e·w	*direct*	*3rd*	*proximate*		
-ima·wa	*direct*	*1st, 2nd*		*3rd*	*obviative*
-ikoyiwa	*inverse*	*3rd*	*obviative*	*1st, 2nd*	
-ik	*inverse*	*3rd*	*proximate*		
-ik	*inverse*	*3rd*	*obviative*	*3rd*	*proximate*
-itin	*inverse*	*1st*			

This single verb and set of inflectional morphemes are together sufficient to model a significant fragment of Cree. However, Cree has a phenomenon that the customization system does not yet support, namely the dropping of arguments, both subject and object. As Dahlstrom (1991:62) writes, "Third person arguments may be expressed with lexical NPs; in the absence of lexical subjects and objects the inflection on the verb functions pronominally." It was possible to describe the verbal inflection of Cree in the questionnaire, but I found it necessary to add several "dummy" pronouns to stand in as overt subjects and objects: *1* for first person, *2* for second person, *3* for third person proximate, and *obv* for third person obviative. For this reason, the test sentences for Cree, unlike the test sentences for most of the test cases in this dissertation, are not grammatical sentences of the real language.

The precise details of the fragment of Cree can be found in the choices file, which is in Appendix E. The set of test sentences, along with the grammaticality judgments assigned by the test grammar, is shown in Table 4.4.

4.4.2 Fore

I also described a fragment of Fore in the customization system questionnaire to produce a grammar. I described it as follows: case-marking is ergative-absolutive with cases labeled, following Scott (1978), *ergative* and *nominative*; it has the genders *human*, *animate*, and *inanimate*; it has an additional syntactic feature called NTYPE with the values *pronoun*, *name*, *kin*, and *common*; and it has the direct-inverse scale in (62), repeated here for convenience:

(77) pronoun, name, kin term > human > animate > inanimate

Modeling Fore required some compromises. Word order in Fore is verb-final (Scott 1978:113), but while the preferred order of NP arguments is subject-first, this can vary. As in Cree, word order in Fore has some interactions with obviation that the customization system cannot yet model. In particular, when both the subject and the object are ranked the same by the scale (e.g., both third person), then word order determines which is the subject; however, either may be marked by *-wama* (see below) to force it to be interpreted as the subject. To avoid this issue, I have described Fore as verb-final (i.e., either OSV or SOV allowed) in the questionnaire.

Fore has lexically specified accents, and also a system of accent induction in which some lexical items can force one of the next two syllables to be accented, while others can suppress the accent on one of the next two syllables (Scott 1978:40). However, the accent specification of each word is not systematically described in Scott's word list (and is beyond the capabilities of the LKB in any case), so accent induction has not been modeled—accents have been treated as stable.

Table 4.4: Test sentences: Cree

Grammatical	**Ungrammatical**	
kise·kihin 2 1	**(no prefix)**	**(no suffix)**
kise·kiha·w 2 3	*se·kihin 2 1	*kise·kih 2 1
kise·kihima·wa 2 obv	*se·kiha·w 2 3	*kise·kih 2 3
nise·kiha·w 1 3	*se·kihima·wa 2 obv	*kise·kih 2 obv
nise·kihima·wa 1 obv	*se·kiha·w 1 3	*nise·kih 1 3
se·kihe·w 3 obv	*se·kihima·wa 1 obv	*nise·kih 1 obv
kise·kihikoyiwa obv 2	*se·kihikoyiwa obv 2	*se·kih 3 obv
kise·kihik 3 2	*se·kihik 3 2	*kise·kih obv 2
kise·kihitin 1 2	*se·kihitin 1 2	*kise·kih 3 2
nise·kihikoyiwa obv 1	*se·kihikoyiwa obv 1	*kise·kih 1 2
nise·kihik 3 1	*se·kihik 3 1	*nise·kih obv 1
se·kihik obv 3		*nise·kih 3 1
	(wrong subject)	*se·kih obv 3
	*kise·kihin 1 1	
	*kise·kihin 3 1	**(wrong object)**
	*kise·kihin obv 1	*kise·kihin 2 2
	*kise·kiha·w 1 3	*kise·kihin 2 3
	*kise·kiha·w 3 3	*kise·kihin 2 obv
	*kise·kiha·w obv 3	*kise·kiha·w 2 2
	*kise·kihima·wa 1 obv	*kise·kiha·w 2 1
	*kise·kihima·wa 3 obv	*kise·kiha·w 2 obv
	*kise·kihima·wa obv obv	*kise·kihima·wa 2 2
	*nise·kiha·w 2 3	*kise·kihima·wa 2 1
	*nise·kiha·w 3 3	*kise·kihima·wa 2 3
	*nise·kiha·w obv 3	*nise·kiha·w 1 2
	*nise·kihima·wa 2 obv	*nise·kiha·w 1 1
	*nise·kihima·wa 3 obv	*nise·kiha·w 1 obv
	*nise·kihima·wa obv obv	*nise·kihima·wa 1 2
	*se·kihe·w 2 obv	*nise·kihima·wa 1 1
	*se·kihe·w 1 obv	*nise·kihima·wa 1 3
	*se·kihe·w obv obv	*se·kihe·w 3 2
	*kise·kihikoyiwa 2 2	*se·kihe·w 3 1
	*kise·kihikoyiwa 1 2	*se·kihe·w 3 3
	*kise·kihikoyiwa 3 2	*kise·kihikoyiwa obv 1
	*kise·kihik 2 2	*kise·kihikoyiwa obv 3
	*kise·kihik 1 2	*kise·kihikoyiwa obv obv
	*kise·kihik obv 2	*kise·kihik 3 1
	*kise·kihitin 2 2	*kise·kihik 3 3
	*kise·kihitin 3 2	*kise·kihik 3 obv
	*kise·kihitin obv 2	*kise·kihitin 1 1
	*nise·kihikoyiwa 2 1	*kise·kihitin 1 3
	*nise·kihikoyiwa 1 1	*kise·kihitin 1 obv
	*nise·kihikoyiwa 3 1	*nise·kihikoyiwa obv 2
	*nise·kihik 2 1	*nise·kihikoyiwa obv 3
	*nise·kihik 1 1	*nise·kihikoyiwa obv obv
	*nise·kihik obv 1	*nise·kihik 3 2
	*se·kihik 2 3	*nise·kihik 3 3
	*se·kihik 1 3	*nise·kihik 3 obv
	*se·kihik 3 3	*se·kihik obv 2
		*se·kihik obv 1
		*se·kihik obv obv

Note that, except for a few examples involving coordination, Scott's examples do not include free pronouns. However, some of the test sentences below do include the free pronoun *ae*, because the customization system does not yet support unexpressed verbal arguments.

Table 4.5: Fore nouns and pronouns

Lexical item	NTYPE	GENDER
ae 'he/she/it'	*pronoun*	*human*
ayore	*name*	*human*
naba: 'my father'	*kin*	*human*
wá 'man'	*common*	*human*
yaga: 'pig'	*common*	*animate*
naninta: 'food'	*common*	*inanimate*

In the lexicon, I defined the nominals shown in Table 4.5 and two inflectional morphemes that mark the ergative case: -*ma*, which is used on human nominals, and -*wama*, which is used on non-human nominals. I defined two verbs, both in the indicative mood and in forms that agree with third person arguments: the intransitive *kanaye* 'come' and the transitive *agaye* 'see'. Finally, I defined lexical rules like those in (75) to constrain argument case based on the direction of the verb: direct verbs mark both of their arguments as *nominative*, but inverse verbs mark the subject *ergative.*

The precise details of the fragment of Fore can be found in the choices file, which is in Appendix F. The set of test sentences, along with the grammaticality judgments assigned by the test grammar, is shown in Table 4.6.

4.5 Summary

In this chapter, I have described direct-inverse languages and explained how I analyzed them in HPSG and incorporated that analysis into the customization system. My analysis of Fore was novel; although its verbs are not marked as direct or inverse, I showed that its pattern of argument marking can be modeled by combining my analysis of case with exactly the same types and rules I use to analyze more traditional direct-inverse languages. Finally, I verified the proper functioning of the direct-inverse library by constructing test cases for fragments of Cree and Fore.

Table 4.6: Test sentences: Fore

Grammatical	**Ungrammatical**
ae kanaye	(intrans with ergative)
naba: kanaye	*aema kanaye
ayore kanaye	*naba:ma kanaye
wá kanaye	*ayorema kanaye
yaga: kanaye	*wáma kanaye
naninta: kanaye	*yaga:wama kanaye
ae ae agaye	*naninta:wama kanaye
wá ae agaye	
yaga: ae agaye	(direct with ergative)
naninta: ae agaye	*aema ae agaye
ae wá agaye	*aema wá agaye
wá wá agaye	*wáma wá agaye
yaga: wá agaye	*aema yaga: agaye
naninta: wá agaye	*wáma yaga: agaye
ae yaga: agaye	*yaga:wama yaga: agaye
wá yaga: agaye	*aema naninta: agaye
yaga: yaga: agaye	*wáma naninta: agaye
naninta: yaga: agaye	*yaga:wama naninta: agaye
ae naninta: agaye	*naninta:wama naninta: agaye
wá naninta: agaye	*ae aema agaye
yaga: naninta: agaye	*wá aema agaye
naninta: naninta: agaye	*yaga: aema agaye
wáma ae agaye	*naninta: aema agaye
yaga:wama ae agaye	*wá wáma agaye
naninta:wama ae agaye	*yaga: wáma agaye
yaga:wama wá agaye	*naninta: wáma agaye
naninta:wama wá agaye	*yaga: yaga:wama agaye
naninta:wama yaga: agaye	*naninta: yaga:wama agaye
ae wáma agaye	*naninta: naninta:wama agaye
ae yaga:wama agaye	
wá yaga:wama agaye	(both ergative)
ae naninta:wama agaye	*aema aema agaye
wá naninta:wama agaye	*wáma aema agaye
yaga: naninta:wama agaye	*yaga:wama aema agaye
	*naninta:wama aema agaye
	*aema wáma agaye
	*wáma wáma agaye
	*yaga:wama wáma agaye
	*naninta:wama wáma agaye
	*aema yaga:wama agaye
	*wáma yaga:wama agaye
	*yaga:wama yaga:wama agaye
	*naninta:wama yaga:wama agaye
	*aema naninta:wama agaye
	*wáma naninta:wama agaye
	*yaga:wama naninta:wama agaye
	*naninta:wama naninta:wama agaye

Chapter 5
AGREEMENT

Following Moravcsik (1978), I take **agreement** to be defined by the covariation of two grammatical constituents in some grammatical property or feature. More fully, Moravcsik's definition is as follows:

> [A] grammatical constituent A will be said to agree with a grammatical constituent B in properties C in language L if C is a set of meaning-related properties of A and there is a covariance relationship between C and some phonological properties of a constituent B_1 across some subset of the sentences of language L, where constituent B_1 is adjacent to constituent B and the only meaning-related non-categorial properties of constituent B_1 are the properties C. (Moravcsik 1978:333)

Moravcsik illustrates this definition using the example of English subject-verb agreement: A is the subject noun phrase, B is the verb, B_1 is the agreeing suffix *-s* (or else zero), and C is the pair of features person and number.

As a working definition of agreement this is nearly sufficient, but I would like to broaden it in two ways. First, Moravcsik requires agreement to be marked by a grammatical constituent adjacent to B, but this would rule out suppletive marking of agreement, since in suppletion there is no separate element doing the marking. I see no reason to exclude suppletive paradigms, which include for example the English verb *be*, from the phenomenon of agreement (and in fact the customization system is capable of supporting such paradigms). Second, Moravcsik restricts agreement features to those that are "meaning-related", but some of the features generally held to fall under agreement, notably gender, do not necessarily have a clear connection with meaning in all languages. Therefore, the system I describe here is capable of modeling agreement of any grammatical feature, whether or not it is "meaning-related".

The term **concord** is sometimes used to refer to phenomena that fall under this definition of agreement. For example, Blake (2001) uses the term to refer to languages "[w]here a determiner or an attributive adjective displays marking for categories of the head noun such as case, number, or gender", but also mentions that "[t]his phenomenon is sometimes referred to as agreement" (Blake 2001:198). Rather than introduce a second term for the same phenomenon, I here use only the term agreement.

In a formalism like HPSG, agreement is a canonical example of what features are good for—features are defined to have a particular set of values, then both lexical rules that build up words and grammatical rules that build up sentences constrain the values of features by unification to in turn constrain the set of sentences parsed or generated by the grammar. The implementation of agreement in the Matrix customization system, therefore, consists of three parts: a way for a user of the system to define what features are relevant in the target language and what values they take, a way to define what lexical items and morphemes are marked for those features and values, and a way to describe the covariation among stems and morphemes by identifying their features.[1]

In this chapter, I will describe the terminology and typology of agreement phenomena, focusing on the typology presented by Corbett (2006). I will then describe the support in the customization system for agreement in the features gender, person, and number, as well as support for the automatic creation of complex feature hierarchies and for merged features.

[1] Grammatical machinery that implements the identification of features is produced by the customization system, but it also relies on cross-linguistically useful types in the core Matrix (e.g., *basic-head-comp-phrase*).

5.1 Typology of Agreement

Corbett (2006) describes a detailed typology for agreement. I here adopt his terminology for agreement, including:

> **controller**: the element which determines the agreement
> **target**: the element whose form is determined by agreement
> **domain**: the syntactic environment in which agreement occurs
> **features**: in what respect there is agreement
> **conditions**: other factors (like word order) which have an effect on agreement but are not directly reflected like features (Corbett 2006:4–5)

Corbett further bases his typology on the concept of **canonical agreement**, wherein he defines a set of criteria for a sort of idealized, prototypical agreement, acknowledging that "canonical instances, which are the best and clearest examples, those most closely matching the 'canon', may well not be the most frequent." (Corbett 2006:9). Corbett places his criteria, of which there are 20, into five groups: criteria related to controllers, to targets, to domains, to features, and to conditions. Each criterion is phrased as a statement about what makes an instance of agreement more canonical, of the form "more canonical > less canonical".

Below, I present Corbett's criteria for canonical agreement. After each criterion, I briefly explain it, then discuss whether the customization system's support for agreement is limited to canonical cases or also includes non-canonical cases. (There is no criterion for which the customization system *only* supports non-canonical agreement.) In this way, I hope to give a sense of the capabilities of the customization system's agreement support before the more detailed description later in this chapter.

Controllers:

C-1: controller present > controller absent

This criterion states, for example, that overt subjects are more canonical as controllers than dropped subjects. The customization system's support is currently only for canonical agreement, because there is, as yet, no way to describe such dropped arguments in the questionnaire. However, the analysis of agreement presented here should extend to dropped arguments when they are supported.

C-2: controller has overt expression of agreement features > controller has covert expression of agreement features

With respect to C-2, the customization system supports non-canonical agreement. Both overt and covert expression of features is supported.

C-3: consistent controller > hybrid controller

C-3 means that, for example, a controller that calls for plural agreement on a determiner but for singular agreement on a verb is less canonical than one that calls for plural agreement everywhere. The customization system supports non-canonical agreement. As will be discussed below, an arbitrary number of syntactic and semantic features can be defined, which allows such hybrid controllers to be modeled.

C-4: controller's part of speech is irrelevant > is relevant

This criterion states that it is more canonical for agreement to be consistent across related lexical types, rather than, for example, different depending on whether the controller is a noun or a pronoun. The customization system supports non-canonical agreement with respect to C-4. The specification of features and inflection can be conditioned on an arbitrary number of lexical classes, allowing finer distinctions in agreement than simply part of speech.

Targets:

C-5: bound > free

C-5 means, for example, that agreement marked by inflection is more canonical than marking via a free word. The customization system supports both kinds of marking; for example, case can be marked either by affixes or by adpositions.

C-6: obligatory > optional

With respect to C-6, the customization system supports non-canonical agreement, since inflectional morphemes can be specified as optional. This generally results in more ambiguity, but this is a property of the target language, not of the customization system.

C-7: regular > suppletive

Although it is generally simpler in the questionnaire for the user-linguist to describe a language where features and agreement are marked by regular, concatenative inflection, it is also possible to separately define each lexical item in a suppletive paradigm. The customization system therefore supports non-canonical agreement in this area.

C-8: alliterative > opaque

C-8 states, for example, that having masculine gender marked by *-o* everywhere is more canonical than having multiple masculine markers. The customization system supports non-canonical agreement, since it is possible to describe languages with any combination of phonetic forms marking agreement.

C-9: productive marking of agreement > sporadic marking

This criterion states, for example, that a language where agreement is marked on all adjectives is more canonical than one where it is marked on only a few.

The customization system supports non-canonical agreement with respect to C-9. Multiple lexical classes may be defined, and inflectional morphemes may be defined to apply only to some of them.

C-10: target always agrees > target agrees only when controller is absent

As with C-2, the customization system supports only canonical agreement with respect to C-10 until dropped arguments are supported.

C-11: target agrees with a single controller > agrees with more than one controller

The customization system supports agreement that is non-canonical with respect to C-11. Verbs, the only targets that can have multiple controllers in the current system, may agree with both agent and patient arguments, and even express that agreement through portmanteau morphemes.

C-12: target has no choice of controller > target has choice of controller

This criterion concerns languages in which targets can agree in some feature with more than one controller. Corbett (2006:18–19) gives the example of Tsez [ddo] (North Caucasian), in which verbs in certain sentences can agree with a clausal complement (taking default gender) or with a nominal *inside* the clausal complement. In the following example, the matrix verb 'know' agrees in gender (III) with the absolutive argument in the embedded clause:

(78) *eni-r* [*už-ā* *magalu* *b-āc'-ru-łi*]

Mother(II)-DAT boy(I)-ERG bread(III)[ABS] III-eat-PST_PTCP-NMLZ[ABS] *b-iy-xo*

III-know-PRS

‘The mother knows that the boy ate the bread.’ [ddo] (Corbett 2006:19)

The customization system does not support such agreement patterns, so only canonical agreement is supported here.

C-13: target’s part of speech is irrelevant > is relevant

This criterion is parallel to C-4 above, except that it has to do with targets rather than controllers. As with C-4, the customization system makes it possible to describe agreement patterns conditioned on arbitrarily many lexical classes, making available finer distinctions than part of speech. It therefore supports agreement patterns that are non-canonical according to C-13.

Domains:

C-14: asymmetric > symmetric

C-14 states that agreement is more canonical if one element is the controller and the other the target, rather than having the agreement on both elements conditioned on some external third factor. Corbett (2006:20–1) cites as an example of symmetric agreement the assignment of case to multiple elements in a noun phrase, though he acknowledges that whether this is symmetric or not depends on the framework of analysis. With respect to C-14, the customization system supports only canonical agreement, since for every pattern of agreement (noun-verb and noun-determiner) that can be described in the system, it is clear which element is the controller—namely, the one on whose HEAD or INDEX the agreeing feature appears (see §5.2 for a discussion of agreement and feature geometry).

C-15: local domain > non-local domain

This criterion states, for example, that agreement between a verb and its subject is more canonical than the agreement of a pronoun with an extra-clausal antecedent because the latter takes place in a local domain. Again, since the customization system only allows the description of verb-argument and noun-determiner agreement and not any kind of long-distance agreement, it only supports agreement that is canonical with respect to C-15.

C-16: domain is one of a set > single domain

This criterion states, for example, that a language with both agreement within the noun phrase and subject-verb agreement is more canonical than a language with only subject-verb agreement. The customization system does not condition agreement in one domain upon agreement in any other, so it supports languages that are non-canonical with respect to C-16.

Features:

C-17: feature is lexical > non-lexical

This criterion means that agreement in features whose assignment is "more semantically based" (Corbett 2006:24) (e.g., person or number) is less canonical than agreement in gender. The customization system allows agreement in all of these features to be described, so it supports non-canonical agreement here.

C-18: features have matching values > non-matching

C-18 states that agreement such as the English *the committee have decided*, where morphologically singular *committee* takes plural agreement, is less canonical than cases where a feature have consistent values. Through the mechanism of "other" features, described in §5.4.2 below, the customization system allows the modeling of such patterns, so it supports agreement that is non-canonical according to C-18.

C-19: no choices of feature values > choice of value

This criterion has to do with languages where more than one choice of feature is fully grammatical in sentences with the same meaning. Corbett uses the following Russian [rus] (Slavic) sentences as an example:

(79) *voš-l-o* *pjat′* *devušek*
come.in-PST-N.SG five[NOM] girl[PL.GEN]
'five girls came in' [rus]

(80) *voš-l-i* *pjat′* *devušek*
come.in-PST-PL five[NOM] girl[PL.GEN]
'five girls came in' [rus] (Corbett 2006:25)

Although the customization system does not include support for number words like *pyat′* 'five', it is possible to model agreement patterns of this sort. For example, if there were a noun that could take either singular or plural verbal agreement, it could be left underspecified for number in the lexicon. If, on the other hand, a language distinguishes between agreement within the NP and agreement with the verb, the language could be modeled with two number features, one syntactic and one semantic (see §5.2 for more on this distinction). The customization system therefore supports languages that are non-canonical according to C-19.

Conditions:

C-20: no conditions > conditions

This criterion concerns additional conditions on agreement beyond the controller, the target, the domain, and the features. The customization system is not capable of modeling such external conditions, so it only supports agreement that is canonical with respect to C-20.

Of Corbett's twenty criteria for canonical agreement, then, the customization system is capable of producing grammars that handle agreement that is non-canonical with respect to fourteen. It is only incapable of modeling agreement that is non-canonical according to six of the criteria, namely C-1, C-10, C-12, C-14, C-15, and C-20, and in several cases this is only because some other linguistic phenomenon (e.g., dropped arguments) is not yet supported.

5.2 Analysis of Agreement

Agreement can be either syntactic or semantic, depending on whether the feature that agrees appears on the syntactic HEAD or the semantic INDEX. Pollard and Sag (1994) provide an account of agreement that is entirely semantic. For example, to account for English subject-verb agreement, they would model a verb like *walks* as follows (Pollard and Sag 1994:82):

(81)
$$\begin{bmatrix} \text{PHONOLOGY} & \left\langle \textit{walks} \right\rangle \\ \text{SYNSEM}\,|\,\text{LOCAL} & \begin{bmatrix} \text{CATEGORY} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix}\text{VFORM} & \textit{fin}\end{bmatrix} \\ \text{SUBCAT} & \left\langle \text{NP}[\textit{nom}]_{\boxed{1}[3rd,sing]} \right\rangle \end{bmatrix} \\ \text{CONTENT} & \begin{bmatrix} \text{RELATION} & \textit{walk} \\ \text{WALKER} & \boxed{1} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

In this analysis, the person and number values on the INDEX of the subject noun phrase that the verb expects (*3rd* and *sing*, respectively) are constrained on the SUBCAT list of the verb.

Kathol (1999), on the other hand, provides a more elaborate analysis of agreement that can account for more complex agreement phenomena. Consider the following French example:

(82) *Vous êtes belle*
you are.PL beautiful.SG.FEM
'You are beautiful.' [fra] (Kathol 1999:239)

There are two kinds of agreement here. The polite second person pronoun *vous* is plural in form, but here refers to a single person. The verb *êtes* agrees with the pronoun in number and is marked plural, but the predicate adjective *belle* also agrees with the pronoun in number and is marked singular. Kathol analyzes this as two different kinds of agreement, modeled using two different number features on the pronoun. The first number feature appears under AGR inside HEAD and is identified with a similar AGR feature on the verb. Kathol refers to this as **morpho-syntactic agreement**. The second number feature appears on the INDEX, as in Pollard and Sag (1994). Kathol refers to this as **index agreement**.

My analysis of agreement is a hybrid of these analyses. By default, grammars produced by filling out the questionnaire model gender, person, and number using GENDER, NUMBER, and PERSON features on INDEX, as in Pollard and Sag (1994).[2] However, the questionnaire also allows the user-linguist to define other features and their associated values, and allows these additional features to be placed on either the syntactic HEAD or the semantic INDEX (see §5.4.2 for details of this "other features" mechanism). Unlike in Kathol's analysis, all agreement is modeled using constraints on the valence lists of targets (verbs and determiners), rather than having an AGR feature that appears on both controllers and targets. The ability to define both syntactic and semantic features provides enough flexibility to describe grammars with the sort of agreement pattern seen in (82) above.

Another effect of having two loci of feature specification can be seen in the MRS semantic representations produced by customization system grammars. Features speci-

[2] For clarity in the text of this dissertation, I refer to the gender, person, and number features as GENDER, PERSON, and NUMBER, respectively, but it should be noted that the customization system actually produces grammars with the abbreviated feature names GEND, PER, and NUM.

fied on the HEAD do not appear in the semantic representation, while features specified on the INDEX do. For example, consider the following MRS for the English sentence *I see him*:

```
[ LTOP: h1
  INDEX: e2 [ e SF: PROP-OR-QUES ... ]
  RELS <
    [ "exist_q_rel"
      LBL: h3
      ARG0: x4 [ x PNG.PERNUM 1SG ]
      RSTR: h5
      BODY: h6 ]
    [ "_pronoun_n_rel"
      LBL: h7
      ARG0: x4 ]
    [ "_see_v_rel"
      LBL: h1
      ARG0: e2
      ARG1: x4
      ARG2: x8 [ x PNG.PERNUM: 3SG ]]
    [ "exist_q_rel"
      LBL: h9
      ARG0: x8
      RSTR: h10
      BODY: h11 ]
    [ "_pronoun_n_rel"
      LBL: h12
      ARG0: x8 ]>
  HCONS < h5 qeq h7 h10 qeq h12 > ]
```

The index `x4` represents the pronoun *I*, while `x8` represents *him*. Notice that the PERNUM values of each appear in the MRS representation, while the values of CASE (*nom* and *acc*, respectively) do not appear because CASE is a purely syntactic feature.

The customization system allows the description of agreement in two domains. The first domain is the agreement between verbs and their mandatory nominal arguments: the subject and, for transitives, the object. To accomplish this, features may be specified in the lexicon on nominals and on verbs. Nominal features are always interpreted as specified on the HEAD or INDEX of the nominal. Features may also be specified on verbs, but with an additional question that asks whether that feature should be interpreted as constraining the HEAD or INDEX on the verb, on the subject, or on the object. The other domain of agreement is agreement between determiners and nominal heads. When a feature is specified on a determiner, it is in fact constrained on the first item on the SPEC list of the determiner—that is, the nominal head. So if a determiner is specified in the lexicon as having singular NUMBER, for example, it will only combine with a nominal head that has a compatible value of NUMBER.

In addition to the specification of features on lexical classes (e.g., a class of common nouns that are all [PERSON *3rd*]), the customization system allows the description of inflectional morphology; see §2.4.4 for a detailed explanation.

5.3 Features

Agreement phenomena involve two or more linguistic elements co-varying in some feature. My work has involved the addition of several features to the customization system, including gender, number, and person, as well as support for an arbitrary number of additional user-defined features.

In HPSG, agreement is analyzed using features (e.g., GENDER) whose values are drawn from hierarchies rooted in similarly-named types (e.g., *gender*). These feature-value types are generally what are known as **atomic types** or **sorts**—that is, they have no features of their own, relying only on inheritance relationships to constrain unification.

In order to implement each of the features discussed below in the customization system, it was necessary to answer three questions. First, what is the typological range of the feature cross-linguistically? Second, what sort of HPSG type hierarchy would be necessary to analyze that range? Third, how should the questionnaire be designed to prompt the user-linguist to describe these features in the target language?

5.3.1 Typology of Gender

Grammatical **gender**s are noun classes. As Corbett writes, "[T]he determining criterion of gender is agreement...[S]aying that a language has three genders implies that there are three classes of nouns which can be distinguished syntactically by the agreements they take." (Corbett 1991:4) Hence, not all noun classes are genders. Inflectional classes like the Latin declensions may control the form of inflectional morphemes on nouns, but they do not control agreement on verbs or nominal dependents. Similarly, English distinguishes syntactically between the classes of common and proper nouns, with the latter generally not taking determiners, but because there is no variation in agreement patterns conditioned by these noun classes, common and proper nouns are not genders in English. Only in cases where we see covariation of an agreeing element are we dealing with grammatical gender.

The assignment of nouns to genders varies cross-linguistically. In some languages, nouns are assigned to genders based primarily on semantics: if a noun's meaning falls into the appropriate semantic category, it is assigned to an associated gender. Languages with semantically-assigned gender systems include several Dravidian languages, several North-East Caucasian languages, the Australian language Diyari, and the Omotic language Dizi (Corbett 1991:8–12). For example, gender in Tamil is assigned according to the criteria shown in Table 5.1.

Table 5.1: Gender assignment in Tamil (Corbett 1991:9)

Criterion	**Gender**	**Example**	**Gloss**
god or male human	masculine (= male rational)	*civaɴ*	Shiva
		aaɳ	man
goddess or female human	feminine (= female rational)	*kaaḷi*	Kali
		peɳ	woman
other	neuter (= non-rational)	*maram*	tree
		viiṭu	house

In other languages, gender assignment is based on formal properties. There are two subtypes of formal gender assignment: phonological and morphological (Corbett 1991:33). In phonological systems, nouns that share some phonological property (e.g., ending in a vowel) are assigned to the same gender, perhaps with some exceptions. Languages with widespread phonologically-assigned gender include Qafar [aar] (East Cushitic), Hausa [hau] (Chadic), several Kru languages (Niger-Kordofanian), and to a lesser extent French [fra] (Romance) (Corbett 1991:51–61). In morphological systems, inflectional classes are conflated with gender, so that a noun's inflection determines its gender. Languages with widespread morphologically-assigned gender include Russian [rus] (Slavic) and the Bantu languages of Africa (Corbett 1991:34–49).

In still other languages, gender assignment is arbitrary; that is, nouns are assigned to genders without any semantic, morphological, or phonological basis. It is important to note that most languages with gender contain a mixture of the various kinds of gender assignment. It is quite common in Indo-European languages, for example, for nouns denoting human males and females to be assigned to different genders, but for non-human nouns to be assigned gender based on phonological or morphological conditions, or purely arbitrarily.

The size and shape of gender systems varies widely across languages. As a sample of this variety, consider four languages: Russian [rus] (Slavic), Tamil [tam] (Dravidian), Archi [aqc] (North-East Caucasian), and Swahili [swh] (Bantu). Russian has three genders: masculine, including male humans and higher animates plus residue (e.g., *otec* 'father', *djadja* 'uncle'); feminine, including female humans and higher animates plus residue (e.g., *mat'* 'mother', *tetja* 'aunt'); and neuter, including the remaining residue (e.g., *vino* 'wine', *taksi* 'taxi') (Corbett 1991:34–5).

Tamil has a similar system: masculine (gods and male humans), feminine (goddesses and female humans), and neuter (other) (for examples, see Table 5.1). However, Tamil also groups masculine and feminine into a larger category, rational, that shares some agreement properties (about which more in §5.3.2 below) (Corbett 1991:9).

Archi has four genders, the first two of which are simple: gender I contains only male rationals (e.g., *dija* ‘father’, *dozja* ‘grandfather’), while gender II contains only female rationals (e.g., *dozba* ‘grandmother’, *baba* ‘aunt’). Gender III and IV, however, are much more eclectic. Gender III contains domestic animals and birds (e.g., *xIon* ‘cow’, *x̄iIili* ‘bull’), larger wild animals and birds (e.g., *pil* ‘elephant’, *jam* ‘wolf’), all insects (e.g., *hilku* ‘fly’, *nibsu* ‘moth’), mythical beings (e.g., *žin* ‘genie’, *ilbis* ‘devil’), musical instruments (e.g., *parx* ‘drum’, *moxol* ‘tambourine’), cereals (e.g., *qoqol* ‘wheat’, *maxa* ‘barley’), trees (e.g., *had* ‘lime’, *kal* ‘fir’), water phenomena (e.g., *x̌at* ‘sea’, *baIri* ‘lake’), and astronomical and meteorological phenomena (e.g., *bac* ‘moon’, *marx̄əla* ‘snow’). Gender IV contains young animals and birds (e.g., *biš* ‘calf’, *k’eIrt* ‘foal (of donkey)’), smaller wild animals and birds (e.g., *ojomči* ‘hare’, *mejmanak* ‘monkey’), most tools and cutting instruments (e.g., *bel* ‘spade’, *dab* ‘awl’), cloth and most clothing (e.g., *at’ras* ‘satin’, *palatnoj* ‘linen’), metals (e.g., *lacut* ‘iron’, *qalaj* ‘tin’), liquids (e.g., *x̌an* ‘water’, *čixir* ‘wine’), and abstract concepts (e.g., *qIit̄aqI* ‘summer’, *mukul* ‘beauty’) (Corbett 1991:27–8).

Swahili, like all the Bantu languages, has a complex system of genders. The Bantu genders are traditionally labeled by pairs of numbers, with the first number determining the prefix of singular nouns in the class, and the second determining the prefix of plural nouns. Swahili gender is based on a mixture of semantic and morphological criteria, resulting in the assignment of nouns to genders 1/2, 3/4, 5/6, 7/8, 9/10, 11/10, and 15.[3] The noun *jogoo* ‘rooster’ is an example of semantic criteria overriding morphological criteria: it belongs to gender 1/2, even though nouns in that gender, such as *m-jusi* ‘lizard’, typically have the morphological form *m-/wa-* (Corbett 1991:46–7).

It is clear from these examples that both the number of genders and the structure of gender varies widely across languages. Few generalizations constraining either

[3] A few numbers are missing because Swahili has lost some of the proto-Bantu gender distinctions. Also, two of the genders, labeled 10, are identical across all forms, and 15 does not make a singular/plural distinction.

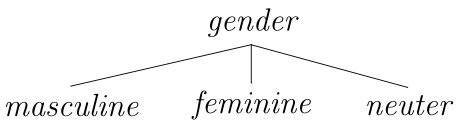
the size or complexity of gender systems are apparent. Consequently, the customization system must be flexible enough to allow the description and modeling of gender systems that are arbitrarily complicated.

5.3.2 Analysis of Gender

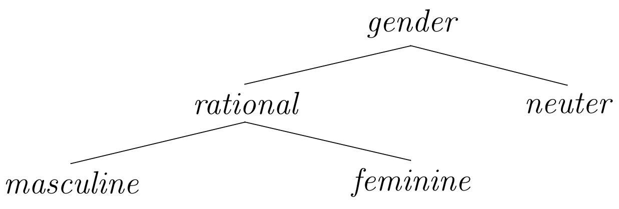
The customization system produces grammars that model gender using a feature GENDER whose value is drawn from a type hierarchy under *gender* tailored to the target language. GENDER is a feature of the nominal INDEX rather than the nominal HEAD since it is important that gender be represented in the semantic representation of a sentence. Otherwise, generating from the semantic representation obtained by parsing an English sentence like *he goes* would produce *he goes*, *she goes*, and *it goes*. Furthermore, there are other operations that could be performed on semantic representations, such as post-processing for coreference resolution, that would benefit from the specification of gender in those representations.

As can be seen from the diversity of gender systems described in §5.3.1 above, the structure of the *gender* type hierarchy can vary from very simple to very complex. For an illustration of this, consider the hierarchies below, each of which is appropriate for one of the languages in §5.3.1:

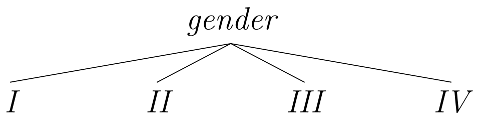
(83) Russian gender:



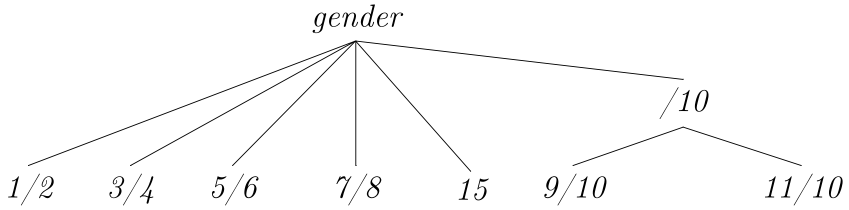
(84) Tamil gender:



(85) Archi gender:



(86) Swahili gender:



It is clear that the above gender hierarchies vary widely in the number of nodes, the labels on those nodes, and the complexity of the inheritance relationships. There is no reason to place an artificial limit on the complexity of gender hierarchies. Consequently, the questionnaire described below in §5.4 is designed to be flexible enough to allow the description of arbitrarily complex hierarchies with an unlimited number of nodes and any system of labels the user-linguist chooses.

5.3.3 Typology of Number

Grammatical **number** is another feature that, like gender, is primarily detectable through agreement patterns, but, unlike gender, is also associated with relatively consistent semantics cross-linguistically. For number, this meaning has to do with the number of real world entities referred to by a noun phrase.[4] In different languages, number can be marked on several different parts of speech, including nominals, nominal modifiers, and verbs, and different languages distinguish different values of number.

[4] Corbett (2000) also discusses verbal number, which has to do with the number or kind of *events* rather than the number of any noun phrase. Verbal number is beyond the scope of this work, and is not supported in the current version of the customization system.

Grammatical number is not the same phenomenon as number words (e.g., *one*, *two*, etc.), although the two phenomena certainly interact in many languages, with number words triggering agreement in grammatical number. Number words are not yet supported in the customization system. Grammatical number can be identified in patterns of agreement triggered by the quantity of entities referred to; however, as mentioned above, the number feature is associated with a consistent semantics such that it can be identified even in the absence of agreement. For example, in Japanese [jpn] (Japanese), pronouns and human nouns can be marked for plurality by the suffix *-tachi*, but there is no agreement in number between nominals and any other part of speech (Makino and Tsutsui 1989:440). Since it is still desirable to capture the distinction in meaning between a singular and a plural nominal—otherwise sentences with nominals that differ in number would be treated as synonymous—we should analyze Japanese as having a category of number, even though it lacks agreement.

Some values of number are associated with specific integer counts of real world entities: the **singular** refers to one entity, the **dual** to two, the **trial** to three, and the **quadral** (if it exists; see Corbett 2000:26–30) to four. In addition, some languages have number categories that refer to ranges of counts of real world entities. Such categories include the **paucal**, which refers to a few entities, and the **plural**, which refers to multiple entities (and sometimes to zero entities, as in English). Some languages subdivide the paucal into the lesser and greater paucals, and some divide the plural into the lesser and greater plurals (Corbett 2000:30–35). Such subdivided number values can be used in some cases to make a contrast between the quantities of two noun phrases. In others, particularly cases involving a subdivided plural, the two values have distinct semantics, as in Hamer [amf] (South Omotic), which has a particular plural, used for a particular number of entities, and a global plural, used for all entities in a class. Finally, some languages have a **general number**, often unmarked, that is used to leave a noun phrase unspecified for the number of real world entities referred to.

It is the case in some languages that all the values of the number feature are distinct, non-overlapping, and mandatory, but not in all languages. The categories paucal and plural, for example, vary cross-linguistically with respect to the range of numbers they can express. Corbett (2000:43) illustrates this contrast using Sanskrit [san] (Indo-Iranian), in which the dual must be used for noun phrases referring to two entities, and Slovene [slv] (Slavic), in which noun phrases referring to two entities can appear in the plural instead of the dual:

(87) *nóge* *me* *bolijo*
foot.PL 1.SG.ACC hurt.PL
'my feet hurt' [slv]

This pattern of number optionality is known as **facultative number**. Some languages have extensive facultative number. For example, Corbett (2000:46–47) describes the number system of Marshallese [mah] (Austronesian). Marshallese has five number values: singular, dual, trial, paucal, and plural, of which the use of the dual, trial, and paucal numbers, however, is optional.

Languages with facultative number demonstrate that we must take care when discussing categories like the plural cross-linguistically. The semantics of the singular, the dual, and the trial are quite consistent cross-linguistically, but the meaning of the plural varies from language to language. In Sanskrit, the plural covers three or more entities, whereas in Slovene is covers two or more, even though Slovene has a dual. There is similar variation in the paucal, both at the low end where it may or may not overlap with lower number values, and also on the high end, where there is cross-linguistic variation as to how many can still be considered "a few".

The cross-linguistic distribution of number values is not random. Greenberg's Universal 34 states, "No language has a trial number unless it has a dual. No language has a dual unless it has a plural." (Greenberg 1963:94) From this can be derived the Number Hierarchy, which governs the presence of certain number categories in languages:

(88) singular > plural > dual > trial (Corbett 2000:38)

To put it another way, this hierarchy states that a language with a plural also has a singular, that a language with a dual also has a plural, and that a language with a trial also has a dual.[5]

Other researchers have proposed extended versions of the Number Hierarchy, including Croft, who states that the existence of either trial or paucal implies the existence of dual:

(89) singular > plural > dual > trial/paucal (Croft 1990:66)[6]

The Number Hierarchy accounts for many of the number systems found in the world's languages, including those shown in Table 5.2. However, as we will see below in §5.3.4 below, the Number Hierarchy is not truly universal.

Table 5.2: Number systems consistent with the Number Hierarchy (Corbett 2000:39)

Language	**Numbers**
Russian [rus] (Slavic)	singular plural
Upper Sorbian [hsb] (Slavic)	singular dual plural
Larike [alo] (Austronesian)	singular dual trial plural
Yimas [yee] (Sepik-Ramu)	singular dual paucal plural

[5] In fact, as Corbett notes, Greenberg's Universal 34 does not say anything about the existence of the singular; instead, the singular appears in the Number Hierarchy simply because there seem to be no languages with a plural but no singular (whatever that might mean). It might therefore be clearer to state the number hierarchy as:

singular, plural > dual > trial

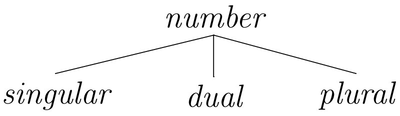
[6] Croft actually uses less-than signs rather than greater-than signs in his hierarchy, to denote "is less marked than" instead of "is more common than", but I have changed these to greater-than signs for consistency.

5.3.4 Analysis of Number

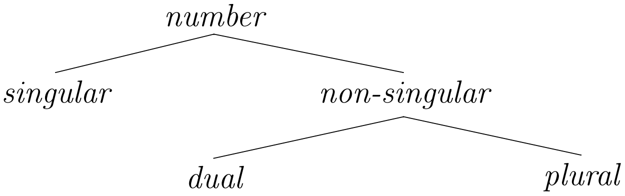
The grammars produced by the customization system model grammatical number using a feature NUMBER whose value is drawn from a type hierarchy under *number* tailored to the target language. As with GENDER, NUMBER is a feature of the nominal index so that it will appear in the semantic representation, preventing potential spurious ambiguity on generation and capturing the semantic distinctions.

To create the *number* hierarchy, it is necessary to determine the set of distinctive values of number in the target language. This is not as straightforward a process as it may seem due to the phenomenon of facultative number. To illustrate why, consider again Sanskrit and Slovene. Recall that the Sanskrit dual is obligatory when referring to two real world entities, but in Slovene the plural can be used instead. This might suggest the following *number* hierarchies for the two languages:

(90) Sanskrit:



(91) Slovene:



The hierarchy in (90) is sufficient to model Sanskrit: singular, dual, and plural noun phrases will have number values of *singular*, *dual*, and *plural*, respectively, and the morphology on the agreeing verb will be specified the same values on the verb's subject, thus licensing only sentences with correct agreement. The hierarchy in (91), however, is more complex than necessary to model Slovene. Notice that in (87), although it is true that the plural is used where we might expect the dual, we still see a plural agreeing with plural marking on the verb (and not, crucially, dual agreeing with plural). Since each value of number only needs to agree with itself, there is no need to postulate a *non-singular* value covering *dual* and *plural*. In fact, both Sanskrit and Slovene can be modeled using the same hierarchy, the one shown in (90).

Motivating a more articulated hierarchy for modeling number requires a language where number distinctions are neutralized in some loci of marking, but not in others. Such languages exist. Consider the following sentences in Modern Hebrew [heb] (South Semitic):[7]

(92) *ha-yom ʕavar maher*
DEF-day pass.PAST.3.SG.MASC quickly
'the day passed quickly' [hbr]

(93) *ha-yom-ayim ʕavru maher*
DEF-day-DUAL pass.PAST.3.PL quickly
'the two days passed quickly' [hbr]

(94) *ha-yam-im ʕavru maher*
DEF-day-PL pass.PAST.3.PL quickly
'the days passed quickly' [hbr] (Corbett 2000:95)

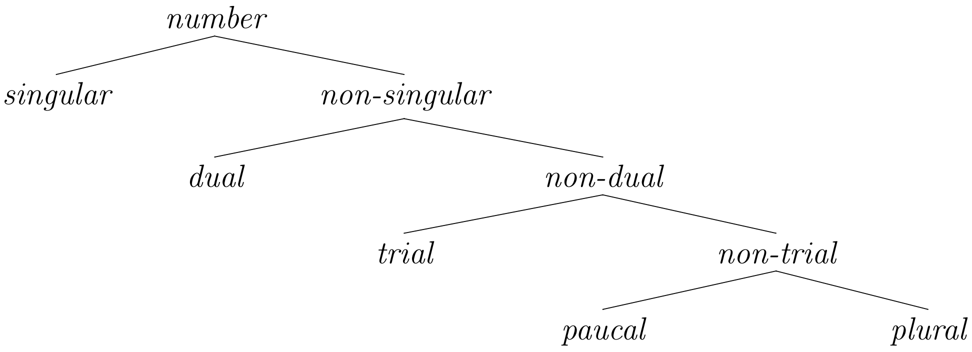
Notice nouns can be marked singular, dual, or plural, while verbs can only be marked for agreement with singular or plural. Such a mismatch calls for the type hierarchy in (91) above, with the "plural" inflection on the verb actually specified with a number value of *non-singular*, which can unify with either *dual* or *plural*. Without the additional *non-singular* type, modeling Modern Hebrew would require

[7] Although the transcription of these examples includes a pharyngeal stop (ʕ) for the Hebrew letter *ayin*, it is worth noting that this phone has been lost in most dialects of Modern Hebrew.

unnecessarily postulating two homophonous forms of the verb, one dual and one plural.

Notice that the more articulated hierarchy in (91) would be perfectly capable of modeling Sanskrit as well, as long as the *non-singular* type were not used—the leaf types *singular*, *dual*, and *plural* will not unify, and so can be used to model those distinct number values in Sanskrit. Given the postulated existence of the Number Hierarchy in (89) and the consequent similarity between the number systems of many languages, it might seem plausible that grammatical number could be modeled using a single, universal hierarchy, perhaps:

(95)



However, such a universal hierarchy is both undesirable and incorrect. It is undesirable because it postulates types that are not required to model the vast majority of languages. It also leads to ambiguity for the grammar writer about which value of number to choose. For example, in a language with a singular-plural distinction, *singular* should clearly be used to model the singular, but should plural be modeled using *plural* or using *non-singular*? This problem is even more apparent in languages that do not obey the Number Hierarchy. Consider Bayso [bsw] (Cushitic), which has four numbers: an unmarked general number and marked singular, paucal, and plural forms, which can be seen in the following examples:

(96) *lúban* *foofe*
lion.GENERAL watched.1.SG
'I watched lion(s)' [bsw]

(97) *lubán-titi foofe*
lion-SG watched.1.SG
'I watched a lion' [bsw]

(98) *luban-jaa* *foofe*
lion-PAUCAL watched.1.SG
'I watched a few lions' [bsw]

(99) *luban-jool foofe*
lion-PL watched.1.SG
'I watched (a lot of) lions' [bsw] (Corbett 2000:11)

This violates the constraint that languages with a paucal also have a dual, and furthermore makes choosing types to model Bayso difficult. Since the Bayso paucal covers noun phrases referring to two entities (i.e., the sort of noun phrases usually included in the dual), some grammar writers might be tempted to model it using *non-singular* instead of *paucal*, but this would produce errors, since *non-singular* will unify with *plural*. The customization system should not force grammar writers to make such confusing decisions and terminological compromises.

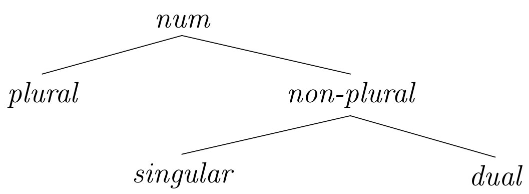
The universal hierarchy is also incorrect for some languages. For example, Hahm (2006) has argued that the agreement facts in American Sign Language [ase] (Deaf sign language) argue for a different hierarchy:

> Verb agreement tells us that ASL has a plural/non-plural number distinction and that the plural number is marked. Verbs do agree either in

> plural or non-plural number although the arguments, including number-incorporated pronouns, can denote any specific number of referents. Verbs in plural number agree with only plural agreement triggers. Otherwise, the default form of the verbs in singular is used. (Hahm 2006)

Hahm concludes that ASL is best modeled using the following number hierarchy:

(100)



This means that the universal hierarchy in (95), which lacks a type corresponding to *non-plural*, is unsuitable for modeling ASL. Therefore, in order to correctly model all languages, as well as to avoid the confusion mentioned above, the customization system allows the construction of an arbitrary type hierarchy for number tailored to the specific target language, rather than attempting to provide a single, universal number hierarchy.

5.3.5 Typology of Person

The grammatical category of **person** is defined in terms of **discourse participants** (Siewierska 2004:1); that is, person marking is the marking of what role a noun phrase takes with respect to an utterance: the speaker (the **first person**), the addressee or person spoken to (the **second person**), or some other person (the **third person**). Like gender, person is primarily detectable through agreement phenomena, though in languages without person marking on verbs or nominal modifiers, person agreement may only be found in anaphoric agreement. Unlike gender, however, and to an even greater extent than number, there is a cross-linguistically consistent semantics associated with person.

All languages have strategies for referring to speakers, addressees, and others; whether all languages have the grammatical category of person is more controversial. If a language shows person agreement on verbs, for example, this provides clear evidence for the displacement of grammatical information, namely the feature person, and therefore of its existence. If a language lacks verbal agreement in person but still has a lexical class of pronouns where first, second, and third person pronouns are only licensed by an antecedent referring to the appropriate discourse participant, this also constitutes evidence of person. However, there exist languages, including Thai, Burmese, Vietnamese, and Japanese, in which pronouns are difficult to distinguish, morphologically and syntactically, from nouns. Whether such languages should be analyzed as having person depends on the goals of the analysis (Siewierska 2004:8–13).

Some languages show complex interactions between person and number, particularly in the first person. In discussing these distinctions, I will use the notation of Cysouw (2003) for referring to groups of participants independent of any language-specific person or number feature. This notation is summarized in Table 5.3.

Table 5.3: Groups of discourse participants (Cysouw 2003:74)

Group	**Description**
1+1	‘we’, mass speaking
1+2	‘we’, including addressee, excluding other
1+3	‘we’, including other, excluding addressee
2+2	‘you-all’, only present audience
2+3	‘you-all’, addressee(s) and others
3+3	‘they’
1+2+3	‘we’, complete

These are the logically possible groups of discourse participants, but not all of them are distinguished in natural languages. There appears to be no language that has a special form for distinguishing 1+1 (a group of people speaking in unison) from other groups including the speaker, and probably no language that distinguishes 2+2 from other second person groups (Cysouw 2003:76).[8] Cross-linguistic variation does exist, however, in the marking of the groups 1+2, 1+3, and 1+2+3.

Cysouw (2003) is an investigation of paradigms within languages rather than of whole languages, whereas the Matrix customization system requires a typology of languages. Still, we can make use of his typology and examples, with the following adjustment: the type of a language is determined by the distinctions within the first person necessary to model every paradigm in the language, not just a single paradigm. Cysouw categorizes paradigms into the following types (Cysouw 2003:80–95), based on the distinctions they make in the first person:

- **Unified-we:** This kind of paradigm, which includes the English pronouns, has a single category covering 1+2, 1+3, and 1+2+3.

- **No-we:** Paradigms of this type lack a distinct form for the first person non-singular. Pirahã [myp] (Mura), a language of Brazil, appears to be of this type.

- **Only-inclusive:** Paradigms of this type have a form for 1+2 (the inclusive), but no special form for 1+3, which is generally referred to using the first person singular. Cysouw provides an example of a single paradigm of this type from Maká (or Maca) [mca] (Mataco-Guaicuru), but does not claim the whole language is of this type.

- **Inclusive/exclusive:** This quite common type of paradigm distinguishes between two categories: the first person **inclusive**, which refers to 1+2 and 1+2+3, and the first person **exclusive**, which refers to 1+3.

[8] Cysouw (2003:75–76) discusses claims that Abkhaz [abk] (North Caucasian) and Mao Naga [nbi] (Sino-Tibetan) have a distinction between 2+2 and 2+3, but concludes that the evidence for these claims is weak.

- **Minimal/augmented:** In paradigms of this type, all three of the groups 1+2, 1+3, and 1+2+3 are distinguished.

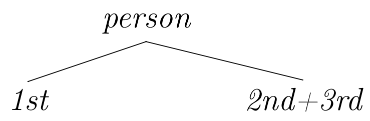
- **Rare types:** Cysouw identifies five other paradigm types that involve various patterns of neutralization (e.g., Kunimaipa [kup] (Trans-New Guinea) perfective suffixes, which include a form *-gi* that marks 1+2+3, 1+3, and 2+3), but none of these types covers a whole language, and none requires any additional group of discourse participants.

5.3.6 Analysis of Person

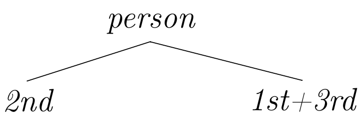
As with number above, grammars produced by the customization system analyze person using a feature PERSON whose value is drawn from a type hierarchy under *person* tailored to the target language. PERSON is a feature of the nominal index, and therefore appears in the semantic representation, preventing potential spurious ambiguity on generation and capturing the semantic distinctions. Otherwise, on parsing a sentence like the English *he goes*, the grammar, unconstrained by grammatical person, would also generate *I go* and *you go*.

As with the other agreement features, the first question that must be answered is what values of person are distinguished in the grammar of the target language. If the user-linguist decides to analyze the target language as lacking person altogether, then the PERSON feature and the *person* hierarchy are unnecessary. If the target language (or at least, the fragment being described) distinguishes only first from second and third, second from first and third, or third from first and second, this can be modeled using the following hierarchies, respectively:

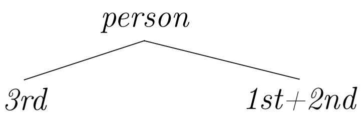
(101)



(102)

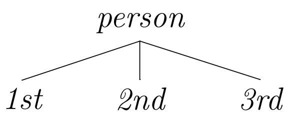


(103)



However, by far the most common case will be a target language that distinguishes all three values of person, which can be modeled using the following hierarchy:

(104)



A more complex hierarchy is required in the case where the target language makes finer distinctions (e.g., inclusive/exclusive) within the first person. Before the analysis of such languages can be discussed, however, we must first examine an alternative to separate PERSON and NUMBER features.

5.3.6.1 Merging Person and Number

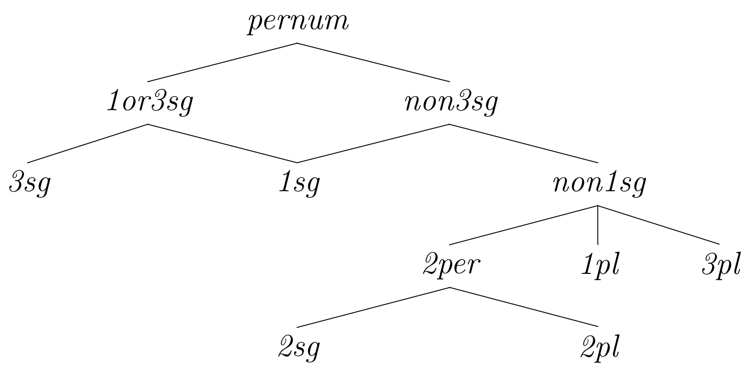
In some lexical types in some languages, the paradigms for person and number are very well-behaved: there is a distinct form for each possible combination of person and number. Such languages can be modeled using distinct features for person and number. However, other languages have neutralizations between different cells in the person/number paradigm. Flickinger (2000) argues that such languages are best modeled by merging the *person* and *number* hierarchies into a single *pernum* hierarchy.

English subject-verb agreement is an example of such a paradigm. In the present tense, all verbs except *be* only make a distinction between the third person singular form and everything else (e.g., *runs* and *run*). The verb *be*, on the other hand, makes more distinctions in the present tense, with special first person singular (*am*) and third person singular (*is*) forms. Furthermore, the pronouns have a neutralized distinction between the singular and plural in the second person (*you*).

It is desirable to model these distinctions in a way that avoids spurious ambiguity, which is difficult using separate *person* and *number* types. The second person pronoun *you*, for example, would have to be modeled using two homophonous second person lexical items, one singular and the other plural. This would cause every sentence containing the word *you* to parse more than once—in fact, a sentence with n occurrences of *you* would parse 2^n times.

To avoid this, Flickinger proposes to model person and number using the *pernum* type below:

(105)



Using this hierarchy, agreement on English verbs can be correctly modeled: *is* specifies a *3sg* subject; *am* specifies a *1sg* subject; *are* specifies a *non1sg*[9] subject; and the verbal suffix -*s* specifies a *non3sg* subject.

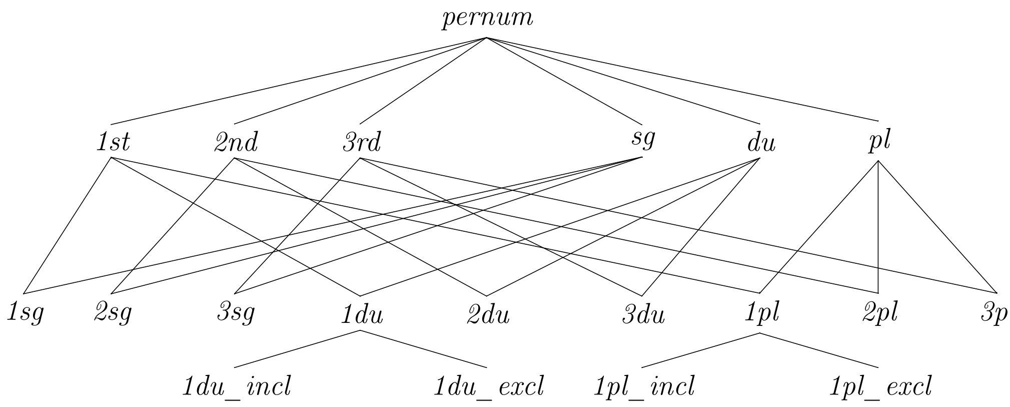
The customization system generally avoids trying to infer the existence of such a merged hierarchy from the answers provided by the user, though the user can define one directly using the "other features" mechanism described in §5.4.2. However, languages that distinguish additional values in the first person non-singular such as

[9] Note that in spite of its name *non1sg*, in addition to excluding the first person singular, also excludes the third person singular, since it is a subtype of *non3sg*.

inclusive and exclusive clearly call for a merged hierarchy because the paradigms in such a language do not form a simple grid of independent cells, instead subdividing one or more of the first person non-singular types.

To model such languages, the customization system will produce a *pernum* hierarchy. Consider, for example, a language that distinguishes three values of number (singular, dual, and plural), three values of person (first, second, and third), and also distinguishes an inclusive and exclusive in the first person dual and plural. For such a language, the customization system produces the following *pernum* hierarchy:

(106)



Note that this hierarchy is not flat, but includes intermediate types such as *1st*, *2nd*, *sg*, *du*, etc. These types may not be necessary for the analysis of any particular language, but it seems likely they will be, and since the *pernum* hierarchy is automatically created when required, it seems wise to provide the intermediate types that would be available if the *person* and *number* hierarchies had been separate.

Repeating hierarchies of such complexity in the discussion below would be difficult to read (not to mention redundant, since all *pernum* hierarchies produced by the customization system share very similar structure in the highest three levels), so they will hereafter be represented by a grid. For example, the hierarchy in (106) can be represented as follows:

(107)

	sg	du	pl
1st	1sg	1du_incl 1du_excl	1pl_incl 1pl_excl
2nd	2sg	2du	2pl
3rd	3sg	3du	3pl

5.3.6.2 Subdivided First Person

Having introduced *pernum* feature hierarchies, we can now turn to the analysis of languages that have a subdivided first person. For the inclusive/exclusive distinction, the analysis is straightforward: a hierarchy like that in (107), with subtypes in the non-singular first person values, will suffice to model the inclusive and exclusive.[10] However, the analysis of minimal/augmented languages is less clear, with three distinct analyses described by Cysouw (2003:85–90) for Ilocano [ilo] (Austronesian), which has the following pronouns (followed by the discourse participants they refer to): *co* (1), *mo* (2), *na* (3), *ta* (1+2), *tayo* (1+2+3), *mi* (1+3), *yo* (2+3), and *da* (3+3).

In the first (traditional) analysis, Ilocano is described as having a dual number that exists only in the pronoun system. Such a system could be modeled using the following hierarchy:

(108)

	sg	du	pl
1st	1sg	1du_incl	1pl_incl 1pl_excl
2nd	2sg	2pl	
3rd	3sg	3pl	

[10] Note that hierarchy for (107) does not include types *incl* or *excl* by default. However, it is a simple matter for the user to get such types via hierarchy augmentation; see §5.4.3 for details.

In the second analysis, the number system in Ilocano is reanalyzed as having two values: *minimal* and *augmented*. The minimal values in the person/number paradigm are those where the number of discourse participants referred to is the minimum logical number for the person value, while the augmented values are those where more than the minimum number are referred to. This analysis can be modeled using the following hierarchy:

(109)

	min	aug
1st_incl	1min_incl	1aug_incl
1st	1min	1aug
2nd	2min	2aug
3rd	3min	3aug

In the third analysis (Cysouw's preferred analysis), Ilocano is again analyzed as having two number values. He refers to these values as "individual" and "group", but for consistency I will call them "singular" and "plural". He groups all the subdivided values of the first person under the first person plural, including the inclusive (1+2), the augmented inclusive (1+2+3), and the exclusive (1+3):

(110)

	sg	pl
1st	1sg	1pl_incl 1pl_augincl 1pl_excl
2nd	2sg	2pl
3rd	3sg	3pl

Depending on the facts of the language being analyzed, any of these three analyses might be preferred by the user-linguist. In particular, there may be additional agreement patterns in the language that argue one way or the other. For example, if the 1+2 form agreed on the verb with the name number value as the 1, 2, and 3 forms, that would argue for the hierarchy in (109). If, on the other hand, the 1+2 form patterned with the plural forms, the user-linguist might prefer the hierarchy in (110). The customization system must be flexible enough to allow the description of any subdivisions of the first person in order to support these various analyses.

5.3.7 Summary

In this section I have described the typology of the features gender, number, and person, and provided an HPSG analysis of each, including an explanation of the structure of the type hierarchies required to model them. Having done so, I now turn to the question of how the user-linguist can describe a target language's gender, number, and person using the customization system questionnaire.

5.4 Questionnaire

For practical reasons, the customization system questionnaire separates questions defining features from those describing agreement. In the lexicon section of the questionnaire, it should be possible to select and specify features and their values. Because it would be extremely difficult to fill out the questionnaire if the user-linguist was required to remember the names and values of all features, the questionnaire uses drop-downs to prompt with the currently available features and their values. In order to achieve this, it is necessary that the user-linguist first answer questions that determine the available features and values, and only then answer questions to which those features and values are possible answers.

In the questionnaire, then, the questions about features are split into several sections. Each feature generally has a subpage dedicated to it, in which the user-linguist specifies whether the feature is present or absent in the language being described, and what values the feature can take. After these subpages are submitted, the user-linguist then fills out the lexicon subpage, which includes sections for specifying features on stems and inflectional morphemes, and for specifying the agreement patterns between those features.

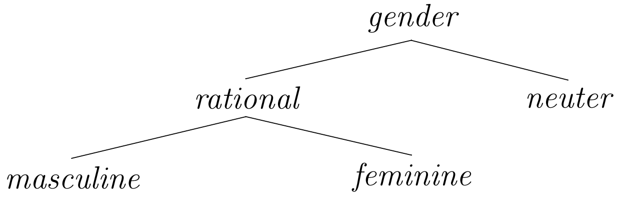
An interesting borderline case is the subpage for describing direct-inverse languages. That subpage both makes use of features described on other pages (e.g., person, gender) and allows the description of the grammatical scale on which the DIRECTION feature is based. Because it both uses and defines features, the direct-inverse subpage comes immediately before the lexicon subpage.

5.4.1 Describing Hierarchies

§5.3 above discusses the feature hierarchies that are necessary to model agreement. In order to create grammars with such hierarchies, it is necessary to prompt the user-linguist with questions that allow their description. This could potentially make the questionnaire difficult to use, since feature hierarchies can be arbitrarily complex. In the questionnaire, we use two methods for creating hierarchies: explicit and implicit.

The explicit method allows the direct construction of hierarchies. The user is asked to provide one or more values (i.e., types) in the hierarchy, and for each to provide one or more supertypes. For example, consider the gender hierarchy for Tamil, repeated here for convenience:

(111) Tamil gender:



This hierarchy can be described explicitly in the questionnaire as:

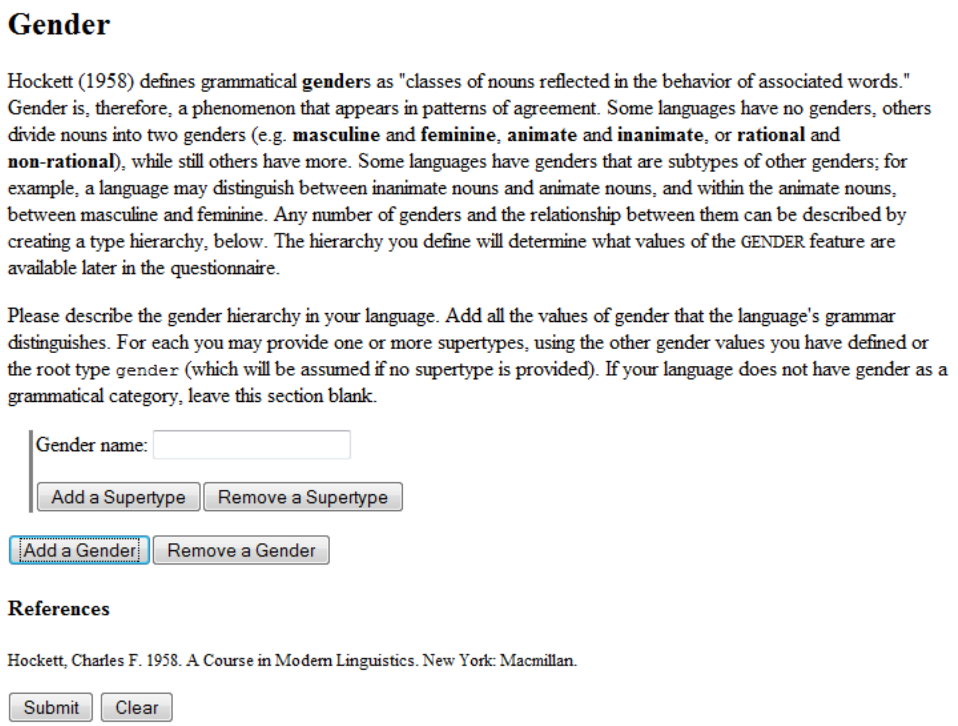
(112)

Type	**Supertype**
gender	
rational	gender
neuter	gender
masculine	rational
feminine	rational

The implicit method is to ask questions about the typological range of some phenomenon, and then to have “canned” hierarchies that correspond to the various answers. The implicit approach puts less of a burden on the user-linguist, who need only know facts about the language being described rather than design a hierarchy from scratch. However, the implicit approach is necessarily more limited—if a feature in the language being described falls outside the expected range, it cannot be described using the questionnaire (but see 5.4.2 below for a way to deal with such cases).

The questionnaire uses the explicit method for gender and number. Gender is expected to vary so widely across languages that the full flexibility of a completely customizable hierarchy is needed. Hierarchies for number are expected to be less divergent than those for gender, but enough variation exists that the implicit method would still be too limiting. See Figures 5.1 and 5.2 for the gender and number sections of the questionnaire.

In contrast, the subpage for person in the questionnaire uses the implicit method. Person is more consistent than either gender or number cross-linguistically, with most of the complexity occurring in the first person plural. Consequently, it is possible to capture the range of variation using the person section shown in Figure 5.3. The user-linguist is first asked what distinctions in person are made in the target language. Next, he or she may choose a “pre-fab” inclusive-exclusive analysis, which produces a hierarchy like that in (106) (suitably modified to reflect the number hierarchy described on the number subpage). If, on the other hand, the target language



Gender

Hockett (1958) defines grammatical **genders** as "classes of nouns reflected in the behavior of associated words." Gender is, therefore, a phenomenon that appears in patterns of agreement. Some languages have no genders, others divide nouns into two genders (e.g. **masculine** and **feminine**, **animate** and **inanimate**, or **rational** and **non-rational**), while still others have more. Some languages have genders that are subtypes of other genders; for example, a language may distinguish between inanimate nouns and animate nouns, and within the animate nouns, between masculine and feminine. Any number of genders and the relationship between them can be described by creating a type hierarchy, below. The hierarchy you define will determine what values of the GENDER feature are available later in the questionnaire.

Please describe the gender hierarchy in your language. Add all the values of gender that the language's grammar distinguishes. For each you may provide one or more supertypes, using the other gender values you have defined or the root type `gender` (which will be assumed if no supertype is provided). If your language does not have gender as a grammatical category, leave this section blank.

Gender name:

References

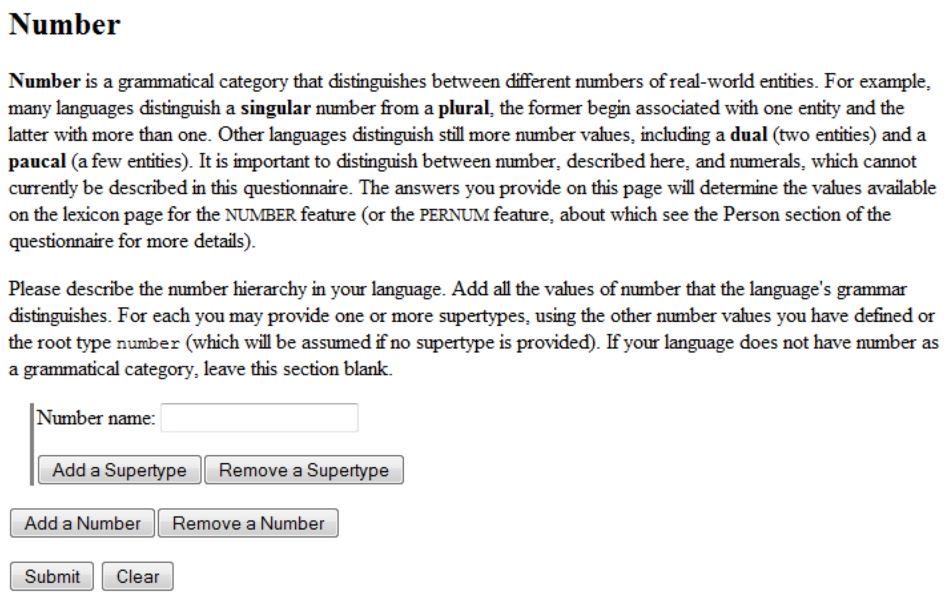
Hockett, Charles F. 1958. A Course in Modern Linguistics. New York: Macmillan.

Figure 5.1: The gender section of the questionnaire

has a minimal/augmented system or some other rare subdivision of the first person, the user-linguist may choose to define those subdivisions explicitly. The questionnaire prompts the description of one or more subtypes, each of which can occur in any of the number values of the target language. The questionnaire in Figure 5.3 is filled out with the appropriate answers to describe a minimal/augmented hierarchy like that in (110).

5.4.2 Other Features

In addition to the features anticipated in the various sections of the questionnaire, the grammar of the target language may require the definition of additional features and their associated hierarchies of values. For example, it might be convenient for a grammar of English to have a feature marking the distinction between proper names,



Number

Number is a grammatical category that distinguishes between different numbers of real-world entities. For example, many languages distinguish a **singular** number from a **plural**, the former begin associated with one entity and the latter with more than one. Other languages distinguish still more number values, including a **dual** (two entities) and a **paucal** (a few entities). It is important to distinguish between number, described here, and numerals, which cannot currently be described in this questionnaire. The answers you provide on this page will determine the values available on the lexicon page for the NUMBER feature (or the PERNUM feature, about which see the Person section of the questionnaire for more details).

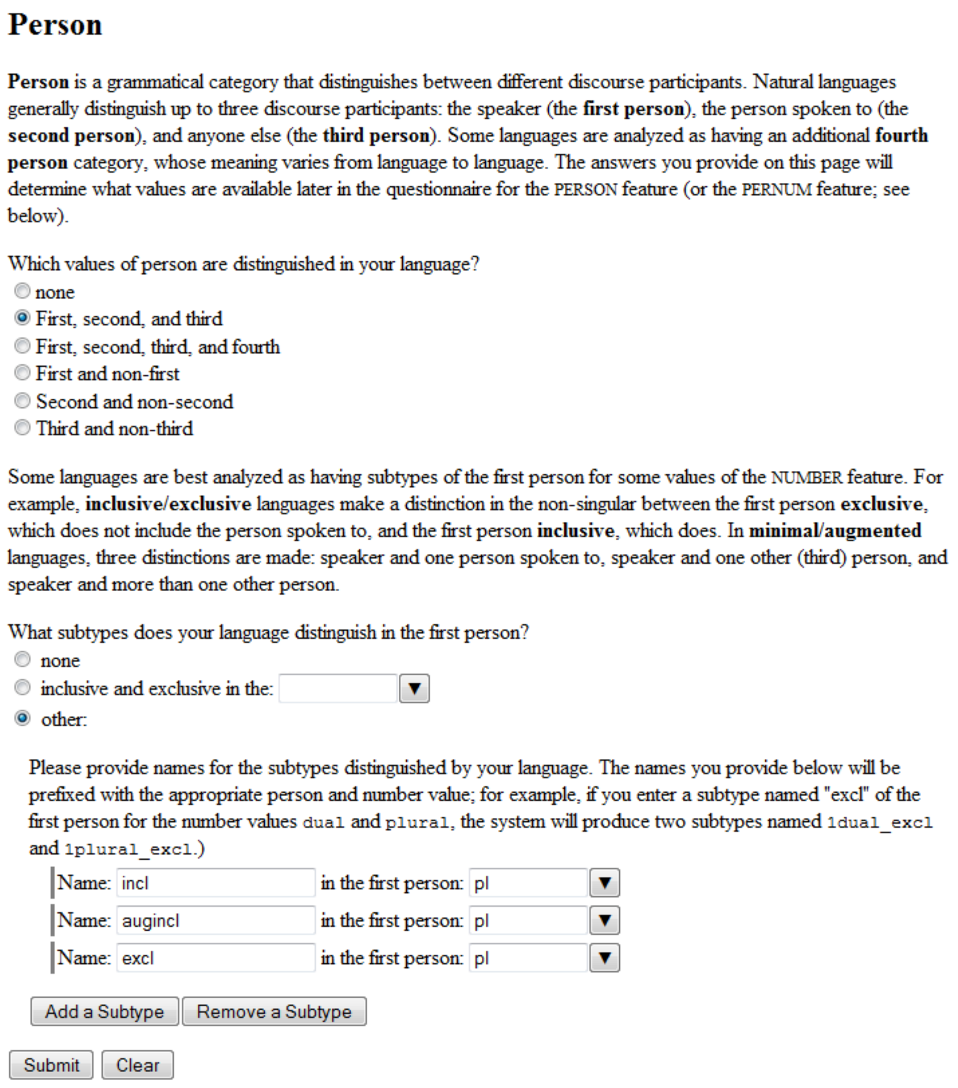
Please describe the number hierarchy in your language. Add all the values of number that the language's grammar distinguishes. For each you may provide one or more supertypes, using the other number values you have defined or the root type `number` (which will be assumed if no supertype is provided). If your language does not have number as a grammatical category, leave this section blank.

Figure 5.2: The number section of the questionnaire

pronouns, and common nouns, but there is no section of the questionnaire specifically designed to allow this. To support the description of such additional features, the questionnaire contains a subpage titled "Other Features" in which the user-linguist can provide the name, values, and hierarchy for any number of additional features. Furthermore, the user-linguist is allowed to choose whether the feature is a semantic feature that appears on the INDEX and consequently is reflected in the MRS representation of the sentence, or syntactic feature that appears on the HEAD but not in the semantics.

It is this mechanism that allows for modeling the kind of hybrid agreement seen in the French example (82) using both the system's default semantic NUMBER feature and an additional syntactic NUMBER feature. It can also be used to define arbitrary merged feature hierarchies such as the PERNUM hierarchy shown above in (105).

The subpage for defining other features is shown in Figure 5.4.



Person

Person is a grammatical category that distinguishes between different discourse participants. Natural languages generally distinguish up to three discourse participants: the speaker (the **first person**), the person spoken to (the **second person**), and anyone else (the **third person**). Some languages are analyzed as having an additional **fourth person** category, whose meaning varies from language to language. The answers you provide on this page will determine what values are available later in the questionnaire for the PERSON feature (or the PERNUM feature; see below).

Which values of person are distinguished in your language?

○ none
● First, second, and third
○ First, second, third, and fourth
○ First and non-first
○ Second and non-second
○ Third and non-third

Some languages are best analyzed as having subtypes of the first person for some values of the NUMBER feature. For example, **inclusive/exclusive** languages make a distinction in the non-singular between the first person **exclusive**, which does not include the person spoken to, and the first person **inclusive**, which does. In **minimal/augmented** languages, three distinctions are made: speaker and one person spoken to, speaker and one other (third) person, and speaker and more than one other person.

What subtypes does your language distinguish in the first person?

○ none
○ inclusive and exclusive in the: ▼
● other:

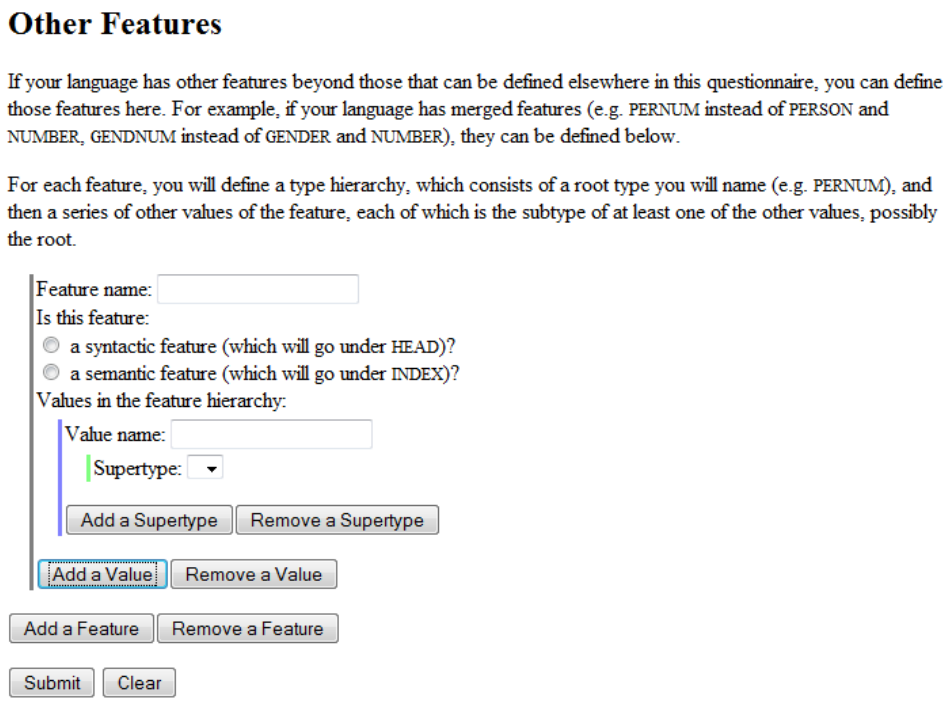
Please provide names for the subtypes distinguished by your language. The names you provide below will be prefixed with the appropriate person and number value; for example, if you enter a subtype named "excl" of the first person for the number values `dual` and `plural`, the system will produce two subtypes named `1dual_excl` and `1plural_excl`.)

Name:	incl	in the first person:	pl ▼
Name:	augincl	in the first person:	pl ▼
Name:	excl	in the first person:	pl ▼

Figure 5.3: The person section of the questionnaire

5.4.3 Hierarchy Augmentation

In addition to these methods for defining hierarchies, the customization system also performs **hierarchy augmentation** in response to patterns in the user-linguist's answers. Consider a hypothetical language in which there are pronouns for the first, second, and third person, but which has three classes of verbs distinguished by inflec-



Other Features

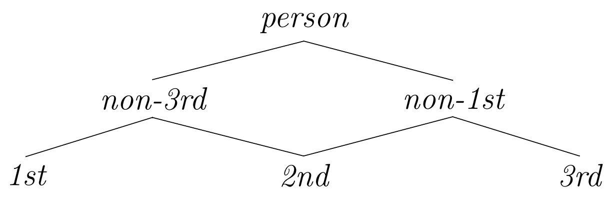
If your language has other features beyond those that can be defined elsewhere in this questionnaire, you can define those features here. For example, if your language has merged features (e.g. PERNUM instead of PERSON and NUMBER, GENDNUM instead of GENDER and NUMBER), they can be defined below.

For each feature, you will define a type hierarchy, which consists of a root type you will name (e.g. PERNUM), and then a series of other values of the feature, each of which is the subtype of at least one of the other values, possibly the root.

Figure 5.4: The “other features” section of the questionnaire

tion for person agreement. The inflection on the first verb class distinguishes first, second, and third person. The inflection on the second verb class distinguishes first person from second and third. The inflection on the third verb class distinguishes first and second person from third. It would of course be possible to analyze the second and third verb classes as having homophonous affixes that mark different values of the person feature, but such an analysis is undesirable, since it produces extra spurious edges during parsing. Instead, such a language could be modeled efficiently using the following hierarchy:

(113)



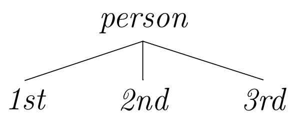
The goal of hierarchy augmentation is to produce just such a hierarchy without unduly burdening the user with the details of implementation. To describe our hypothetical language in the questionnaire, the user first states that the language distinguishes three values of the person feature. On the lexicon page, when describing the inflectional morphemes on each verb class, the user selects "person" in the drop-down for the feature name. The form control presented for the feature value is a multiple-select drop-down, allowing the user to select more than one value for the person feature. This produces entries in the choices file where feature values have comma-separated lists of values, as shown in Table 5.4.

Table 5.4: Portion of a choices file showing multiple-select values

```
verb-slot1_morph1_feat1_name=person
verb-slot1_morph1_feat1_value=1st, 2nd
verb-slot1_morph2_feat1_name=person
verb-slot1_morph2_feat1_value=2nd, 3rd
```

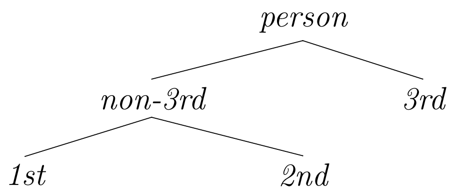
When interpreting such choices, the customization system adds additional types to the person hierarchy that correspond to the choices the user has made. Recall that, as described above in §2.1.1, adding such types to a hierarchy allows the modeling of disjunctions. The customization system, therefore, begins with the user-defined person hierarchy:

(114)



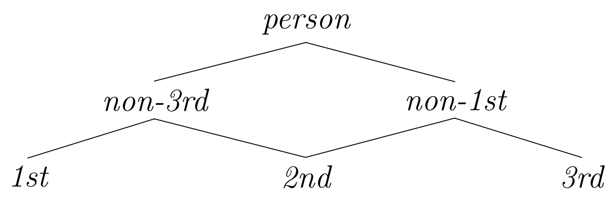
When it encounters the value `1st, 2nd`, it adds another type to the hierarchy:

(115)



When it encounters the value `2nd, 3rd`, it adds another:

(116)

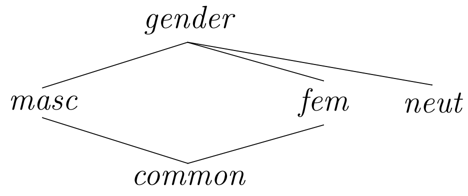


This result is, as desired, exactly the hierarchy in (113). The algorithm for augmenting a hierarchy is shown in Figure 5.5.

The name of a new type n created by augmentation is calculated as follows: if the set of types covered by n includes all but one of the leaves l, the name of the new type is the string "non-" prepended to the name of l. This produces the names "non-3rd" and "non-1st" above. Otherwise, the name of the new type is the name of each of its new subtypes, separated by "+".

There are cases where this algorithm breaks down. Suppose the user-linguist had specified the following *gender* hierarchy:

(117)



In this hierarchy, the types *masc* and *fem* cannot be distinguished by considering the sets of leaf nodes they cover, since they both cover only the leaf type *common*. When the customization system detects a hierarchy like this, it short-circuits the algorithm above, and instead simply creates a new subtype inheriting only from the root. For example, suppose the user requested a type covering *common* and *neuter* in the hierarchy in (117). The system would simply insert a new type called *common+neuter*, a subtype of *gender* and a supertype of *common* and *neuter*. It should be noted that a hierarchy like (117) is almost certainly an error on the part of the user-linguist—the

```
Given an input hierarchy $H$ and a set $S$ of types in that hierarchy (which
corresponds to a comma-separated list of feature values in the choices file):

Calculate the set $L$ of leaf types in $H$
Calculate the set $N$ of non-leaf types in $H$

for each $t \in N$ do
    for each $l \in L$ do
        if $l$ inherits from $t$ in $H$
            insert $l$ into $coverage[t]$
for each $t \in N$ do
    remove $t$ from $S$
    insert $coverage[t]$ into $S$
if there exists a $t \in S$ where $coverage[t] = S$
    return $t$
for each $t \in S$ do
    if $coverage[t] \supseteq S$
        store $t$ in $supers[]$
for each $t \in S$ do
    if $coverage[t] \subseteq S$
        store $t$ in $subs[]$
for each $t, t' \in supers$ do
    if $supers[t] \supseteq supers[t']$
        remove $t$ from $supers$
for each $t, t' \in subs$ do
    if $subs[t] \subseteq subs[t']$
        remove $t$ from $subs$
for each $t \in supers, t' \in subs$ do
    if $t'$ inherits from $t$ in $H$
        remove the inheritance

Finally, insert a new type $n$ into $H$ that inherits from all $t \in supers$, and
from which all $t' \in subs$ inherit
```

Figure 5.5: Algorithm for hierarchy augmentation

types *masc* and *fem* will unify because they have a common subtype—but the system handles them nonetheless.

Augmentation works just as well with the explicit or the implicit methods for hierarchy description. In either case, the input to the augmentation process is a hierarchy as defined by the user-linguist, and the output is a type hierarchy that maintains all the distinctions in the input while compactly representing generalizations about the target language inferred from the answers on the lexicon subpage.

Depending on the agreement pattern in the target language, hierarchies can become extremely complex after augmentation. For an example of such a hierarchy, see the PERNUM hierarchy produced by the customization system for Sahaptin in §6.2.3. Since the customization system performs augmentation automatically, based only on the value entered into multi-select drop-downs, the user-linguist is saved the trouble of having to design and construct such the hierarchies by hand.

5.5 Test Cases

The operation of the customization system's support for agreement can be seen in several of the test case grammars in other chapters. In particular, see the test cases for German in §3.4.2.1, which includes agreement between determiners and nouns, and for Cree in §4.4.1, which includes agreement between verbs and subjects and objects. In addition, the following chapter contains an extensive case study of Sahaptin, a language with extremely complex argument marking and agreement patterns.

5.6 Summary

In this chapter, I have described my analysis of person, number, gender, and agreement, and discussed the theoretical and typological foundations of that analysis. Through the questionnaire, the user-linguist can produce grammars based on my analysis that are capable of modeling a wide variety of agreement patterns. Furthermore, I have introduced the concept of hierarchy augmentation, by which the customization system creates type hierarchies suited to the specific facts of agreement in the target language, even very complex ones, without the user-linguist's intervention.

Chapter 6
CASE STUDY: SAHAPTIN

Having introduced and described my improvements to the customization system supporting case, direct-inverse languages, and agreement in several different features, I will now demonstrate all of these libraries working together in a single language. This chapter consists of an extended case study of Umatilla Sahaptin [uma] (Penutian), a language with extremely complex argument marking and agreement patterns. I will show that a substantial fragment of Sahaptin covering these patterns can be straightforwardly described using the customization system questionnaire, thus producing a grammar in software that correctly models a significant fragment of Sahaptin morphosyntax.

It is common in computational linguistics, particularly when building statistical language models that are trained on a corpus of natural language, to divide the data into three parts: a training set, a development set, and a test set. The training set is used to train the model. During development, the model's performance is repeatedly evaluated by running it across the development set. Finally, when development is finished, the true performance of the model is evaluated on the test set, which it has never seen before. My work on the Matrix customization system has followed an analogous path. The libraries for case, direct-inverse, and agreement were designed based on the typological literature, then repeatedly tested during development on the test cases described in Chapters 3 and 4. After development was finished, I turned to Sahaptin, a language I knew had very complex agreement patterns, but for which the customization system had not been specifically designed, making Sahaptin analogous to the test set.

In this chapter, the facts of Sahaptin are drawn from a description of the language by Rigsby and Rude (1996) (henceforth R&R). I will focus on the Umatilla dialect of Sahaptin, since that is the dialect described in the most detail by R&R. The chapter will consist of three sections: a brief sketch of the Sahaptin language, a description of how the questionnaire can be filled out to model it, and a demonstration that the resulting grammar produces correct analyses of more than 6000 test sentences.

6.1 A Brief Sketch of Sahaptin

Sahaptin is a convenient language for this case study because it illustrates a number of the phenomena I have added to the customization system, including:

- Case marking on verbal arguments.

- Argument marking sensitive to a grammatical scale, including what I analyze below as proximate and obviative marking on third-person nominals.

- Two loci of agreement (a verbal prefix and a second-position enclitic) with both the subject and the object.

- A distinction in number between singular, dual, and plural on nominals, but only between singular and plural on agreement morphology.

- An inclusive/exclusive distinction in person reflected only in the second-position enclitic.

Below, I will summarize the description of these phenomena by Rigsby and Rude (1996), including detailed paradigms of intransitive clauses, transitive clauses, and pronouns.

To get a sense of the structure of Sahaptin sentences, let us begin by considering the following simple example:

(118) *ín=aš á-tux̣nana yáamaš-na*
I=1SG 3ABS-shot mule.deer-OBJ
‘I shot the mule deer.’ [uma] (Rigsby and Rude 1996:676)

In (118) the first word consists of the first person singular pronoun in its unmarked form, the nominative, followed by a second-position enclitic that agrees with the pronoun. The second word is the verb, consisting of a verbal prefix appropriate to the person and number of the subject and object (glossed by R&R as 3ABS, but see §6.2 below for a different analysis) and the verb stem. The third word consists of the noun stem meaning ‘mule deer’ and a suffix marking the objective case.

Although (118) is in SVO order, Sahaptin word order is extremely flexible. R&R write, “All sentences may have their constituent words scrambled, provided that the enclitic is in sentence-second position.” (Rigsby and Rude 1996:677) Furthermore, R&R describe three types of sentences, which they call types A, B, and C, that are distinguished by their specificity of marking. Type A sentences are fully marked with case, verbal prefixes, enclitics, and so on. Type B and C sentences are less marked, and therefore more ambiguous, but they occur commonly in discourse. Since R&R focus their detailed description almost exclusively on sentences of type A, I consequently restrict my analysis to sentences of that type.

6.1.1 Case Marking

R&R describe several core cases in Sahaptin (in addition to a number of oblique cases that the customization system cannot yet handle). These include an unmarked “nominative” case, a marked “objective” case, an “inverse ergative” case, and an “obviative ergative” case. In spite of their use of the term “ergative”, R&R make it clear that the subject consistently appears in the nominative case in both transitive and intransitive clauses, and that the object consistently appears in the objective case in transitive clauses. The “inverse ergative” and “obviative ergative” forms are only available for third person singular nominals, both nouns and pronouns, in addition to the subject and object forms, and they are used to distinguish the subject from the object in transitive clauses.[1]

6.1.2 Agreement and Argument Marking

In addition to case marking on nominals, Sahaptin has two ways to cross-reference the arguments of verbs: a verbal prefix and a second-position enclitic that attaches to whichever word comes first in the sentence. R&R characterize the prefixes and enclitics in two ways: first, they provide a general description of the distribution of each; second, they provide detailed paradigms of intransitive and transitive sentence patterns that cover most, but not all, of the logical combinations.

R&R describe Sahaptin's second-position enclitics as shown in Table 6.1.

Table 6.1: Sahaptin enclitics (Rigsby and Rude 1996:675)

Enclitic	**Description**
=naš ∼ *=aš* ∼ *=š*	"first-person singular"
=na	"first-person plural inclusive"
=nataš ∼ *=ataš* ∼ *=taš*	"first-person plural exclusive"
=nam ∼ *=am* ∼ *=m*	"second-person singular"
=pam	"second-person plural"
=maš	"second-person object with first-person subject (both singular)"
=mataš	"second-person object with first-person subject (one or both plural)"

Notice in particular that several of the enclitics are associated with a person and number, but R&R do not mention whether those values are associated with the subject

[1] Rude (personal communication to Hargus, 6/1/09) writes that Sahaptin "has an ergative noun case (-*nɨ́m*) which marks a kind of semantic inverse," but that Sahaptin lacks "ergative-absolutive alignment," which disqualifies it as an ergative or split-ergative language. Rude (1996) now refers to -*in* as "associative" rather than "obviative ergative" (because -*in* is also used in NP coordination) and to -*nɨ́m* as "ergative" rather than "inverse ergative".

or the object. The reason for this becomes clear when we examine the full paradigm of clauses in Tables 6.3 and 6.4. The enclitic *=nataš*, for example, occurs with first person plural exclusive subjects in intransitive clauses; in transitive clauses, however, it occurs when one argument is first person plural exclusive and the other is third person, regardless of which is the subject and which is the object. A similar pattern can be observed for *=na* and *=naš*. This variant of scale-sensitive argument marking is the pattern that motivated the enhancement to the customization system described in §4.2.1.

As for Sahaptin's verbal prefixes, R&R describe them as shown in Table 6.2.[2]

Table 6.2: Sahaptin verbal prefixes (Rigsby and Rude 1996:675)

Prefix	**Description**
i-	"third-person nominative"
pa-	"third-person plural nominative"
á- ∼ *áw-*	"third-person absolutive"
pá-	"inverse"
patá- ∼ *patáw-*	"third-person plural subject with third-person object"

These descriptions are less straightforward than those for the enclitics. In particular, the description of *á-* ∼ *áw-* as "absolutive" is misleading. Regarding that prefix, R&R write, "...this pronominal marks subjects in intransitive clauses when they are possessors, and objects in transitive clauses when the subject is first or second person." (675) In other words, it is not associated with all transitive clauses, and only with intransitive clauses where the subject is possessive. Furthermore, all the prefixes above appear on the verb, not the nominal arguments, as one might expect for an "absolutive" affix. I do not believe, therefore, that the prefix *á-* ∼ *áw-* is evidence of ergative alignment in Sahaptin. Similarly, although there is evidence of argument

[2] There are three further verbal prefixes in Sahaptin that mark reflexives and reciprocals, but there is currently no support for these phenomena in the customization system, so I omit them from the current discussion.

marking sensitive to a grammatical scale, the *pá-* prefix cannot be simply categorized as inverse marking on the verb. As with the second-position enclitic, the true behavior of Sahaptin verbal prefixes is best illustrated by the full paradigms in Tables 6.3 and 6.4.[3] Note that to save space in these tables, I show only one variant of prefixes and enclitics with multiple surface forms.

The paradigm for intransitive verbs in Table 6.3 is fairly straightforward: the conditions under which each enclitic and prefix appear are simple, and as a result, so will be their analysis in the following section. The transitive paradigm in Table 6.4, on the other hand, is extremely complex, and its analysis in the next section is correspondingly much more elaborate.

Table 6.3: Sahaptin Agreement in Intransitive Clauses (Rigsby and Rude 1996:676)

Subject Nominal	**Enclitic**	**Verb Prefix**	**Case-marking on Subject Nominal**
1 sg	=*naš*	—	—
1 pl inc	=*na*	—	—
1 pl exc	=*nataš*	—	—
2 sg	=*nam*	—	—
2 pl	=*pam*	—	—
3 sg	—	*i-*	—
3 pl	—	*pa-*	—

6.2 Customizing a Sahaptin Grammar

In this section, I will show how the customization system questionnaire can be filled out in order to produce a grammar that models a fragment of Sahaptin covering the phenomena described in the previous section. However, some aspects of Sahaptin

[3] Hargus (to appear:xlvi) describes a slightly different agreement pattern that is attested, but not required, in Yakima Sahaptin: sentences with the obviative ergative have a verb marked with *pá-*, and sentences with *pá-* but without the obviative ergative may take an instrumental suffix on the subject when the agent is not human. However, for the purposes of this chapter, namely testing the customization system, I have chosen to model the agreement pattern described in R&R.

Table 6.4: Sahaptin Agreement in Transitive Clauses (Rigsby and Rude 1996:676)

Subject	**Object**		**Verb**	**Noun case-marking**	
Nominal	**Nominal**	**Enclitic**	**Prefix**	**Subject**	**Object**
1 sg	2 sg	=*maš*	—	—	obj sg
1 sg	2 pl	=*mataš*	—	—	obj pl
1 pl exc	2 sg	=*mataš*	—	—	obj sg
1 pl exc	2 pl	=*mataš*	—	—	obj pl
1 sg	3 sg	=*naš*	*á-*	—	obj sg
1 sg	3 pl	=*naš*	*á-*	—	obj pl
1 pl inc	3 sg	=*na*	*á-*	—	obj sg
1 pl inc	3 pl	=*na*	*á-*	—	obj pl
1 pl exc	3 sg	=*nataš*	*á-*	—	obj sg
1 pl exc	3 pl	=*nataš*	*á-*	—	obj pl
2 sg	1 sg	=*nam*	*pá-*	—	obj sg
2 sg	1 pl	=*nam*	—	—	obj pl
2 pl	1 sg	=*pam*	—	—	obj sg
2 pl	1 pl	=*pam*	—	—	obj pl
2 sg	3 sg	=*nam*	*á-*	—	obj sg
2 sg	3 pl	=*nam*	*á-*	—	obj pl
2 pl	3 sg	=*pam*	*á-*	—	obj sg
2 pl	3 pl	=*pam*	*á-*	—	obj pl
3 sg	1 sg	=*naš*	*i-*	inv erg	obj sg
3 sg	1 pl inc	=*na*	*i-*	inv erg	obj pl
3 sg	1 pl exc	=*nataš*	*i-*	inv erg	obj pl
3 pl	1 sg	=*naš*	*pa-*	—	obj sg
3 pl	1 pl inc	=*na*	*pa-*	—	obj pl
3 pl	1 pl exc	=*nataš*	*pa-*	—	obj pl
3 sg	2 sg	=*nam*	*i-*	inv erg	obj sg
3 sg	2 pl	=*pam*	*i-*	inv erg	obj pl
3 pl	2 sg	=*nam*	*pa-*	—	obj sg
3 pl	2 pl	=*pam*	*pa-*	—	obj pl
3 sg top	3 sg	—	*i-*	obv erg	obj sg
3 sg	3 sg top	—	*pá-*	—	obj sg
3 sg	3 pl	—	*i-*	—	obj pl
3 pl top	3 sg	—	*pa-*	—	obj sg
3 pl	3 sg top	—	*patá-*	—	obj sg
3 pl	3 pl	—	*pa-*	—	obj pl

are beyond the current capabilities of the customization system, so some simplifying assumptions are necessary.

Recall that the customization system models complex morphosyntax, but not complex morphophonology; the Sahaptin grammar will therefore use a single spelling for each stem and morpheme. The facts of Sahaptin word order are also too complex for the customization system. In particular, it cannot model truly free word order (i.e., discontinuous noun phrases), and the attachment behavior of the second-position enclitic is similarly beyond its capability.

With these simplifying assumptions, however, the customization system is capable of modeling all the complex agreement and marking patterns described in §6.1 above. In the following sections (whose structure parallels that of the questionnaire), I describe the answers required to produce a Sahaptin grammar. The full set of answers can be seen in the Sahaptin choices file, which is shown in Appendix G.

6.2.1 Word Order

In this grammar, I treat Sahaptin as a VSO language, and the enclitic as a suffix on verbs. This means that the sentences recognized and generated by the grammar are in a legal word order—VSO sentences where the verb is followed by the enclitic are grammatical in Sahaptin—but there are other word orders that the grammar will not allow. The analysis of the enclitic is therefore limited by the current capabilities of the customization system's word order library; however, if that library is enhanced in the future to support second-position clitics, the analysis presented below should transfer straightforwardly.

6.2.2 Number

I analyze Sahaptin as having three values of number: singular (sg), dual (du), and plural (pl). All three values are distinguished on pronouns, as shown in Table 6.5; however, agreement with enclitics and verbal prefixes only shows a singular/plural distinction (with dual pronouns agreeing with the plural morpheme). It will be necessary in several places for the grammar to refer to non-singular (i.e., *du* or *pl*). Rather than constructing a hierarchy that includes a *non-sg* value for number, however, I instead filled out the questionnaire for a flat hierarchy containing only *sg*, *du*, and *pl* in order to also demonstrate multi-select drop-downs and hierarchy augmentation in this example grammar.

Table 6.5: Umatilla Sahaptin Pronouns (Rigsby and Rude 1996:682–683)

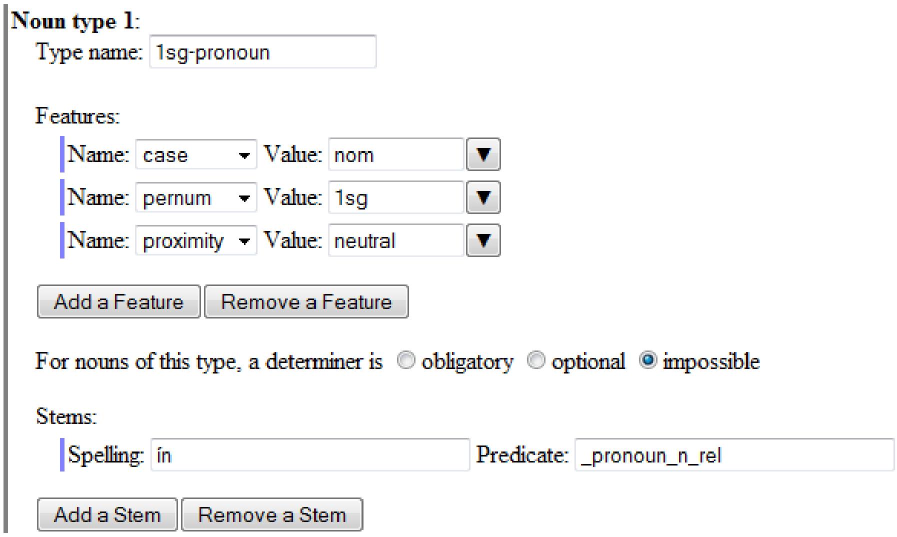
	Singular		**Dual**		**Plural**	
	Subject	**Object**	**Subject**	**Object**	**Subject**	**Object**
1	*ín*	*ináy*	*napiiní*	*napiinamanáy*	*náma*	*naamanáy*
2	*ím*	*imanáy*	*imiiní*	*imiinamanáy*	*imáy*	*imaamanáy*
3	*pɨ́n*	*paanáy*	*piiní*	*piinamanáy*	*pmáy*	*paamanáy*
3 obv erg	*piiní*					
3 inv erg	*pnɨ́m*					

Table 6.5 shows the Sahaptin pronoun forms that distinguish singular, dual, and plural. In the questionnaire, each of these pronouns specifies one of the number values mentioned above (*sg*, *du*, or *pl*), as can be seen for the first person singular nominative pronoun in Figure 6.1.[4] On the other hand, agreement morphemes that do not distinguish between the dual and plural are simply specified to have both values using a multiple-select drop-down, producing a type covering both via hierarchy augmentation.

6.2.3 Person

Sahaptin distinguishes three values of person: first, second, and third. The enclitics (but, interestingly, not the pronouns) further distinguish a first person inclusive and

[4] To see how the rest of the Sahaptin grammar was defined, visit the web questionnaire (http://www.delph-in.net/matrix/sfddiss/) and click the “Umatilla Sahaptin” link at the bottom of the main page. This will fill out the questionnaire with answers that describe the grammar developed for this case study.



Noun type 1:

Type name: 1sg-pronoun

Features:

Name: case Value: nom

Name: pernum Value: 1sg

Name: proximity Value: neutral

For nouns of this type, a determiner is ○ obligatory ○ optional ● impossible

Stems:

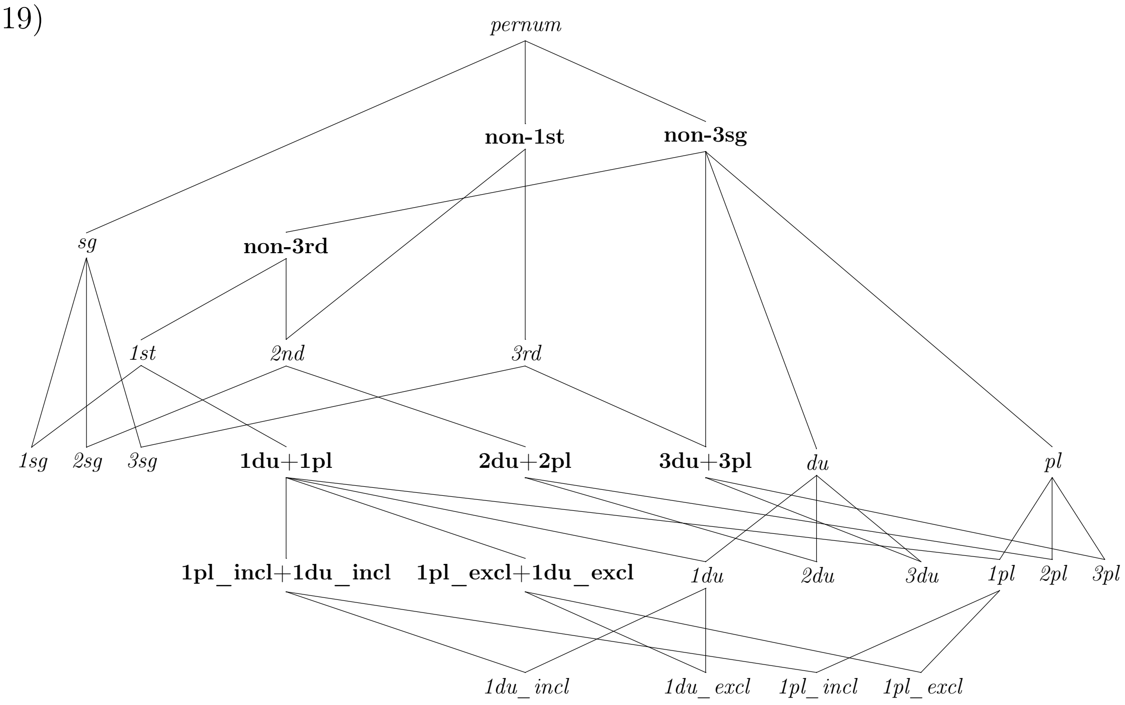
Spelling: ín Predicate: _pronoun_n_rel

Figure 6.1: A Sahaptin pronoun defined in the questionnaire

first person exclusive. I filled out the person section of the questionnaire with answers reflecting this. As mentioned in §5.3.6, this results in the production of a merged PERNUM hierarchy covering both person and number. The hierarchy produced for Sahaptin starts off the same as that shown in (106) in Chapter 5, but several new types are added by augmentation. The final PERNUM hierarchy for Sahaptin is shown in (119) below, with types inserted during augmentation shown in **boldface**:[5]

[5] See §5.3.6.1 for more about the merged PERNUM hierarchies produced by the customization system.

(119)



6.2.4 Case

As described above, Sahaptin has a nominative case that marks intransitive and transitive subjects and an objective case that marks transitive objects. This is the common nominative-accusative pattern, so in the case section of the questionnaire I describe it as such, with the label *nom* for nominative and *obj* for objective. Note that I do *not* analyze the inverse ergative and obviative ergative as case; see §6.2.6 for details.

6.2.5 Direct-Inverse

I analyze Sahaptin as a direct-inverse language—that is, a language whose argument marking is sensitive to a grammatical scale—though one that (like Fore in §4.1.2) lacks clear direct or inverse forms of the verb, with the exception of the *pá-* prefix. The scale I propose for Sahaptin is:

(120) 1st person > 2nd person > 3rd person topic > 3rd person non-topic

Sahaptin appears to treat sentences in which the subject and object are equal according to the scale as direct rather than inverse, so I have chosen the corresponding choice in the questionnaire. See the Lexicon section below to see how this scale is used to model Sahaptin grammar.

6.2.6 Other Features

I use two additional features in my analysis of Sahaptin: a semantic TOPICALITY feature and a syntactic PROXIMITY feature, both on nominals.

In Table 6.4, it can be seen that Sahaptin marks transitive clauses involving third person arguments differently depending on which argument is the topic. Topicality is a discourse feature, and so would require a semantic representation that could model intersentential information; however, the MRS representation used by grammars produced by the customization system only models the semantics of single sentences. Rather than simply neutralizing the distinction between topic and non-topic noun phrases, which would slightly blunt the point of this demonstration of the flexibility of the customization system, I have chosen to explicitly mark third person nominals as topical or not, and to record this marking using a *topicality* feature on nominal indices. Since this is a semantic feature, it appears in the MRS representation of sentences parsed by the grammar, so it could in principle be hooked up to some hypothetical component that models intersentential semantics.[6]

I use the syntactic PROXIMITY feature to model the "inverse ergative" and "obviative ergative" forms of nominals. An examination of Table 6.4 in light of the argument-marking scale defined in (120) shows that the inverse ergative occurs precisely when the subject is third person singular and the clause is inverse (that is, the object is higher on the scale). The obviative ergative occurs in exactly one case: when the subject is third person singular and the object is a topical third person singular.

[6] See also Bender and Goss-Grubbs (2008) and Borthen and Haugereid (2005) for other examples of using semantic features to store discourse information in the MRS representation.

These “ergative” forms function very much like the proximate and obviative forms in Algonquian languages (see §4.2.2). However, unlike those languages, I analyze Sahaptin as having three values of the PROXIMITY feature: *proximate*, corresponding to the inverse ergative *-nɨ́m*, which promotes the marked nominal up the scale; *obviative*, corresponding to the obviative ergative *-in*, which demotes the marked nominal down the scale; and *neutral*, the unmarked form, which does not affect the nominal’s position on the scale.[7]

6.2.7 Lexicon

Now that we have all the necessary features and their values defined, it is finally possible to describe the lexicon of the Sahaptin grammar fragment, which consists of both lexical types and inflectional morphemes attaching to those types. As detailed in §2.4.4, inflectional morphology is described as a series of slots, each of which attaches to one or more lexical types or other slots, and each of which contains one or more morphemes, each of which can specify features. As will be seen below, each set of morphemes should have features specified so that no morpheme overlaps another, but also so that no legal combination of features goes unexpressed. In general, the simplest grammars are those that do not resort to homophony—that is, to having multiple morphemes that have the same spelling but specify different features. It is often possible to avoid homophony by adding complexity to feature hierarchies,[8] but overly complex hierarchies can be as difficult to manage as extensive homophony. In the Sahaptin grammar, I have attempted to strike a balance between homophony and hierarchy complexity.

To include nominals displaying the full range of PERNUM possibilities, I define lexical types for each of the pronouns in Table 6.5. Each type specifies the pronoun

[7] Note that, for consistency with R&R’s description, I will nonetheless continue to refer to the marked forms as the “inverse ergative” and “obviative ergative”.

[8] This is the principle that gives us a merged PERNUM hierarchy, for example.

relation (pronoun_n_rel) and the appropriate PERNUM value.[9]

In addition to the lexical types, I have defined a number of inflectional slots in order to model Sahaptin morphosyntax. Note that, as in Tables 6.3 and 6.4 above, I have included only one variant of prefixes and enclitics that have multiple surface forms—such forms could have been described in the questionnaire using morphemes that were identical except for their spellings, but doing so would not have probed the capabilities of the system any more deeply. Note also that in the following tables describing inflectional slots, null morphemes (i.e., those that have no phonological content) are represented by the ∅ symbol, while features that are unspecified are represented by empty cells.

Recall, as described in §6.2.6 above, that I have chosen to explicitly mark topicality on nominals in the grammar fragment, although Sahaptin does not do so. This is accomplished by defining an inflectional slot that attaches to noun types. Two morphemes can appear in this slot, each of which specifies a value for TOPICALITY and one of which specifies a value for PERNUM, as shown in Table 6.6.

Table 6.6: Morphemes appearing in the topic slot

Topic marker	TOPICALITY	PERNUM
-TOP	*topic*	*3rd*
∅	*non-topic*	

The morphemes in this slot allow third person nominals to be marked either as the topic or not; however, first and second person nominals are incompatible with the value of the PERNUM feature on the *-TOP* suffix, and so are never marked topical in the grammar fragment.

For verbs, I define two lexical types, one for intransitive verbs, and one for tran-

[9] The output lexicon would be simpler if it were possible to define a single supertype for all pronouns specifying the pronoun relation, but the customization system does not yet allow the definition of arbitrary super- and subtype relationships in the lexicon.

sitive verbs, each with a single stem. The intransitive verb type specifies that its subject is nominative, and its stem is *wína* ‘go’ and its predicate is `_wína_v_rel`. The transitive verb type specifies that its subject is nominative, its object is objective, and the verb is constrained by the direct-inverse scale. The single transitive stem is *q̓ínun* ‘see’ and its predicate is `_q̓ínun_v_rel`.

I define five inflectional slots on verbs, two of which attach to intransitive stems and three to transitive stems. The first intransitive slot is for verbal prefixes and can contain one of three morphemes, each of which specifies PERNUM on the subject NP, and one of which, *i-*, also specifies PROXIMITY in order to prevent inverse and obviative ergatives from appearing in intransitive clauses with third person subjects. The morphemes that appear in this slot are shown in Table 6.7.

Table 6.7: Morphemes appearing in the intransitive prefix slot

Intransitive prefix	**Subject** PERNUM	**Subject** PROXIMITY
i-	*3sg*	*neutral*
pa-	*3du, 3pl*	
∅	*1st, 2nd*	

The second intransitive slot is for enclitics. It can contain one of six morphemes, each of which specifies PERNUM on the subject NP as shown in Table 6.8.

Table 6.8: Morphemes appearing in the intransitive enclitic slot

Intransitive enclitic	**Subject** PERNUM
=naš	*1sg*
=na	*1du_ incl, 1pl_ incl*
=nataš	*1du_ excl, 1pl_ excl*
=nam	*2sg*
=pam	*2du, 2pl*
∅	*3rd*

The slots attaching to transitive verbs are rather more involved. The first of these contains the transitive variants of the verbal prefixes. Each specifies PERNUM and optionally TOPICALITY on the subject NP, and optionally specifies PERNUM and TOPICALITY on the object, as shown in Table 6.9.

Table 6.9: Morphemes appearing in the transitive prefix slot

Transitive prefix	**Subject** PERNUM	**Subject** TOPICALITY	**Object** PERNUM	**Object** TOPICALITY
i-	*3sg*			*non-topic*
pa-	*3du, 3pl*			*non-topic*
á-	*1st, 2nd*		*3rd*	
pá-	*2sg*		*1sg*	
pá-	*3sg*	*non-topic*	*3sg*	*topic*
patá-	*3du, 3pl*	*non-topic*	*3sg*	*topic*
∅	*1st*		*2nd*	
∅	*2du, 2pl*		*1st*	
∅	*2sg*		*1du, 1pl*	

There are two instances of homophony in the transitive prefixes. The first involves the two variants of the *pá-* prefix. This distinction is quite well motivated—the two variants share none of the same PERNUM or TOPICALITY constraints on the subject or the object. The homophony of the three variants of the null morpheme, on the other hand, is less desirable, but necessary. If it were not for the variant of the *pá-* prefix that appears with second person subjects and first person objects, the null morpheme could simply be modeled with a single variant having [PERNUM *1st, 2nd*] on both subject and object. However, the existence of that *pá-* variant necessitates the definition of several variants of the null morpheme in such a way that they cover the rest of the range covered by *pá-* without overlapping. The first variant covers all clauses with first person subjects and second person objects. The second covers clauses with a second person non-singular subject and a first person object. That leaves just one combination, clauses with second person singular subject and a first person non-singular object, which is covered by the third variant.

The second transitive slot contains one of eleven enclitics, each of which specifies the PERNUM of its subject NP and, optionally, the PERNUM of its object. However, three of the morphemes do not specify PERNUM directly on the subject or object; instead, they use the mechanism available in the questionnaire for direct-inverse languages to specify features on NP arguments based on their relative position on the grammatical scale (see §4.2.1 for details). In fact, it was this case study of Sahaptin that made it clear such a mechanism was desirable. In an earlier version of the Sahaptin grammar (not included here), each of these three morphemes was duplicated, with the two variants having opposite features specified on the verb's arguments. For example, one variant of *=naš* specified PERNUM to be *1sg* on the subject and *3rd* on the object, while the other specified *3rd* on the subject and *1sg* on the object. It was clear that the system was missing a generalization about Sahaptin, namely that some enclitics had agreement that was sensitive to relative scale position, so I enhanced the system to allow such agreement to be modeled in a single lexical rule.

The three scale-sensitive morphemes are shown in Table 6.10, while the other transitive enclitics that specify subject and object PERNUM directly are shown in Table 6.11.

Table 6.10: Morphemes appearing in the transitive enclitic slot

Transitive enclitic	**Higher-ranked** NP PERNUM	**Lower-ranked** NP PERNUM
=naš	*1sg*	*3rd*
=na	*1du_incl, 1pl_incl*	*3rd*
=nataš	*1du_excl, 1pl_excl*	*3rd*

There are three instances of homophony here. The distributions of *=nam* and *=pam* are nearly, but not quite, the sort that could be analyzed with a single scale-sensitive morpheme. *=nam*, for example, occurs with second person singular subjects

Table 6.11: Morphemes appearing in the transitive enclitic slot

Transitive	**Subject**	**Object**
enclitic	PERNUM	PERNUM
=nam	*2sg*	
=nam	*3rd*	*2sg*
=pam	*2du, 2pl*	
=pam	*3rd*	*2du, 2pl*
=maš	*1sg*	*2sg*
=mataš	*1st*	*2du, 2pl*
=mataš	*1du, 1pl*	*2sg*
∅	*3rd*	*3rd*

and third person objects, and with third person subjects and second person singular objects. However, both *=nam* and *=pam* also appear with second person subjects and first person objects, but not the reverse. The two homophonous variants of *=nam* and of *=pam* in Table 6.11 account for all these patterns. The morpheme *=mataš* has two variants for a different reason. That morpheme appears when one argument or the other, but not both, is non-singular; put another way, *=mataš* appears with a first person subject and second person object **unless** *=maš* appears. The version of HPSG used here, however, cannot express such a conditional constraint. Instead, I define one variant of *=mataš* that covers any first person subject (singular or non-singular) and a second person non-singular object. The other variant covers a first person non-singular subject and a second person singular object. This licenses three of the four possible combinations of a first person subject and second person object, leaving the fourth (first person singular subject, second person singular object) to be licensed by *=maš*.

The final inflectional slot on verbs controls the appearance of the inverse ergative (i.e., proximate) and obviative ergative (i.e., obviative) forms. The purpose of this slot is to create a sheaf of lexical rules that correctly constrain the marking of the nominal arguments, rather like the sheaf of rules used to model direct-inverse verbs in (69) in

Chapter 4. Recall that the customization system questionnaire creates a lexical rule for each morpheme described in the questionnaire; I therefore create the sheaf of rules for constraining proximity by describing several "morphemes" that appear in the slot, all of which are phonologically null. These "morphemes" are shown in Table 6.12.

Table 6.12: Null "morphemes" appearing in the proximity slot

Subject	**Subject**	**Subject**	**Object**	**Object**	**Verb**
PROXIMITY	PERNUM	TOPICALITY	PERNUM	TOPICALITY	DIRECTION
proximate	*3sg*			*non-topic*	*inverse*
obviative	*3sg*	*topic*	*3sg*		
neutral	*3sg*		*3sg*	*topic*	
neutral	*3sg*		*3du, 3pl*		
neutral	*1st, 2nd,* *3du, 3pl*				

Each of these morphemes serves a purpose, licensing a particular set of transitive clause types. The first covers all clause types where inverse ergative marking appears on the subject.[10] The second morpheme picks out the single clause type where obviative ergative marking appears. The third and fourth morphemes together cover the remaining clause types with third person singular subjects, in which neither inverse nor obviative ergative appear. Finally, the fifth morpheme covers all other clause types.

It is worth noting that, in addition to some redundancy (homophony) appearing within inflectional slots, I have chosen to separate the transitive and intransitive forms of prefixes and enclitics. In some cases, this was clearly necessary; for example, there is no intransitive version of *patá-*. However, a few prefixes or enclitics appear to specify similar features in transitive and intransitive clauses; *=naš*, for example,

[10] Notice that this is the only place in the grammar where the DIRECTION feature is used explicitly, and also that it is only specified on one morpheme. It turns out that apart from this single morpheme, specifying PROXIMITY, TOPICALITY, and PERNUM on the subject and object is sufficient to model the distribution of proximity marking on Sahaptin nominals.

always specifies either the subject or object to be first person singular. Despite these similarities, the transitive and intransitive variants cannot straightforwardly be defined using the same morpheme (i.e., the same lexical rule). The transitive variants specify features on the object NP, which would make them incompatible with intransitive verbs. The only transitive morphemes that do not specify features on the object are one variant of *=nam* and *=pam*, respectively. By taking advantage of this, it would have been possible to describe these morphemes another way, using three inflectional slots for enclitics: the existing slot that attaches only to intransitives, the existing slot that attaches only to transitives, and a new slot that attaches to either, with only *=nam* and *=pam* appearing in the third slot. However, to keep the organization of the grammar simple, I have chosen to segregate transitive and intransitive inflection into separate slots.

6.3 Testing the Sahaptin Grammar

In order to test the correctness of the Sahaptin grammar, it was necessary to create a suite of test sentences, some grammatical and some not, that are within its expected lexical and grammatical coverage. I started with the sentence patterns in Tables 6.3 and 6.4; from each, I created a sentence with the appropriate prefix, verb, enclitic, subject, and object. In every case where a plural argument was called for, I also created a second sentence with dual argument—and in patterns with two plural arguments, I created four: *du/du*, *du/pl*, *pl/du*, and *pl/pl*.

All these sentences were expected to be grammatical. To generate ungrammatical sentences, I permuted the grammatical sentences in the following ways:

1. For each grammatical sentence with a prefix, I created an ungrammatical variant with the prefix missing.

2. For each grammatical sentence with an enclitic, I created an ungrammatical variant with the enclitic missing.

3. For each grammatical sentence, I created variants that contained every incorrect prefix and variants that contained every incorrect enclitic.

After duplicates were removed, this produced a list of 89 grammatical and 220 ungrammatical sentences, for a total of 309. This set of test sentences is shown in Appendix H.

Without too much effort, it was possible to permute the grammatical intransitive sentences by hand to produce the negative examples mentioned above: sentences made up of correctly-formed words in the correct basic word order, but with an ungrammatical agreement pattern. However, the number of analogous ungrammatical transitive sentences is much larger. In order to test the grammar thoroughly, I wrote a small program to generate all possible transitive sentences containing the verb *q̓ínun* 'see' of the form:

(121) **prefix-***q̓ínun*=**enclitic subject object**

The possible fillers for each slot in (121) are shown in Table 6.13.

Table 6.13: Fillers for positions in (121)

prefix	*i-, pa-, á-, pá-, patá-,* and ∅
enclitic	*=naš, =na, =nataš, =nam, =pam, =maš, =mataš,* and ∅
subject	subject forms in Table 6.5
object	object forms in Table 6.5

As mentioned above, the lexicon of the Sahaptin grammar, and consequently the test sentences, use the various forms of the personal pronoun to represent the various possible person, number, case, and proximity values of subject and object noun phrases. In addition to appropriately case-marked pronouns, the subject and object slots may contain third person pronouns marked as the topic with *-TOP*.

Generating every permutation of the above pattern produced 6048 sentences, but some additional filtering was required. First, since I have assumed that topic marking is only relevant when a clause contains two third person arguments, I removed all sentences where the *-TOP* suffix appeared with a first or second person pronoun. Second, 192 of the permutations of (121) are actually duplicates of the ungrammatical transitive test sentences created by hand above, so I removed those as well. After filtering, 5856 randomly-generated sentences remained. Added to the aforementioned 309 examples, this made 6165 unique test sentences.

After using the customization system to generate a grammar of Sahaptin (based on the choices file in Appendix G), I used the grammar to attempt to parse every test sentence. All 89 sentences corresponding to patterns in Tables 6.3 and 6.4 parsed and were assigned exactly one analysis.[11] Among the ungrammatical sentences, 5848 out of 5856 failed to parse, as expected. To my surprise, however, eight of the sentences did parse. These sentences were:

(122) a. *i-q̓ínun pɨ́n-TOP piinamanáy*
3sg-see 3sg.nom-top 3du.obj

b. *i-q̓ínun pɨ́n-TOP paamanáy*
3sg-see 3sg.nom-top 3pl.obj

c. *pa-q̓ínun piiní paanáy*
3nonsg-see 3du.nom 3sg.obj

d. *pa-q̓ínun pmáy paanáy*
3nonsg-see 3pl.nom 3sg.obj

e. *pa-q̓ínun piiní-TOP piinamanáy*
3nonsg-see 3du.nom-top 3du.obj

[11] Multiple analyses would not necessarily have been wrong—some sentences in some languages are structurally ambiguous—but the grammatical Sahaptin sentences in the test suite are marked explicitly enough for agreement that none was ambiguous.

f. *pa-q̓ínun* *piiní-TOP* *paamanáy*
3NONSG-see 3DU.NOM-TOP 3PL.OBJ

g. *pa-q̓ínun* *pmáy-TOP* *piinamanáy*
3NONSG-see 3PL.NOM-TOP 3DU.OBJ

h. *pa-q̓ínun* *pmáy-TOP* *paamanáy*
3NONSG-see 3PL-TOP.NOM 3PL.OBJ

Notice that the eight sentences fall into three patterns. The first two sentences have a third person singular topical subject and a third person non-singular non-topical object. The next two sentences have a third person non-singular non-topical subject and a third person singular non-topical object. The last four sentences have a third person non-singular topical subject and a third person non-topical object. These are precisely the patterns that are absent from Table 6.4; their corresponding test sentences were therefore not included in the list of 89 grammatical sentences. In the absence of data about the patterns from R&R, I made plausible guesses about the correct marking on the verb, using *i-* in the first two sentences and *pa-* in the last six.

In order to determine whether these guesses were correct, Sharon Hargus presented the Yakima Sahaptin equivalents of the sentences in (122) by telephone to Virginia Beavert, a native speaker of that dialect, who accepted all eight of them with the readings shown in (123) below:[12]

(123) a. *i-q̓ínun-a* *pɨ́nk* *piinamanák*
3SG-see-PST 3SG.NOM-TOP 3DU.OBJ
'He saw them (DU).'

[12] Note that, in order for these sentences to be acceptable, they had to be in the past tense, a feature not modeled in my Sahaptin grammar fragment.

b. *i-q̓ínun-a pɨ́nk piimanák*
3SG-see-PST 3SG.NOM-TOP 3PL.OBJ
'He saw them.'

c. *pa-q̓ínun-a piiník piinák*
3NONSG-see-PST 3DU.NOM 3SG.OBJ
'They (DU) saw him.'[13]

d. *pa-q̓ínun-a pmák piinák*
3NONSG-see-PST 3PL.NOM 3SG.OBJ
'They saw him.'

e. *pa-q̓ínun-a piiník piinamanák*
3NONSG-see-PST 3DU.NOM-TOP 3DU.OBJ
'They (DU) saw them (DU).'

f. *pa-q̓ínun-a piiník piimanák*
3NONSG-see-PST 3DU.NOM-TOP 3PL.OBJ
'They (DU) saw them.'

g. *pa-q̓ínun-a pmák piinamanák*
3NONSG-see-PST 3PL.NOM-TOP 3DU.OBJ
'They saw them (DU).'

h. *pa-q̓ínun-a pmák piimanáy*
3NONSG-see-PST 3PL-TOP.NOM 3PL.OBJ
'They saw them.'

[13] Dr. Beavert considered this example somewhat less acceptable, saying that it is "[a] little awkward, but has meaning."

My Sahaptin grammar, therefore, which was created simply by filling out the customization system questionnaire, has provided correct analyses for all of the test sentences, including eight that fell outside of the agreement patterns described by R&R.

6.4 Summary

Based on these results, I conclude that even Sahaptin, a language with extremely complex argument marking morphology, can be modeled using the customization system. It is important to note that the system was not designed with the grammatical facts of Sahaptin in mind, and with two exceptions, the system did not need to be modified to enable it to handle Sahaptin. One of the exceptions was trivial: formerly, grammars produced by the system were treating the character "=" as punctuation, stripping it out and breaking words containing it. The other exception has already been described: the enhancement of the questionnaire to allow morphemes to address arguments based on their relative position on the direct-inverse scale. However, although this latter change was a substantive one that allows grammars to be described more compactly, it did not increase the descriptive power of the system, since languages showing that pattern of agreement can be alternatively modeled using duplicated, homophonous morphemes. Furthermore, this enhancement to the system is a good example of the feedback loop between grammar engineering and the development of the customization system, in which new languages with new phenomena (or new variations of old phenomena) inform the design and, in some cases, the descriptive power of the system.

After constructing the Sahaptin grammar and test suite described here, it was natural to include it in two places in the customization system. First, it is now one of the regression tests that is regularly run to ensure that future enhancement of the system does not break earlier features. Second, Sahaptin has been added to the list of sample grammars accessible from the main page of the questionnaire—by clicking on links in this list, users can see detailed examples of how to fill out the questionnaire to model their target language.

Chapter 7
CONCLUSION

7.1 The Matrix and Typology

Since its initial version, the investigation of language universals has been an important aim of the Grammar Matrix, with Bender et al. (2002) describing it as following “the HPSG community’s general bottom-up approach to language universals, which involves aiming for good coverage of a variety of languages first, and leaving the task of what they have in common for later.” Bender and Flickinger (2005) further refined the relationship between the Matrix and typology in their initial paper about Matrix libraries (then called “modules”), writing:

> Our strategy is thus consistent with a bottom-up, data-driven investigation of linguistic universals and constraints on cross-linguistic variation. As the number and breadth of implemented grammars grows, we expect linguistic predictions to emerge and become part of improved modules, particularly with respect to interactions among the distinct phenomena covered. Our approach should in time be instrumental in assisting large-scale typological investigations (covering hundreds of languages), making use of the linguistically precise constraints encoded in these modules to uncover deeper and more subtle facts about languages. (Bender and Flickinger 2005:204)

In the subsequent development of Matrix libraries (including those described in Drellishak and Bender 2005, Poulson forthcoming, and this dissertation), the relationship between the Matrix and typology has been largely one-way: Matrix developers have drawn on typological surveys and literature in the design of Matrix libraries. The recent development of the customization system, however, has begun to open up the possibility of the Matrix project contributing to typology. In this section, I will situate the Matrix and Matrix libraries within the field of typology, and discuss the kinds of contributions that are made possible by the methodology used in Matrix development.

For some researchers, the purpose of linguistic typology has been to identify how much languages differ, and furthermore, how much languages *can* differ. For them, typology is about finding explanations, either functional or innate, for the differences among human languages. However, Bickel (2007:239) has written that, "Over the past decade, typology has begun to emancipate itself from this goal and to turn from a method into a full-fledged discipline, with its own research agenda, its own theories, its own problems." He identifies the main question in typology as, "What's where why?" (Bickel 2007:239) The Grammar Matrix, in contrast, is still focused on the more traditional question of what is a possible human language—our approach cannot provide explanation for the distribution of language features, but it can help in determining the possible range of variation.

7.1.1 Regression Testing

Developers of Matrix libraries are interested in analyzing and modeling as many variations as possible of the linguistic phenomena covered by the libraries. Doing so within a single formal, computational framework has advantages over non-computational methods; in particular, it makes it easier to test our analyses. When developing a single grammar in our software framework, the grammar can be used to parse a large number of test sentences into their semantic representations, then those representations can be used to generate sentences. If the grammar contains a flaw, then one of four results will occur: either some grammatical sentences will not parse, some ungrammatical sentences will, or some grammatical sentences will parse too many times (producing spurious ambiguity) or with the wrong semantics. By storing suites of test sentences and their semantics and, during grammar development, repeatedly verifying that the grammar is behaving as expected, we can ensure that later changes to the grammar do not invalidate earlier development.[1]

This single-grammar testing methodology can be extended to the process of library development. We create test suites that contain, in addition to a set of sentences and their semantics, a choices file describing the language of the sentences. During development of new libraries, grammars can be generated from the choices files and their behavior validated against the stored test sentences. This ensures that future libraries adding analyses of new phenomena do not interfere with the parts of grammars created by earlier libraries. To this end, every example grammar described in this dissertation has been formalized as such a test suite and added to our process of development—before any change is allowed to the system, the developer must verify that no regressions have occurred in any existing test suite. This methodology for detecting regressions in the customization system helps keep it rigorously correct and consistent, at least to the extent that our test suites are representative of real languages.

Formally implementing analyses as Matrix libraries exposes the underlying analyses to potential disproof. Creating a library and its associated section of the questionnaire commits the developer to a claim: that the phenomenon covered by the library can be described in the questionnaire and properly modeled by his or her analysis. If a future user of the system finds a language that falls outside of the claimed typological range or whose behavior, when described, does not match that of the grammar produced by the customization system, then the analysis in the system is known to require improvement.

[1] The usefulness of grammar engineering for linguistic hypothesis testing is explored in more detail by Bender (2008a) and Bender et al. (to appear).

7.1.2 Testing Typology

Formalizing analyses on a common framework has another advantage. Each library is intended to cover some linguistic phenomenon, based on the existing typological literature describing that phenomenon. Such literature is generally produced by a typologist surveying single-language or language-family descriptions written by other authors. However, authors of descriptive grammars do not always agree on theoretical foundations or terminology. Typologists take pains to reconcile these differences, but risks exist, either that two unlike phenomena will be conflated into a single type, or that two similar phenomena will be mistaken for different types. Formalizing an analysis of phenomenon as a Matrix library can serve to detect such mistakes. In Chapters 4 and 6, for example, we saw that direct-inverse languages and languages with scale-sensitive case marking can be modeled with precisely the same formal machinery, which argues that instead of two related language types, they are in fact a single type.

The formalization of analyses cross-linguistically, therefore, can both improve the analyses and check the correctness of typological surveys. The implementation of such analyses as libraries in the customization system, furthermore, serves to amplify these advantages. It is now the work of a few hours for a linguist to fill out the questionnaire, generate a grammar, and test it on data from the target language. This greatly decreases the investment in time that must be made to develop a grammar, and so increases the ease with which analyses can be tested. The results turned up are valuable contributions to HPSG, to “bottom-up typology”, to theoretical syntax, or to linguistic typology, depending on the extent to which we can convince ourselves that the results are theory-independent.

The contribution of Matrix development to typology is analogous to the contribution of users of the customization system to the Matrix. In the same way that creating and validating a grammar using the customization system tests the correctness of the system, the implementation of a Matrix library based on the typological literature tests the generalizations presented there. The Matrix project, therefore, provides feedback to the field of linguistic typology. Typological generalizations that have not been tested through formalization are in danger of being flawed in the same way that a single-language grammar is in danger—both are complex systems intended to account for a range of data, and complex systems can contain inconsistencies that are difficult to detect without computational aid.

Nonetheless, care must be taken not to make claims that are too strong for the utility of the Matrix to typology. Libraries are almost certain to miss some language type, so the analyses, and hence the generalizations based on them, are prone to be incomplete. The solution, of course, is to continue to do future work on Matrix libraries, so that over time libraries really approach full cross-linguistic coverage.

7.1.3 A Database of Choices Files

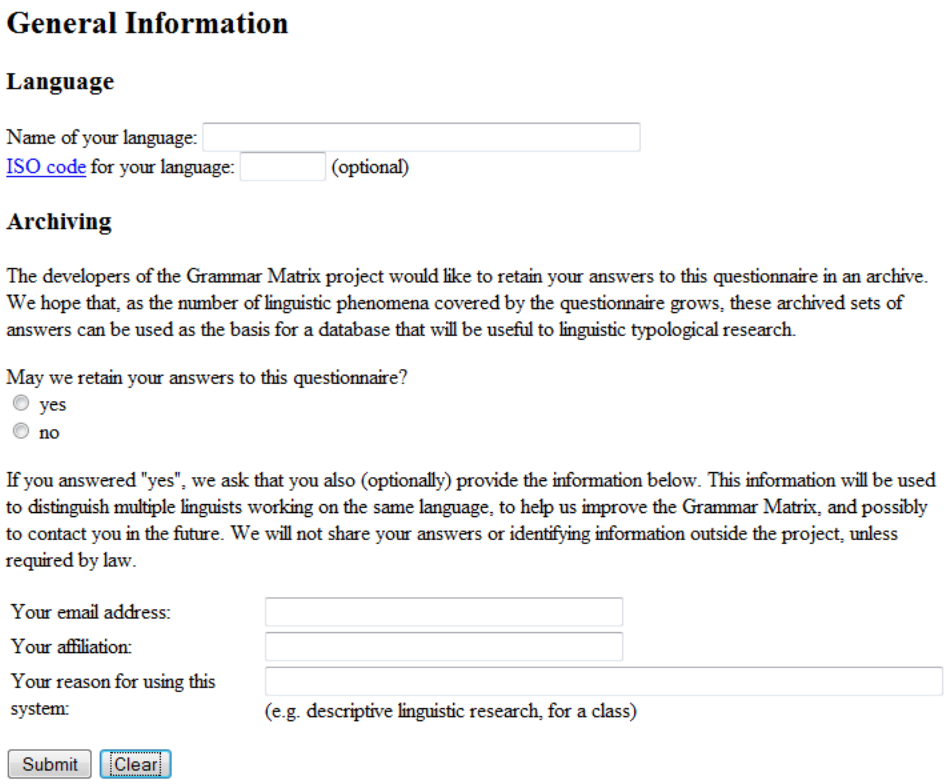
Recall that the customization system stores the state of the questionnaire, including all the user's current answers, in a text file called a choices file. The questions in the questionnaire define a many-dimensional space of language types, and a choices file represents a point in that space that identifies the target language. A collection of choices files could be a valuable tool for typological research, since it would allow typologists to compare languages categorized into types that are known to be consistent.

It may seem that the answers in a choices file are analogous to parameter settings in the Chomskyan typological tradition. This is not the case: while the choices space is intended to characterize variation among human languages, the dimensions in the space were not designed to correspond to any theoretical entity like UG or the contents of the language faculty, and also tend to be less abstract than P&P-style parameters. Furthermore, the design of many of the dimensions was affected by engineering constraints and matters of questionnaire design, so some of the dimensions are better motivated typologies than others. For example, the set of possible `case-marking` choices looks quite like a typology of core case marking, taking values like `nom-acc`, `erg-abs`, `split-n`, `split-v`, etc. The choices associated with number, on the other hand, much more directly represent the number hierarchy in the target language, for reasons discussed in §5.4.1—however, the choices for number could be post-processed in order to determine, for example, the list of all leaf types (e.g., `singular`, `dual`, and `plural`) in each language, which might be more useful to a typologist.

In order to begin collecting a database of choices files, I modified the customization system to enable it to save a copy of the choices file when a user successfully fills out a properly validating set of answers and presses the "Create Grammar" button. I also added questions to the "General Information" subpage of the questionnaire asking for consent to have the choices file stored in this database and to provide some additional information about the user, including an email address. This section of the questionnaire is shown in Figure 7.1. Since these changes went live in early September 2008, more than 450 choices files created by 15 different users and describing 15 different languages (Ainu, Arabic, Basque, Cherokee, Classical Japanese, English, French, Hixkaryana, Japanese, Korean, Mauritian Creole, Norwegian, Tamil, Thai, and Turkish) have been saved. We hope that in the future, as this list grows, it can be assembled into a database of languages against which typologists can make complex queries.

7.1.4 Computational Linguistic Typology

I refer to the approach described above that the Matrix project takes toward language universals as **computational linguistic typology**. By this, I do not mean the use of computers to organize and publish data and databases about language variation, an increasingly important part of modern linguistic typology (see §2.2.3 for some examples). Rather, I use the term to refer to a methodology that involves the formalization, in a common framework, of grammars of genetically and typologically



General Information

Language

Name of your language:

ISO code for your language: (optional)

Archiving

The developers of the Grammar Matrix project would like to retain your answers to this questionnaire in an archive. We hope that, as the number of linguistic phenomena covered by the questionnaire grows, these archived sets of answers can be used as the basis for a database that will be useful to linguistic typological research.

May we retain your answers to this questionnaire?

○ yes
○ no

If you answered "yes", we ask that you also (optionally) provide the information below. This information will be used to distinguish multiple linguists working on the same language, to help us improve the Grammar Matrix, and possibly to contact you in the future. We will not share your answers or identifying information outside the project, unless required by law.

Your email address:

Your affiliation:

Your reason for using this system: (e.g. descriptive linguistic research, for a class)

Figure 7.1: The general information section of the questionnaire

diverse natural languages. This methodology was used in the original development of the Grammar Matrix, and it continues to be used in our efforts to add libraries covering more phenomena to the customization system.

I believe that computational linguistic typology can bridge the gap between the Greenbergian and Chomskyan approaches to typology. Recall how I characterized the two in Chapter 2: Greenbergian typology as data-driven, shallow, and broad, Chomskyan typology as theoretically focused, narrow, and deep. These characterizations might meet with some resistance. For example, Nichols (2007) attempts to counter what she considers the misperception that “typology deals with only superficial grammatical phenomena, while formal grammar deals with deeper abstractions.” She writes:

> It is true that until fifteen or twenty years ago, anything larger than a small sample was necessarily restricted to what I call “lookup characters”—the relatively superficial structural properties that one could find in most of the then available grammars simply by looking them up. By now, things are very different. I see no difference in analytic or theoretical profundity or abstraction between generative parameters and original contributions of typology such as direct object vs. primary object (Dryer 1986), verb-framed vs. satellite-framed lexicalization patterns (Talmy 1985, Slobin 2004), various aspects of alignment (e.g., Dixon 1994, Dixon & Aikhenvald (eds.) 2000), differential object marking (Bossong 1998, Aissen 2003), referential density (Bickel 2003), and others. (Nichols 2007:233, references retained from original)

Of course, it is true that typology does not merely look at surface phenomena. As mentioned in Chapter 2, typology has always been the comparison of analyses, not directly the comparison of languages. But even if it is true that the depth of analyses upon which typologists base their surveys has increased, truly broad surveys are still limited by variations in terminology and theoretical foundations in descriptions of languages. Practically, to be certain that the languages being surveyed really are of the same type, typologists performing surveys would have to essentially re-analyze each language to ensure the similarities were completely well-founded, or else go to enormous effort to develop standards to which other researchers contributing to the project would have to conform. Either solution is expensive in terms of resources and time.

This is not to denigrate the efforts of typologists. Typologists know about the problem of dissimilar analyses and take pains to ameliorate it, and the Matrix has certainly depended on results produced by typologists. But in a computational framework, especially with a tool like the customization system, languages can be analyzed in greater depth at a lower cost, and that means more and more consistent analyses for typologists to compare. There is little or no variation in the shared foundation underlying the grammars, and the mappings between surface forms and semantic representations are thorough, consistent, and operate in both directions.

The test cases in this dissertation, for example, had to be developed down to the smallest detail in order to correctly analyze every test sentences, and that depth of analysis bore fruit. While developing the Sahaptin grammar, I recognized that some verbal inflection agreed with the more highly-ranked argument, and enhanced the system to make this easy to describe; then, during the development of the Cree test case (which occurred later than that of the Sahaptin grammar), I realized the same pattern appeared in that language. This is not the sort of language feature that is easily gleaned from a printed grammar—deep, precision grammar development was required.

It is for this reason that I think computational linguistic typology has the virtues of both the Greenbergian and Chomskyan approaches to typology. With computational grammars rapidly developed on an entirely consistent foundation, researchers can have both deep analysis and broad surveys, without having to sacrifice one for the other. Of course, the customization system's analyses are currently expressed only in HPSG, but there is no reason why other systems comparable to the customization system could not be created for any framework where grammars can be implemented in software. In fact, it would be a great benefit to the Matrix project and to the computational linguistic approach to typology if there were additional systems based in different formalisms that could produce grammars based on typological descriptions, perhaps even the same choices files. If the output of two systems in different frameworks could be compared, it would be clearer what parts of the analyses were both language- and framework-independent. Hopefully, we are not far from the day when the computational methods described here begin to contribute results back to typology.

7.2 Future Work

The Matrix customization system is an ongoing software project with an extremely ambitious goal.[2] There are always more linguistic phenomena to be analyzed, and every time we add a phenomenon to the system, there is the possibility that the new functionality will expose a weakness in a previous library. The libraries described in this dissertation are no exception. In this section, I will discuss several areas that have been exposed by my work where future work is required.

7.2.1 Coordination and Agreement

The coordination library (Drellishak and Bender 2005) allows the user-linguist to describe a target language with coordination of nouns, noun phrases, verb phrases, sentences, or any combination of those phrase types. Under its analysis, only constituents of the same type can be coordinated, and the resulting phrase will be of the same type—for example, coordinating two verb phrases produces another verb phrase. The supertype from which all coordinated phrases derive is (slightly abbreviated):

(124)
$$\left[\begin{array}{ll} \textit{coord-phrase} & \\ \text{CAT} & \left[\begin{array}{l} \text{HEAD}\,|\,\text{MOD}\ \boxed{1} \\ \text{VAL}\ \boxed{2} \end{array}\right] \\ \text{LCOORD-DTR} & \boxed{3} \left[\begin{array}{ll} \text{CAT} & \left[\begin{array}{ll} \text{HEAD}\,|\,\text{MOD} & \boxed{1} \\ \text{VAL} & \boxed{2} \end{array}\right] \end{array}\right] \\ \text{RCOORD-DTR} & \boxed{4} \left[\begin{array}{ll} \text{CAT} & \left[\begin{array}{ll} \text{HEAD}\,|\,\text{MOD} & \boxed{1} \\ \text{VAL} & \boxed{2} \end{array}\right] \end{array}\right] \\ \text{ARGS} < \boxed{3}, \boxed{4} > & \end{array}\right]$$

[2] The text of ticket #1 in the Matrix bug-tracking system reads, “Build comprehensive, implemented, correct Universal Grammar.”

Notice that the VAL and MOD features of the two coordinands and the coordinated phrase are identified. VAL contains all the valence features; identifying it ensures, for example, that two coordinated verb phrases will share the same subject. Similarly, identifying MOD ensures that coordinated adjectives or adverbs (which are not yet definable through the customization system) will modify the same phrase.

Crucially, *coord-phrase* does not identify the three syntactic HEAD features or the three semantic INDEX features. Identifying INDEX would certainly be incorrect, since it would prevent, for example, the coordination of singulars and plurals. Identifying HEAD is also too strong; coordinated phrases like *quickly and without complaint* (in which adverb and prepositional phrase are coordinated) would be disallowed.

However, leaving HEAD and INDEX unconstrained is obviously wrong in grammars with syntactic features like CASE or semantic features like PERSON or NUMBER. Failing to restrict these features at all would allow sentences like the following to parse:

(125) *Her and he left (mismatch in case)

(126) *She and they am here (mismatch in person and number)

How languages deal with mismatches of these kinds is called **resolution**. Although the coordination library did nothing to support resolution, leaving it up to the user-linguist to further constrain the output grammar if desired, my addition of libraries for case and agreement has raised the lack of resolution support to a new level of visibility. Any grammar with both case and coordination, for example, will parse sentences with any case on the coordinands.

The solution, of course, is more development of the customization system. The typological literature about case, person, number, and gender resolution must be consulted, a general analysis for resolution developed, and new questions or sections about resolution added to the questionnaire.

7.2.2 Syntactic Ergativity

As mentioned in §3.2.2, the current version of the customization system always produces grammars that are syntactically accusative. At the moment, this does not much affect the grammars, since syntactic ergativity by definition is an inter-clausal phenomenon, and the only phenomenon describable in the questionnaire that involves multiple clauses is the coordination of sentences. However, as the customization system grows, it will likely gain support for other phenomena that are sensitive to syntactic ergativity, such as relative clauses, control, and binding.

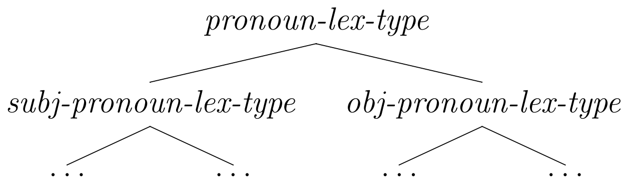
At some point, the libraries for case, word order, coordination, and any that support multi-clausal phenomena must be revisited and extended to support syntactic ergativity. Fortunately, there is already an HPSG analysis of syntactic ergativity by Manning and Sag (1995); it simply needs to be integrated everywhere it is relevant. Furthermore, questions will need to be added to the questionnaire that allow the user-linguist to describe the behavior of each multi-clausal phenomenon with respect to syntactic ergativity.

7.2.3 Lexical Type Hierarchies

Type hierarchies are particularly powerful in linguistic description because they allow the compact expression of generalizations. For example, consider a language that has four types of verbs: intransitive-stative, intransitive-non-stative, transitive-stative, and transitive-non-stative. Rather than having to repeat the feature constraints that describe transitive in two places (on *trans-stative-verb-lex* and *trans-non-stative-verb-lex*), we can instead use a type hierarchy and inheritance to state those constraints once (in a *trans-verb-lex* type, say), then have *trans-stative-verb-lex* and *trans-non-stative-verb-lex* inherit from *trans-verb-lex*. This mechanism is powerful and saves significant redundancy, especially when describing a large lexicon, where overlapping lexical types are common.

As part of my work, I added the ability to describe an arbitrarily large number of lexical types in the questionnaire. However, the current version of the questionnaire does not support the kind of structured lexicon described above. In the Sahaptin grammar in Chapter 6, for example, each of the twenty pronouns (eleven subject forms, nine object forms) were defined as separate lexical types, each of which specified a predicate named `_pronoun_n_rel`. This is redundant and repetitive for the user. A more appropriate lexical hierarchy would have included at least as much structure as the following example, in which *pronoun-lex-type* specifies the pronoun relation, *subj-pronoun-lex-type* specifies subject case, *obj-pronoun-lex-type* specifies object case, and the actual lexical items inherit from one of the latter two types:

(127)



Work is ongoing by Poulson (forthcoming) to address this problem as part of her work on tense and aspect. In a future version of the system, the Lexicon section of the questionnaire will be split into two sections. The first will allow the description of lexical types and their hierarchical relationships; the second will allow the definition of actual lexical items.

7.2.4 Usability

A final issue that has become clear after my enhancements to the system is the general problem of usability. It used to be the case that filling out one section of the questionnaire had little impact on other sections, but now the questionnaire has become very dynamic. Drop-downs for feature names and values in the Lexicon section, for example, are filled in with lists of options based on the user-linguist's answers to questions in section including Case, Number, and Gender. Although these interactions are intended to make life easier, they can be surprising.

In addition, there are sections of the questionnaire that allow very complex structured answers to be provided, but in a way that may lead to confusion. In the Number and Gender sections, for example, the user-linguist is allowed to describe type hierarchies of arbitrary complexity. Filling out anything more than a handful of number or gender values can rapidly become confusing. Filling out these sections can be more like writing code in a visual programming language than answering a series of questions.

These shortcomings can be addressed by adding more visual feedback and more in-context help to the user. We have discussed several ideas for this within the project. Two of the leading ideas are tooltips and graphical hierarchies. Tooltips would be small icons appearing near sections of the form that are likely to be confusing, and if the user moves the mouse over one, a paragraph or two of explanation would appear. Graphical hierarchies would be rectangular areas off to the right side of the page that display a graphical tree-diagram that updates as the user is constructing a hierarchy. These kinds of enhancements would not take a large investment of resources, but neither are they trivial; in particular, they require a web-development skillset that not all computational linguists possess. Hopefully, like the "red asterisk" mechanism in the current system, future enhancements of this kind can be implemented during breaks in more substantial library development by sufficiently usability-minded researchers.

7.3 Conclusion

In this dissertation, I have explored the typology of case, agreement, and direct-inverse languages, provided an HPSG analysis of each, and described the implementation of those analyses as libraries in the LinGO Grammar Matrix customization system. Several aspects of my analyses are novel in HPSG; in particular, my inclusion of a CASE feature on adpositional heads and my entire analysis of direct-inverse languages. I described the customization system questionnaire, which asks typological questions whose answers the system uses to automatically produce grammars of natural languages in software. Part of the grammar-creation process involves hierarchy augmentation, a powerful technique for turning linguistic description into detailed, complex type hierarchies. I verified the correct function of all my libraries by creating test cases for a wide variety of languages. This included a detailed case study of Sahaptin, a language that illustrates all of the phenomena for which I have created libraries. Finally, I have discussed the interaction of computational linguistics and linguistic typology and, I hope, provided convincing arguments that the two fields can both benefit from research projects like the one I have described here.

BIBLIOGRAPHY

Bateman, John A., Kruijff-Korbayová, Ivana and Kruijff, Geert-Jan. 2005. Multilingual Resource Sharing Across Both Related and Unrelated Languages: An Implemented, Open-Source Framework for Practical Natural Language Generation. *Research on Language and Computation* 3, 191–219.

Beesley, Kenneth R. and Karttunen, Lauri. 2003. *Finite State Morphology*. Stanford: CSLI.

Bender, Emily M. 2007. Combining Research and Pedagogy in the Development of a Crosslinguistic Grammar Resource. In Tracy Holloway King and Emily M. Bender (eds.), *Proceedings of the GEAF07 Workshop*, pages 26–45, Stanford: CSLI.

Bender, Emily M. 2008a. Grammar Engineering for Linguistic Hypothesis Testing. In Nicholas Gaylord, Alexis Palmer and Elias Ponvert (eds.), *Proceedings of the Texas Linguistics Society X Conference: Computational Linguistics for Less-Studied Languages*, pages 16–36, Stanford: CSLI Publications ONLINE.

Bender, Emily M. 2008b. Radical Non-Configurationality without Shuffle Operators: An Analysis of Wambaya. In Stefan Müller (ed.), *The Proceedings of the 15th International Conference on Head-Driven Phrase Structure Grammar*, pages 6–24, Stanford: CSLI Publications.

Bender, Emily M. and Flickinger, Dan. 2005. Rapid Prototyping of Scalable Grammars: Towards Modularity in Extensions to a Language-Independent Core. In *Proceedings of the 2nd International Joint Conference on Natural Language Processing IJCNLP-05*, Jeju Island, Korea.

Bender, Emily M., Flickinger, Dan and Oepen, Stephan. 2002. The Grammar Matrix. In *Proceedings of COLING 2002 Workshop on Grammar Engineering and Evaluation*, Taipei, Taiwan.

Bender, Emily M., Flickinger, Dan and Oepen, Stephan. to appear. Grammar Engineering and Linguistic Hypothesis Testing: Computational Support for Complexity in Syntactic Analysis .

Bender, Emily M. and Good, Jeff. 2005. Implementation for Discovery: A Bipartite Lexicon to Support Morphological and Syntactic Analysis. *Chicago Linguistic Society 41: The Panels* .

Bender, Emily M. and Goss-Grubbs, David. 2008. Semantic Representations of Syntactically Marked Discourse Status in Crosslinguistic Perspective. In *Proceedings of Semantics in Systems for Text Processing (STEP 2008)*, College Publications.

Bickel, Balthasar. 2007. Typology in the 21st Century: Major Current Developments. *Linguistic Typology* 11, 239–251.

Bickel, Balthasar, Comrie, Bernard and Haspelmath, Martin. 2008. The Leipzig Glossing Rules: Conventions for Interlinear Morpheme by Morpheme Glosses Max Planck Institute for Evolutionary Anthropology and Department of Linguistics, University of Leipzig.

Bickel, Balthasar and Nichols, Johanna. 2002. Autotypologizing Databases and Their Use in Fieldwork. In *Proceedings of the International LREC Workshop on Resources and Tools in Field Linguistics*, Las Palmas, Canarias.

Blake, Barry J. 2001. *Case, Second Edition*. Cambridge: Cambridge University Press.

Boeckx, Cedric. 2008. Approaching Parameters from Below Http://ling.auf.net/lingBuzz/000641.

Borthen, Aja and Haugereid, Petter. 2005. Representing Referential Properties of Nominals. *Research on Language and Computation* 3, 221–246.

Butt, Miriam, Dyvik, Helge, King, Tracy Holloway, Masuichi, Hiroshi and Rohrer, Christian. 2002. The Parallel Grammar Project. In *Proceedings of COLING-2002 Workshop on Grammar Engineering and Evaluation*, pages 1–7.

Cahill, Aoife, Burke, Michael, Forst, Martin, O'donovan, Ruth, Rohrer, Christian, van Genabith, Josef and Way, Andy. 2005. Treebank-Based Acquisition of Multilingual Unification Grammar Resources. *Research on Language and Computation* 3, 247–279.

Chomsky, Noam. 1981. *Lectures on Government and Binding*. Dordrecht: Foris.

Chomsky, Noam. 1995. *The Minimalist Program*. Cambridge, MA: MIT Press.

Chomsky, Noam. 2005. Three Factors in Language Design. *Linguistic Inquiry* 36, 1–22.

Chomsky, Noam, Hauser, Marc D. and Fitch, W. Tecumseh. 2002. The Faculty of Language: What Is It, Who Has It, and How Did It Evolve? *Science* 298, 1569–79.

Comrie, Bernard. 1989. *Language Universals & Linguistic Typology, Second Edition*. Chicago: University of Chicago.

Comrie, Bernard and Smith, Norval. 1977. Lingua Descriptive Series: Questionnaire. *Lingua* 42, 1–72.

Copestake, Ann. 2002. *Implementing Typed Feature Structure Grammars*. Stanford: CSLI.

Copestake, Ann and Flickinger, Dan. 2000. An Open Source Grammar Development Environment and Broad-coverage English Grammar Using HPSG. In *Proceedings of the Second conference on Language Resources and Evaluation*, Athens, Greece.

Copestake, Ann, Flickinger, Dan, Pollard, Carl and Sag, Ivan A. 2005. Minimal Recursion Semantics: An Introduction. *Research on Language & Computation* 3(2–3), 281–332.

Corbett, Greville G. 1991. *Gender*. Cambridge: Cambridge University Press.

Corbett, Greville G. 2000. *Number*. Cambridge: Cambridge University Press.

Corbett, Greville G. 2006. *Agreement*. Cambridge: Cambridge University Press.

Croft, William. 1990. *Typology and Univerals*. Cambridge: Cambridge University Press.

Cysouw, Michael. 2003. *The Paradigmatic Structure of Person Marking*. Oxford: Oxford University Press.

Dahlstrom, Amy. 1991. *Plains Cree Morphosyntax*. New York: Garland.

Dixon, R. M. W. 1968. *The Dyirbal Language of North Queensland*. Ph. D.thesis, University of London.

Dixon, R. M. W. 1972. *The Dyirbal Language of North Queensland*. Cambridge: Cambridge University Press.

Dixon, R. M. W. 1979. Ergativity. *Language* 55, 59–138.

Dixon, R. M. W. 1994. *Ergativity*. Cambridge: Cambridge University Press.

Donato, Robert Di, Clyde, Monica D. and Vansant, Jacqueline. 2004. *Deutsch: Na klar!*. New York: McGraw-Hill.

Drellishak, Scott. 2008. Complex Case Phenomena in the Grammar Matrix. In Stefan Müller (ed.), *The Proceedings of the 15th International Conference on Head-Driven Phrase Structure Grammar*, pages 67–86, Stanford: CSLI Publications.

Drellishak, Scott and Bender, Emily M. 2005. A Coordination Module for a Crosslinguistic Grammar Resource. In *Proceedings of the 12th International Conference on Head-Driven Phrase Structure Grammar*, Lisbon, Portugal.

Dryer, Matthew S. 2008. Order of Subject, Object and Verb. In Martin Haspelmath, Matthew S. Dryer, David Gil and Bernard Comrie (eds.), *The World Atlas of Language Structures Online*, Munich: Max Planck Digital Library.

Flickinger, Dan. 2000. On Building a More Efficient Grammar by Exploiting Types. *Natural Language Engineering* 6 (1), 15 – 28.

Flickinger, Dan and Bender, Emily M. 2003. Compositional Semantics in a Multilingual Grammar Resource. In Emily M. Bender, Dan Flickinger, Frederik Fouvry and Melanie Siegel (eds.), *Proceedings of the Workshop on Ideas and Strategies for Multilingual Grammar Development, ESSLLI 2003*, pages 33–42, Vienna, Austria.

Givón, T. 1994. The Pragmatics of De-transitive Voice: Functional and Typological Aspects of Inversion. In T. Givón (ed.), *Voice and Inversion*, pages 3–44, Amsterdam: Benjamins.

Greenberg, Joseph H. 1963. Some Universals of Grammar with Particular Reference to the Order of Meaningful Elements. In Joseph H. Greenberg (ed.), *Universals of Language*, pages 73–113, Cambridge, MA: MIT Press.

Hahm, Hyun-Jong. 2006. Person and Number Agreement in American Sign Language. In *Proceedings of the 13th International Conference on Head-Driven Phrase Structure Grammar*, pages 195–211, Varna, Bulgaria.

Hargus, Sharon. to appear. Design and Organization of *Ichishkíin Sɨ́nwit*. In Virginia Beavert and Sharon Hargus (eds.), *Ichishkíin Sɨ́nwit Yakama/Yakima Sahaptin Dictionary*, Toppenish and Seattle: Heritage University and University of Washington Press.

Haspelmath, Martin. 2009. Terminology of Case. In Andrej Malchukov and Andrew Spencer (eds.), *The Oxford Handbook of Case*, pages 505–517, Oxford: Oxford University Press.

Haspelmath, Martin, Dryer, Matthew, Gil, David and Comrie, Bernard (eds.). 2005. *The World Atlas of Language Structures*. Oxford: Oxford University Press.

Hellan, Lars and Haugereid, Petter. 2003. NorSource: An Exercise in Matrix Grammar-Building Design. In Emily M. Bender, Dan Flickinger, Frederik Fouvry and Melanie Siegel (eds.), *Proceedings of the Workshop on Ideas and Strategies for Multilingual Grammar Development, ESSLLI 2003*, pages 41–48, Vienna, Austria.

Holisky, Dee Ann. 1987. The Case of the Intransitive Subject in Tsova-Tush (Batsbi). *Lingua* 71, 103–32.

Kathol, Andreas. 1999. Agreement and the Syntax-morphology Interface in HPSG. In R. Levine and G. Green (eds.), *Readings in HPSG*, pages 223–274, Cambridge: Cambridge University Press.

Keenan, Edward L. and Comrie, Bernard. 1977. Noun Phrase Accessibility and Universal Grammar. *Linguistic Inquiry* 8, 63–99.

Kim, Roger, Dalrymple, Mary, Kaplan, Ronald M., King, Tracy Holloway, Masuichi, Hiroshi and Ohkuma, Tomoko. 2003. Multilingual Grammar Development via Grammar Porting. In Emily M. Bender, Dan Flickinger, Frederik Fouvry and Melanie Siegel (eds.), *Proceedings of the ESSLLI 2003 Workshop "Ideas and Strategies for Multilingual Grammar Development"*, pages 49–56, Vienna, Austria.

King, Tracy Holloway, Frost, Martin, Kuhn, Jonas and Butt, Miriam. 2005. The Feature Space in Parallel Grammar Writing. *Research on Language and Computation* 3, 139–163.

Kordoni, Valia and Neu, Julia. 2005. Deep Analysis of Modern Greek. *Lecture Notes in Computer Science* 3248, 674–683.

Krieger, Hans-Ulrich and Schäfer, Ulrich. 1994. TDL – A Type Description Language for Constraint-Based Grammars. In *Proceedings of the 15th International Conference on Computational Linguistics*, pages 893–899, Kyoto, Japan.

Kroeger, Paul. 1993. *Phrase Structure and Grammatical Relations in Tagalog*. Stanford: CSLI.

Levin, Lori S. and Simpson, Jane. 1981. Quirky Case and Lexical Representations of Icelandic Verbs. *Chicago Linguistic Society* 17, 185–196.

Lønning, Jan Tore, Oepen, Stephan, Beermann, Dorothee, Hellan, Lars, Carroll, John, Dyvik, Helge, Flickinger, Dan, Johannessen, Janne Bondi, Meurer, Paul, Nordgård, Torbjørn, Rosén, Victoria and Velldal, Erik. 2004. LOGON. A Norwegian MT Effort. In *Proceedings of the Workshop in Recent Advances in Scandinavian Machine Translation*, Uppsala, Sweden.

Maddieson, Ian. 1980. Phonological Generalizations from the UCLA Phonological Segment Inventory Database. *UCLA Working Papers in Phonetics* 50, 57–68.

Makino, Seiichi and Tsutsui, Michio. 1989. *A Dictionary of Basic Japanese Grammar*. Tokyo: The Japan Times.

Manning, Christopher D. 1996. *Ergativity*. Stanford: CSLI.

Manning, Christopher D. and Sag, Ivan. 1995. Dissociations between Argument Structure and Grammatical Relations. In *Lexical and Constructional Aspects of Linguistic Explanation*, pages 63–78, CSLI.

Marimon, Montserrat, Bel, Núria and Seghezzi, Natalia. 2007. Test-suite Construction for a Spanish Grammar. In Tracy Holloway King and Emily M. Bender (eds.), *Proceedings of the GEAF07 Workshop*, pages 224–237, Stanford: CSLI.

McDonald, M. and Wurm, S. A. 1979. *Basic Materials in Wankumara (Galali): Grammar, Sentences and Vocabulary*. Canberra, Australia: Pacific Linguistics.

Meurers, W. Detmar, Penn, Gerald and Richter, Frank. 2002. A Web-based Instructional Platform for Constraint-Based Grammar Formalisms and Parsing. In Dragomir Radev and Chris Brew (eds.), *Effective Tools and Methodologies for Teaching NLP and CL*, pages 18–25, New Brunswick, NJ: The Association for Computational Linguistics.

Mixco, Mauricio. 1997. *Mandan*. Munich: LINCOM EUROPA.

Mohanan, Tara. 1994. *Argument Structure in Hindi*. Stanford: CSLI.

Moravcsik, Edith. 1978. Agreement. In Joseph H. Greenberg, Charles A. Ferguson and Edith A. Moravcsik (eds.), *Universals of Human Language, Vol. 4: Syntax*, pages 331–374, Stanford: Stanford University Press.

Nichols, Johanna. 2007. What, If Anything, Is Typology? *Linguistic Typology* 11, 231–8.

Nirenburg, Sergei. 1998. Project Boas: "A Linguist in the Box" as a Multi-Purpose Language Resource. In *Proceedings of The First Lexical Resources and Evaluation Conference*, Granada, Spain.

O'Hara, Kelly. 2008. *A Morphotactic Infrastructure for a Grammar Customization System*. Masters Thesis, University of Washington.

Plank, Frans and Filimonova, Elena. 2000. The Universals Archive: A Brief Introduction for Prospective Users. *Sprachtypologie und Universalienforschung* 53, 109–123.

Pollard, Carl and Sag, Ivan A. 1994. *Head-Driven Phrase Structure Grammar*. Stanford: CSLI.

Poulson, Laurie. forthcoming. *Grammatical Tense and Aspect: A Cross-Linguistic Implementation*. Ph. D.thesis, University of Washington.

Probst, Katharina, Levin, Lori, Peterson, Erik, Lavie, Alon and Carbonell, Jaime. 2002. MT for Minority Languages Using Elicitation-Based Learning of Syntactic Transfer Rules. *Machine Translation* 17(4).

Rigsby, Bruce and Rude, Noel. 1996. Sketch of Sahaptin, a Sahaptian Language. In Ives Goddard (ed.), *Languages*, pages 666–92, Washington DC: Smithsonian Institution.

Rude, Noel. 1996. *Umatilla Sahaptin Dictionary*. Manuscript.

Sag, Ivan A., Wasow, Thomas and Bender, Emily M. 2003. *Syntactic Theory: A Formal Introduction*. Stanford: CSLI.

Schachter, Paul and Otanes, Fe T. 1972. *Tagalog Reference Grammar*. Berkeley: University of California Press.

Scott, Graham. 1978. *The Fore Language of Papua New Guinea*. Canberra, Australia: Pacific Linguistics.

Siegel, Melanie and Bender, Emily M. 2002. Efficient Deep Processing of Japanese. In *Proceedings of the 3rd Workshop on Asian Language Resources and Standardization at the 19th International Conference on Computational Linguistics*, Taipei, Taiwan.

Siewierska, Anna. 2004. *Person*. Cambridge: Cambridge University Press.

Smrž, Pavel. 2005. A Parallel Metagrammar for Closely Related Languages – A Case Study of Czech and Russian. *Research on Language and Computation* 3, 165–189.

Valentine, J. Randolph. 2001. *Nishnaabemwin Reference Grammar*. Toronto: University of Toronto Press.

Vihman, Marilyn. 1977. *A Reference Manual and User's Guide for the Stanford Phonology Archive, Part I*. Stanford University.

Appendix A
GERMAN CHOICES

```
version=17

section=general
language=German
iso-code=deu

section=word-order
word-order=svo
has-dets=yes
noun-det-order=det-noun
has-aux=no

section=number

section=person
person=none

section=gender
  gender1_name=masc
  gender2_name=fem
  gender3_name=neut

section=case
case-marking=nom-acc
nom-acc-nom-case-name=nom
nom-acc-acc-case-name=acc
  case1_name=dat

section=direct-inverse

section=tense-aspect

section=other-features
```

```
section=sentential-negation

section=coordination

section=matrix-yes-no

section=lexicon
  noun1_name=masculine
    noun1_feat1_name=gender
    noun1_feat1_value=masc
  noun1_det=obl
    noun1_stem1_orth=Mann
    noun1_stem1_pred=_mann_n_rel
  noun2_name=weak-masculine
    noun2_feat1_name=gender
    noun2_feat1_value=masc
  noun2_det=obl
    noun2_stem1_orth=Mensch
    noun2_stem1_pred=_mensch_n_rel
  noun3_name=feminine
    noun3_feat1_name=gender
    noun3_feat1_value=fem
  noun3_det=obl
    noun3_stem1_orth=Frau
    noun3_stem1_pred=_frau_n_rel
  noun4_name=neuter
    noun4_feat1_name=gender
    noun4_feat1_value=neut
  noun4_det=obl
    noun4_stem1_orth=Mädchen
    noun4_stem1_pred=_mädchen_n_rel
  noun-slot1_name=weak-acc
  noun-slot1_order=after
    noun-slot1_input1_type=noun2
    noun-slot1_morph1_orth=en
      noun-slot1_morph1_feat1_name=case
      noun-slot1_morph1_feat1_value=acc
      noun-slot1_morph2_feat1_name=case
      noun-slot1_morph2_feat1_value=nom, dat
  verb1_name=nominative
```

```
verb1_valence=nom
  verb1_stem1_orth=schläft
  verb1_stem1_pred=_schlafen_v_rel
verb2_name=accusative
verb2_valence=nom-acc
  verb2_stem1_orth=sieht
  verb2_stem1_pred=_sehen_v_rel
verb3_name=dative
  verb3_feat1_name=case
  verb3_feat1_value=nom
  verb3_feat1_head=subj
  verb3_feat2_name=case
  verb3_feat2_value=dat
  verb3_feat2_head=obj
verb3_valence=trans
  verb3_stem1_orth=hilft
  verb3_stem1_pred=_helfen_v_rel
  det1_stem1_orth=der
  det1_stem1_pred=exist_q_rel
  det1_feat1_name=case
  det1_feat1_value=nom
  det1_feat2_name=gender
  det1_feat2_value=masc
  det2_stem1_orth=die
  det2_stem1_pred=exist_q_rel
  det2_feat1_name=case
  det2_feat1_value=nom, acc
  det2_feat2_name=gender
  det2_feat2_value=fem
  det3_stem1_orth=das
  det3_stem1_pred=exist_q_rel
  det3_feat1_name=case
  det3_feat1_value=nom, acc
  det3_feat2_name=gender
  det3_feat2_value=neut
  det4_stem1_orth=den
  det4_stem1_pred=exist_q_rel
  det4_feat1_name=case
  det4_feat1_value=acc
  det4_feat2_name=gender
  det4_feat2_value=masc
```

```
    det5_stem1_orth=dem
    det5_stem1_pred=exist_q_rel
    det5_feat1_name=case
    det5_feat1_value=dat
    det5_feat2_name=gender
    det5_feat2_value=masc, neut
    det6_stem1_orth=der
    det6_stem1_pred=exist_q_rel
    det6_feat1_name=case
    det6_feat1_value=dat
    det6_feat2_name=gender
    det6_feat2_value=fem

section=test-sentences
```

Appendix B
DYIRBAL CHOICES

```
version=17

section=general
language=Dyirbal
iso-code=dbl

section=word-order
word-order=sov
has-dets=yes
noun-det-order=det-noun
has-aux=no

section=number

section=person
person=1-2-3
first-person=none

section=gender
  gender1_name=I
  gender2_name=II

section=case
case-marking=split-n
split-n-nom-case-name=nom
split-n-acc-case-name=acc
split-n-erg-case-name=erg
split-n-abs-case-name=abs

section=direct-inverse

section=tense-aspect
```

```
section=other-features

section=sentential-negation

section=coordination

section=matrix-yes-no

section=lexicon
    noun1_feat1_name=gender
    noun1_feat1_value=I
    noun1_feat2_name=case
    noun1_feat2_value=abs
    noun1_feat3_name=person
    noun1_feat3_value=3rd
  noun1_det=obl
    noun1_stem1_orth=yaɽa
    noun1_stem1_pred=_yaɽa_n_rel
    noun2_feat1_name=gender
    noun2_feat1_value=I
    noun2_feat2_name=case
    noun2_feat2_value=erg
    noun2_feat3_name=person
    noun2_feat3_value=3rd
  noun2_det=obl
    noun2_stem1_orth=yaɽaŋgu
    noun2_stem1_pred=_yaɽa_n_rel
    noun3_feat1_name=gender
    noun3_feat1_value=II
    noun3_feat2_name=case
    noun3_feat2_value=abs
    noun3_feat3_name=person
    noun3_feat3_value=3rd
  noun3_det=obl
    noun3_stem1_orth=ɖugumbil
    noun3_stem1_pred=_ɖugumbil_n_rel
    noun4_feat1_name=gender
    noun4_feat1_value=II
    noun4_feat2_name=case
    noun4_feat2_value=erg
    noun4_feat3_name=person
    noun4_feat3_value=3rd
  noun4_det=obl
    noun4_stem1_orth=ɖugumbiɽu
    noun4_stem1_pred=_ɖugumbil_n_rel
    noun5_feat1_name=case
    noun5_feat1_value=nom
    noun5_feat2_name=person
    noun5_feat2_value=1st
  noun5_det=imp
    noun5_stem1_orth=ŋaɖa
    noun5_stem1_pred=_pronoun_n_rel
    noun6_feat1_name=case
    noun6_feat1_value=acc
    noun6_feat2_name=person
    noun6_feat2_value=1st
  noun6_det=imp
    noun6_stem1_orth=ŋayguna
    noun6_stem1_pred=_pronoun_n_rel
    noun7_feat1_name=case
    noun7_feat1_value=nom
    noun7_feat2_name=person
    noun7_feat2_value=2nd
  noun7_det=imp
    noun7_stem1_orth=ŋinda
    noun7_stem1_pred=_pronoun_n_rel
    noun8_feat1_name=case
    noun8_feat1_value=acc
    noun8_feat2_name=person
    noun8_feat2_value=2nd
  noun8_det=imp
    noun8_stem1_orth=ŋinuna
    noun8_stem1_pred=_pronoun_n_rel
  verb1_valence=s
    verb1_stem1_orth=baniɲu
    verb1_stem1_pred=_baniɲu_v_rel
  verb2_valence=a-o
    verb2_stem1_orth=balgan
    verb2_stem1_pred=_balgan_v_rel
    det1_stem1_orth=bayi
    det1_stem1_pred=exist_q_rel
    det1_feat1_name=gender
    det1_feat1_value=I
    det1_feat2_name=case
    det1_feat2_value=abs
    det2_stem1_orth=baŋgul
    det2_stem1_pred=exist_q_rel
    det2_feat1_name=gender
    det2_feat1_value=I
    det2_feat2_name=case
    det2_feat2_value=erg
    det3_stem1_orth=balan
    det3_stem1_pred=exist_q_rel
    det3_feat1_name=gender
    det3_feat1_value=II
    det3_feat2_name=case
    det3_feat2_value=abs
    det4_stem1_orth=baŋgun
    det4_stem1_pred=exist_q_rel
    det4_feat1_name=gender
    det4_feat1_value=II
    det4_feat2_name=case
    det4_feat2_value=erg

section=test-sentences
```

Appendix C

HINDI CHOICES

```
version=17

section=general
language=Hindi
iso-code=hin

section=word-order
word-order=sov
has-dets=no
has-aux=no

section=number

section=person
person=none

section=gender

section=case
case-marking=split-v
split-v-nom-case-name=nom
split-v-acc-case-name=acc
split-v-erg-case-name=erg
split-v-abs-case-name=abs

section=direct-inverse

section=tense-aspect
tense-definition=build
  tense1_name=future
    tense1_supertype1_name=tense
  tense2_name=perfective
    tense2_supertype1_name=tense
```

```
section=other-features

section=sentential-negation

section=coordination

section=matrix-yes-no

section=lexicon
  noun1_name=name
  noun1_det=imp
    noun1_stem1_orth=raam
    noun1_stem1_pred=_raam_n_rel
    noun1_stem2_orth=ravii
    noun1_stem2_pred=_ravii_n_rel
  noun-slot1_name=case
  noun-slot1_order=after
    noun-slot1_input1_type=noun
      noun-slot1_morph1_feat1_name=case
      noun-slot1_morph1_feat1_value=nom
    noun-slot1_morph2_orth=ne
      noun-slot1_morph2_feat1_name=case
      noun-slot1_morph2_feat1_value=erg
    noun-slot1_morph3_orth=ko
      noun-slot1_morph3_feat1_name=case
      noun-slot1_morph3_feat1_value=acc
  verb1_name=nom
    verb1_feat1_name=case
    verb1_feat1_value=nom
    verb1_feat1_head=subj
  verb1_valence=intrans
    verb1_stem1_orth=gir
    verb1_stem1_pred=_gir_v_rel
  verb2_name=erg
    verb2_feat1_name=case
    verb2_feat1_value=erg
    verb2_feat1_head=subj
  verb2_valence=intrans
    verb2_stem1_orth=cʰĩĩk
    verb2_stem1_pred=_cʰĩĩk_v_rel
  verb3_name=fluid
    verb3_feat1_name=case
    verb3_feat1_value=nom, erg
    verb3_feat1_head=subj
  verb3_valence=intrans
    verb3_stem1_orth=naac
    verb3_stem1_pred=_naac_v_rel
    verb4_feat1_name=case
    verb4_feat1_value=acc
    verb4_feat1_head=obj
  verb4_valence=trans
    verb4_stem1_orth=piiṭ
    verb4_stem1_pred=_piiṭ_v_rel
  verb-slot1_order=after
    verb-slot1_input1_type=iverb
    verb-slot1_morph1_orth=aa
      verb-slot1_morph1_feat1_name=tense
      verb-slot1_morph1_feat1_value=perfective
      verb-slot1_morph1_feat1_head=verb
    verb-slot1_morph2_orth=egaa
      verb-slot1_morph2_feat1_name=tense
      verb-slot1_morph2_feat1_value=future
      verb-slot1_morph2_feat1_head=verb
  verb-slot2_order=after
    verb-slot2_input1_type=tverb
    verb-slot2_morph1_orth=aa
      verb-slot2_morph1_feat1_name=tense
      verb-slot2_morph1_feat1_value=perfective
      verb-slot2_morph1_feat1_head=verb
      verb-slot2_morph1_feat2_name=case
      verb-slot2_morph1_feat2_value=erg
      verb-slot2_morph1_feat2_head=subj
    verb-slot2_morph2_orth=egaa
      verb-slot2_morph2_feat1_name=tense
      verb-slot2_morph2_feat1_value=future
      verb-slot2_morph2_feat1_head=verb
      verb-slot2_morph2_feat2_name=case
      verb-slot2_morph2_feat2_value=nom
      verb-slot2_morph2_feat2_head=subj

section=test-sentences
```

Appendix D

TAGALOG CHOICES

```
version=17

section=general
language=Tagalog
iso-code=tgl

section=word-order
word-order=v-initial
has-dets=no
has-aux=no

section=number

section=person
person=none

section=gender

section=case
case-marking=focus
focus-focus-case-name=foc
focus-a-case-name=a
focus-o-case-name=o

section=direct-inverse

section=tense-aspect

section=other-features

section=sentential-negation

section=coordination
```

```
section=matrix-yes-no

section=lexicon
  noun1_name=common
  noun1_det=imp
    noun1_stem1_orth=babae
    noun1_stem1_pred=_babae_n_rel
    noun1_stem2_orth=lalaki
    noun1_stem2_pred=_lalaki_n_rel
  verb1_valence=intrans
    verb1_stem1_orth=tulog
    verb1_stem1_pred=_tulog_v_rel
  verb2_valence=trans
    verb2_stem1_orth=kita
    verb2_stem1_pred=_kita_v_rel
  verb-slot1_order=before
    verb-slot1_input1_type=iverb
    verb-slot1_morph1_orth=ma
      verb-slot1_morph1_feat1_name=argument structure
      verb-slot1_morph1_feat1_value=focus
      verb-slot1_morph1_feat1_head=verb
  verb-slot2_order=before
    verb-slot2_input1_type=tverb
    verb-slot2_morph1_orth=ma
      verb-slot2_morph1_feat1_name=argument structure
      verb-slot2_morph1_feat1_value=focus-o
      verb-slot2_morph1_feat1_head=verb
    verb-slot2_morph2_orth=maka
      verb-slot2_morph2_feat1_name=argument structure
      verb-slot2_morph2_feat1_value=a-focus
      verb-slot2_morph2_feat1_head=verb
  adp1_orth=ang
  adp1_order=before
    adp1_feat1_name=case
    adp1_feat1_value=focus
  adp2_orth=ng
  adp2_order=before
    adp2_feat1_name=case
    adp2_feat1_value=a, o
```

```
section=test-sentences
```

Appendix E
CREE CHOICES

```
version=17

section=general
language=Plains Cree
iso-code=crk

section=word-order
word-order=vso
has-dets=no
has-aux=no

section=number

section=person
person=1-2-3
first-person=none

section=gender

section=case
case-marking=none

section=direct-inverse
    scale1_feat1_name=person
    scale1_feat1_value=2nd
    scale2_feat1_name=person
    scale2_feat1_value=1st
    scale3_feat1_name=person
    scale3_feat1_value=3rd
    scale3_feat2_name=proximity
    scale3_feat2_value=proximate
    scale4_feat1_name=person
    scale4_feat1_value=3rd
    scale4_feat2_name=proximity
    scale4_feat2_value=obviative
scale-equal=other

section=tense-aspect

section=other-features
  feature1_name=proximity
  feature1_type=head
    feature1_value1_name=proximate
      feature1_value1_supertype1_name=proximity
    feature1_value2_name=obviative
      feature1_value2_supertype1_name=proximity

section=sentential-negation

section=coordination

section=matrix-yes-no

section=lexicon
    noun1_feat1_name=person
    noun1_feat1_value=1st
  noun1_det=imp
    noun1_stem1_orth=1
    noun1_stem1_pred=_pronoun_n_rel
    noun2_feat1_name=person
    noun2_feat1_value=2nd
  noun2_det=imp
    noun2_stem1_orth=2
    noun2_stem1_pred=_pronoun_n_rel
    noun3_feat1_name=person
    noun3_feat1_value=3rd
    noun3_feat2_name=proximity
    noun3_feat2_value=proximate
  noun3_det=imp
    noun3_stem1_orth=3
    noun3_stem1_pred=_pronoun_n_rel
    noun4_feat1_name=person
    noun4_feat1_value=3rd
    noun4_feat2_name=proximity
    noun4_feat2_value=obviative
  noun4_det=imp
    noun4_stem1_orth=obv
    noun4_stem1_pred=_pronoun_n_rel
  verb1_name=dummy
  verb1_valence=intrans
    verb1_stem1_orth=dummy
    verb1_stem1_pred=dummy
  verb2_name=transitive-animate
  verb2_valence=trans,dirinv
    verb2_stem1_orth=se·kih
    verb2_stem1_pred=_se·kih_v_rel
  verb-slot1_name=prefix
  verb-slot1_order=before
    verb-slot1_input1_type=verb2
    verb-slot1_morph1_orth=ki
      verb-slot1_morph1_feat1_name=person
      verb-slot1_morph1_feat1_value=2nd
      verb-slot1_morph1_feat1_head=higher
    verb-slot1_morph2_orth=ni
      verb-slot1_morph2_feat1_name=person
      verb-slot1_morph2_feat1_value=1st
      verb-slot1_morph2_feat1_head=higher
      verb-slot1_morph3_feat1_name=person
      verb-slot1_morph3_feat1_value=3rd
      verb-slot1_morph3_feat1_head=higher
      verb-slot1_morph3_feat2_name=proximity
      verb-slot1_morph3_feat2_value=proximate
      verb-slot1_morph3_feat2_head=higher
      verb-slot1_morph3_feat3_name=person
      verb-slot1_morph3_feat3_value=3rd
      verb-slot1_morph3_feat3_head=lower
      verb-slot1_morph3_feat4_name=proximity
      verb-slot1_morph3_feat4_value=obviative
      verb-slot1_morph3_feat4_head=lower
  verb-slot2_name=suffix
  verb-slot2_order=after
    verb-slot2_input1_type=verb-slot1
    verb-slot2_morph1_orth=in
      verb-slot2_morph1_feat1_name=direction
      verb-slot2_morph1_feat1_value=dir
  verb-slot2_morph1_feat1_head=verb
  verb-slot2_morph1_feat2_name=person
  verb-slot2_morph1_feat2_value=1st
  verb-slot2_morph1_feat2_head=obj
verb-slot2_morph2_orth=a·w
  verb-slot2_morph2_feat1_name=direction
  verb-slot2_morph2_feat1_value=dir
  verb-slot2_morph2_feat1_head=verb
  verb-slot2_morph2_feat2_name=person
  verb-slot2_morph2_feat2_value=3rd
  verb-slot2_morph2_feat2_head=obj
  verb-slot2_morph2_feat3_name=proximity
  verb-slot2_morph2_feat3_value=proximate
  verb-slot2_morph2_feat3_head=obj
verb-slot2_morph3_orth=e·w
  verb-slot2_morph3_feat1_name=direction
  verb-slot2_morph3_feat1_value=dir
  verb-slot2_morph3_feat1_head=verb
  verb-slot2_morph3_feat2_name=person
  verb-slot2_morph3_feat2_value=3rd
  verb-slot2_morph3_feat2_head=subj
  verb-slot2_morph3_feat3_name=proximity
  verb-slot2_morph3_feat3_value=proximate
  verb-slot2_morph3_feat3_head=subj
verb-slot2_morph4_orth=ima·wa
  verb-slot2_morph4_feat1_name=direction
  verb-slot2_morph4_feat1_value=dir
  verb-slot2_morph4_feat1_head=verb
  verb-slot2_morph4_feat2_name=person
  verb-slot2_morph4_feat2_value=1st, 2nd
  verb-slot2_morph4_feat2_head=subj
  verb-slot2_morph4_feat3_name=person
  verb-slot2_morph4_feat3_value=3rd
  verb-slot2_morph4_feat3_head=obj
  verb-slot2_morph4_feat4_name=proximity
  verb-slot2_morph4_feat4_value=obviative
  verb-slot2_morph4_feat4_head=obj
verb-slot2_morph5_orth=ikoyiwa
  verb-slot2_morph5_feat1_name=direction
  verb-slot2_morph5_feat1_value=inv
  verb-slot2_morph5_feat1_head=verb
```

```
    verb-slot2_morph5_feat2_name=person
    verb-slot2_morph5_feat2_value=3rd
    verb-slot2_morph5_feat2_head=subj
    verb-slot2_morph5_feat3_name=proximity
    verb-slot2_morph5_feat3_value=obviative
    verb-slot2_morph5_feat3_head=subj
    verb-slot2_morph5_feat4_name=person
    verb-slot2_morph5_feat4_value=1st, 2nd
    verb-slot2_morph5_feat4_head=obj
  verb-slot2_morph6_orth=ik
    verb-slot2_morph6_feat1_name=direction
    verb-slot2_morph6_feat1_value=inv
    verb-slot2_morph6_feat1_head=verb
    verb-slot2_morph6_feat2_name=person
    verb-slot2_morph6_feat2_value=3rd
    verb-slot2_morph6_feat2_head=subj
    verb-slot2_morph6_feat3_name=proximity
    verb-slot2_morph6_feat3_value=proximate
    verb-slot2_morph6_feat3_head=subj
  verb-slot2_morph7_orth=ik
    verb-slot2_morph7_feat1_name=direction
    verb-slot2_morph7_feat1_value=inv
    verb-slot2_morph7_feat1_head=verb
    verb-slot2_morph7_feat2_name=person
    verb-slot2_morph7_feat2_value=3rd
    verb-slot2_morph7_feat2_head=subj
    verb-slot2_morph7_feat3_name=proximity
    verb-slot2_morph7_feat3_value=obviative
    verb-slot2_morph7_feat3_head=subj
    verb-slot2_morph7_feat4_name=person
    verb-slot2_morph7_feat4_value=3rd
    verb-slot2_morph7_feat4_head=obj
    verb-slot2_morph7_feat5_name=proximity
    verb-slot2_morph7_feat5_value=proximate
    verb-slot2_morph7_feat5_head=obj
  verb-slot2_morph8_orth=itin
    verb-slot2_morph8_feat1_name=direction
    verb-slot2_morph8_feat1_value=inv
    verb-slot2_morph8_feat1_head=verb
    verb-slot2_morph8_feat2_name=person
    verb-slot2_morph8_feat2_value=1st
```

```
      verb-slot2_morph8_feat2_head=subj

section=test-sentences
```

Appendix F
FORE CHOICES

```
version=17

section=general
language=Fore
iso-code=for

section=word-order
word-order=v-final
has-dets=no
has-aux=no

section=number

section=person
person=none

section=gender
  gender1_name=human
    gender1_supertype1_name=gender
  gender2_name=animate
    gender2_supertype1_name=gender
  gender3_name=inanimate
    gender3_supertype1_name=gender

section=case
case-marking=erg-abs
erg-abs-erg-case-name=ergative
erg-abs-abs-case-name=nominative

section=direct-inverse
    scale1_feat1_name=ntype
    scale1_feat1_value=non-common
    scale2_feat1_name=ntype
    scale2_feat1_value=common
    scale2_feat2_name=gender
    scale2_feat2_value=human
    scale3_feat1_name=ntype
    scale3_feat1_value=common
    scale3_feat2_name=gender
    scale3_feat2_value=animate
    scale4_feat1_name=ntype
    scale4_feat1_value=common
    scale4_feat2_name=gender
    scale4_feat2_value=inanimate
scale-equal=direct

section=tense-aspect

section=other-features
  feature1_name=ntype
  feature1_type=head
    feature1_value1_name=non-common
      feature1_value1_supertype1_name=ntype
    feature1_value2_name=pronoun
      feature1_value2_supertype1_name=non-common
    feature1_value3_name=name
      feature1_value3_supertype1_name=non-common
    feature1_value4_name=kin
      feature1_value4_supertype1_name=non-common
    feature1_value5_name=common
      feature1_value5_supertype1_name=ntype

section=sentential-negation

section=coordination

section=matrix-yes-no

section=lexicon
    noun1_feat1_name=ntype
    noun1_feat1_value=pronoun
    noun1_feat2_name=gender
    noun1_feat2_value=human
  noun1_det=imp
```

```
    noun1_stem1_orth=ae
    noun1_stem1_pred=_pronoun_n_rel
    noun2_feat1_name=ntype
    noun2_feat1_value=kin
    noun2_feat2_name=gender
    noun2_feat2_value=human
  noun2_det=imp
    noun2_stem1_orth=naba:
    noun2_stem1_pred=_naba:_n_rel
    noun3_feat1_name=ntype
    noun3_feat1_value=name
    noun3_feat2_name=gender
    noun3_feat2_value=human
  noun3_det=imp
    noun3_stem1_orth=ayore
    noun3_stem1_pred=_ayore_n_rel
    noun4_feat1_name=ntype
    noun4_feat1_value=common
    noun4_feat2_name=gender
    noun4_feat2_value=human
  noun4_det=imp
    noun4_stem1_orth=wá
    noun4_stem1_pred=_wá_n_rel
    noun5_feat1_name=ntype
    noun5_feat1_value=common
    noun5_feat2_name=gender
    noun5_feat2_value=animate
  noun5_det=imp
    noun5_stem1_orth=yaga:
    noun5_stem1_pred=_yaga:_n_rel
    noun6_feat1_name=ntype
    noun6_feat1_value=common
    noun6_feat2_name=gender
    noun6_feat2_value=inanimate
  noun6_det=imp
    noun6_stem1_orth=naninta:
    noun6_stem1_pred=_naninta:_n_rel
  noun-slot1_name=case
  noun-slot1_order=after
    noun-slot1_input1_type=noun
    noun-slot1_morph1_orth=ma
    noun-slot1_morph1_feat1_name=case
    noun-slot1_morph1_feat1_value=erg
    noun-slot1_morph1_feat2_name=gender
    noun-slot1_morph1_feat2_value=human
  noun-slot1_morph2_orth=wama
    noun-slot1_morph2_feat1_name=case
    noun-slot1_morph2_feat1_value=erg
    noun-slot1_morph2_feat2_name=gender
    noun-slot1_morph2_feat2_value=animate, inanimate
    noun-slot1_morph3_feat1_name=case
    noun-slot1_morph3_feat1_value=abs
verb1_valence=abs
  verb1_stem1_orth=kanaye
  verb1_stem1_pred=_kana_v_rel
verb2_valence=trans,dirinv
  verb2_stem1_orth=agaye
  verb2_stem1_pred=_ga_v_rel
verb-slot1_name=direction
verb-slot1_order=after
  verb-slot1_input1_type=verb2
  verb-slot1_morph1_name=direct
    verb-slot1_morph1_feat1_name=direction
    verb-slot1_morph1_feat1_value=dir
    verb-slot1_morph1_feat1_head=verb
    verb-slot1_morph1_feat2_name=case
    verb-slot1_morph1_feat2_value=abs
    verb-slot1_morph1_feat2_head=subj
    verb-slot1_morph1_feat3_name=case
    verb-slot1_morph1_feat3_value=abs
    verb-slot1_morph1_feat3_head=obj
  verb-slot1_morph2_name=inverse
    verb-slot1_morph2_feat1_name=direction
    verb-slot1_morph2_feat1_value=inv
    verb-slot1_morph2_feat1_head=verb
    verb-slot1_morph2_feat2_name=case
    verb-slot1_morph2_feat2_value=erg
    verb-slot1_morph2_feat2_head=subj
    verb-slot1_morph2_feat3_name=case
    verb-slot1_morph2_feat3_value=abs
    verb-slot1_morph2_feat3_head=obj
```

section=test-sentences

Appendix G

SAHAPTIN CHOICES

```
version=16

section=general
language=Umatilla Sahaptin
iso-code=uma

section=word-order
word-order=vso
has-dets=no
has-aux=no

section=number
  number1_name=sg
  number2_name=du
  number3_name=pl

section=person
person=1-2-3
first-person=incl-excl
incl-excl-number=du, pl

section=gender

section=case
case-marking=nom-acc
nom-acc-nom-case-name=nom
nom-acc-acc-case-name=obj

section=direct-inverse
    scale1_feat1_name=pernum
    scale1_feat1_value=1st
    scale2_feat1_name=pernum
    scale2_feat1_value=2nd
    scale3_feat1_name=pernum
    scale3_feat1_value=3rd
    scale3_feat2_name=topicality
    scale3_feat2_value=topic
    scale4_feat1_name=pernum
    scale4_feat1_value=3rd
    scale4_feat2_name=topicality
    scale4_feat2_value=non-topic
scale-equal=direct

section=tense-aspect

section=other-features
  feature1_name=topicality
  feature1_type=index
    feature1_value1_name=topic
      feature1_value1_supertype1_name=topicality
    feature1_value2_name=non-topic
      feature1_value2_supertype1_name=topicality
  feature2_name=proximity
  feature2_type=head
    feature2_value1_name=proximate
      feature2_value1_supertype1_name=proximity
    feature2_value2_name=obviative
      feature2_value2_supertype1_name=proximity
    feature2_value3_name=neutral
      feature2_value3_supertype1_name=proximity

section=sentential-negation

section=coordination

section=matrix-yes-no

section=lexicon
  noun1_name=1sg-pronoun
    noun1_feat1_name=case
    noun1_feat1_value=nom
    noun1_feat2_name=pernum
    noun1_feat2_value=1sg
    noun1_feat3_name=proximity
  noun1_feat3_value=neutral
noun1_det=imp
  noun1_stem1_orth=ín
  noun1_stem1_pred=_pronoun_n_rel
noun2_name=2sg-pronoun
  noun2_feat1_name=case
  noun2_feat1_value=nom
  noun2_feat2_name=pernum
  noun2_feat2_value=2sg
noun2_det=imp
  noun2_stem1_orth=ím
  noun2_stem1_pred=_pronoun_n_rel
noun3_name=3sg-pronoun
  noun3_feat1_name=case
  noun3_feat1_value=nom
  noun3_feat2_name=pernum
  noun3_feat2_value=3sg
  noun3_feat3_name=proximity
  noun3_feat3_value=neutral
noun3_det=imp
  noun3_stem1_orth=pɨ́n
  noun3_stem1_pred=_pronoun_n_rel
noun4_name=3sgobv-pronoun
  noun4_feat1_name=case
  noun4_feat1_value=nom
  noun4_feat2_name=pernum
  noun4_feat2_value=3sg
  noun4_feat3_name=proximity
  noun4_feat3_value=obviative
noun4_det=imp
  noun4_stem1_orth=piiní
  noun4_stem1_pred=_pronoun_n_rel
noun5_name=3sginv-pronoun
  noun5_feat1_name=case
  noun5_feat1_value=nom
  noun5_feat2_name=pernum
  noun5_feat2_value=3sg
  noun5_feat3_name=proximity
  noun5_feat3_value=proximate
noun5_det=imp
  noun5_stem1_orth=pnɨ́m
```

```
    noun5_stem1_pred=_pronoun_n_rel
  noun6_name=1du-pronoun
    noun6_feat1_name=case
    noun6_feat1_value=nom
    noun6_feat2_name=pernum
    noun6_feat2_value=1du
    noun6_feat3_name=proximity
    noun6_feat3_value=neutral
  noun6_det=imp
    noun6_stem1_orth=napiiní
    noun6_stem1_pred=_pronoun_n_rel
  noun7_name=2du-pronoun
    noun7_feat1_name=case
    noun7_feat1_value=nom
    noun7_feat2_name=pernum
    noun7_feat2_value=2du
    noun7_feat3_name=proximity
    noun7_feat3_value=neutral
  noun7_det=imp
    noun7_stem1_orth=imiiní
    noun7_stem1_pred=_pronoun_n_rel
  noun8_name=3du-pronoun
    noun8_feat1_name=case
    noun8_feat1_value=nom
    noun8_feat2_name=pernum
    noun8_feat2_value=3du
    noun8_feat3_name=proximity
    noun8_feat3_value=neutral
  noun8_det=imp
    noun8_stem1_orth=piiní
    noun8_stem1_pred=_pronoun_n_rel
  noun9_name=1pl-pronoun
    noun9_feat1_name=case
    noun9_feat1_value=nom
    noun9_feat2_name=pernum
    noun9_feat2_value=1pl
    noun9_feat3_name=proximity
    noun9_feat3_value=neutral
  noun9_det=imp
    noun9_stem1_orth=náma
    noun9_stem1_pred=_pronoun_n_rel
```

```
noun10_name=2pl-pronoun
  noun10_feat1_name=case
  noun10_feat1_value=nom
  noun10_feat2_name=pernum
  noun10_feat2_value=2pl
  noun10_feat3_name=proximity
  noun10_feat3_value=neutral
noun10_det=imp
  noun10_stem1_orth=imáy
  noun10_stem1_pred=_pronoun_n_rel
noun11_name=3pl-pronoun
  noun11_feat1_name=case
  noun11_feat1_value=nom
  noun11_feat2_name=pernum
  noun11_feat2_value=3pl
  noun11_feat3_name=proximity
  noun11_feat3_value=neutral
noun11_det=imp
  noun11_stem1_orth=pmáy
  noun11_stem1_pred=_pronoun_n_rel
noun12_name=1sgobj-pronoun
  noun12_feat1_name=case
  noun12_feat1_value=acc
  noun12_feat2_name=pernum
  noun12_feat2_value=1sg
noun12_det=imp
  noun12_stem1_orth=ináy
  noun12_stem1_pred=_pronoun_n_rel
noun13_name=2sgobj-pronoun
  noun13_feat1_name=case
  noun13_feat1_value=acc
  noun13_feat2_name=pernum
  noun13_feat2_value=2sg
noun13_det=imp
  noun13_stem1_orth=imanáy
  noun13_stem1_pred=_pronoun_n_rel
noun14_name=3sgobj-pronoun
  noun14_feat1_name=case
  noun14_feat1_value=acc
  noun14_feat2_name=pernum
  noun14_feat2_value=3sg
```

```
noun14_det=imp
  noun14_stem1_orth=paanáy
  noun14_stem1_pred=_pronoun_n_rel
noun15_name=1duobj-pronoun
  noun15_feat1_name=case
  noun15_feat1_value=acc
  noun15_feat2_name=pernum
  noun15_feat2_value=1du
noun15_det=imp
  noun15_stem1_orth=napiinamanáy
  noun15_stem1_pred=_pronoun_n_rel
noun16_name=2duobj-pronoun
  noun16_feat1_name=case
  noun16_feat1_value=acc
  noun16_feat2_name=pernum
  noun16_feat2_value=2du
noun16_det=imp
  noun16_stem1_orth=imiinamanáy
  noun16_stem1_pred=_pronoun_n_rel
noun17_name=3duobj-pronoun
  noun17_feat1_name=case
  noun17_feat1_value=acc
  noun17_feat2_name=pernum
  noun17_feat2_value=3du
noun17_det=imp
  noun17_stem1_orth=piinamanáy
  noun17_stem1_pred=_pronoun_n_rel
noun18_name=1plobj-pronoun
  noun18_feat1_name=case
  noun18_feat1_value=acc
  noun18_feat2_name=pernum
  noun18_feat2_value=1pl
noun18_det=imp
  noun18_stem1_orth=naamanáy
  noun18_stem1_pred=_pronoun_n_rel
noun19_name=2plobj-pronoun
  noun19_feat1_name=case
  noun19_feat1_value=acc
  noun19_feat2_name=pernum
  noun19_feat2_value=2pl
noun19_det=imp
```

```
  noun19_stem1_orth=imaamanáy
  noun19_stem1_pred=_pronoun_n_rel
noun20_name=3plobj-pronoun
  noun20_feat1_name=case
  noun20_feat1_value=acc
  noun20_feat2_name=pernum
  noun20_feat2_value=3pl
noun20_det=imp
  noun20_stem1_orth=paamanáy
  noun20_stem1_pred=_pronoun_n_rel
noun-slot1_name=topic
noun-slot1_order=after
  noun-slot1_input1_type=noun
  noun-slot1_morph1_orth=-TOP
    noun-slot1_morph1_feat1_name=pernum
    noun-slot1_morph1_feat1_value=3rd
    noun-slot1_morph1_feat2_name=topicality
    noun-slot1_morph1_feat2_value=topic
    noun-slot1_morph2_feat1_name=pernum
    noun-slot1_morph2_feat1_value=3rd
    noun-slot1_morph2_feat2_name=topicality
    noun-slot1_morph2_feat2_value=non-topic
    noun-slot1_morph3_feat1_name=pernum
    noun-slot1_morph3_feat1_value=1st, 2nd
    noun-slot1_morph3_feat2_name=topicality
    noun-slot1_morph3_feat2_value=non-topic
verb1_valence=nom
  verb1_stem1_orth=wína
  verb1_stem1_pred=_wína_v_rel
verb2_valence=nom-acc,dirinv
  verb2_stem1_orth=q̓ínun
  verb2_stem1_pred=_q̓ínun_v_rel
verb-slot1_name=intrans-prefix
verb-slot1_order=before
  verb-slot1_input1_type=verb1
  verb-slot1_morph1_orth=i-
    verb-slot1_morph1_feat1_name=pernum
    verb-slot1_morph1_feat1_value=3sg
    verb-slot1_morph1_feat1_head=subj
    verb-slot1_morph1_feat2_name=proximity
    verb-slot1_morph1_feat2_value=neutral
```

```
      verb-slot1_morph1_feat2_head=subj
    verb-slot1_morph2_orth=pa-
      verb-slot1_morph2_feat1_name=pernum
      verb-slot1_morph2_feat1_value=3du, 3pl
      verb-slot1_morph2_feat1_head=subj
      verb-slot1_morph3_feat1_name=pernum
      verb-slot1_morph3_feat1_value=1st, 2nd
      verb-slot1_morph3_feat1_head=subj
  verb-slot2_name=trans-prefix
  verb-slot2_order=before
    verb-slot2_input1_type=verb2
    verb-slot2_morph1_orth=i-
      verb-slot2_morph1_feat1_name=pernum
      verb-slot2_morph1_feat1_value=3sg
      verb-slot2_morph1_feat1_head=subj
      verb-slot2_morph1_feat2_name=topicality
      verb-slot2_morph1_feat2_value=non-topic
      verb-slot2_morph1_feat2_head=obj
    verb-slot2_morph2_orth=pa-
      verb-slot2_morph2_feat1_name=pernum
      verb-slot2_morph2_feat1_value=3du, 3pl
      verb-slot2_morph2_feat1_head=subj
      verb-slot2_morph2_feat2_name=topicality
      verb-slot2_morph2_feat2_value=non-topic
      verb-slot2_morph2_feat2_head=obj
    verb-slot2_morph3_orth=á-
      verb-slot2_morph3_feat1_name=pernum
      verb-slot2_morph3_feat1_value=1st, 2nd
      verb-slot2_morph3_feat1_head=subj
      verb-slot2_morph3_feat2_name=pernum
      verb-slot2_morph3_feat2_value=3rd
      verb-slot2_morph3_feat2_head=obj
    verb-slot2_morph4_orth=pá-
      verb-slot2_morph4_feat1_name=pernum
      verb-slot2_morph4_feat1_value=2sg
      verb-slot2_morph4_feat1_head=subj
      verb-slot2_morph4_feat2_name=pernum
      verb-slot2_morph4_feat2_value=1sg
      verb-slot2_morph4_feat2_head=obj
    verb-slot2_morph5_orth=pá-
      verb-slot2_morph5_feat1_name=pernum
  verb-slot2_morph5_feat1_value=3sg
  verb-slot2_morph5_feat1_head=subj
  verb-slot2_morph5_feat2_name=pernum
  verb-slot2_morph5_feat2_value=3sg
  verb-slot2_morph5_feat2_head=obj
  verb-slot2_morph5_feat3_name=topicality
  verb-slot2_morph5_feat3_value=non-topic
  verb-slot2_morph5_feat3_head=subj
  verb-slot2_morph5_feat4_name=topicality
  verb-slot2_morph5_feat4_value=topic
  verb-slot2_morph5_feat4_head=obj
verb-slot2_morph6_orth=patá-
  verb-slot2_morph6_feat1_name=pernum
  verb-slot2_morph6_feat1_value=3du, 3pl
  verb-slot2_morph6_feat1_head=subj
  verb-slot2_morph6_feat2_name=pernum
  verb-slot2_morph6_feat2_value=3sg
  verb-slot2_morph6_feat2_head=obj
  verb-slot2_morph6_feat3_name=topicality
  verb-slot2_morph6_feat3_value=non-topic
  verb-slot2_morph6_feat3_head=subj
  verb-slot2_morph6_feat4_name=topicality
  verb-slot2_morph6_feat4_value=topic
  verb-slot2_morph6_feat4_head=obj
  verb-slot2_morph7_feat1_name=pernum
  verb-slot2_morph7_feat1_value=1st
  verb-slot2_morph7_feat1_head=subj
  verb-slot2_morph7_feat2_name=pernum
  verb-slot2_morph7_feat2_value=2nd
  verb-slot2_morph7_feat2_head=obj
  verb-slot2_morph8_feat1_name=pernum
  verb-slot2_morph8_feat1_value=2du, 2pl
  verb-slot2_morph8_feat1_head=subj
  verb-slot2_morph8_feat2_name=pernum
  verb-slot2_morph8_feat2_value=1st
  verb-slot2_morph8_feat2_head=obj
  verb-slot2_morph9_feat1_name=pernum
  verb-slot2_morph9_feat1_value=2sg
  verb-slot2_morph9_feat1_head=subj
  verb-slot2_morph9_feat2_name=pernum
  verb-slot2_morph9_feat2_value=1du, 1pl
```

```
      verb-slot2_morph9_feat2_head=obj
  verb-slot3_name=intrans-enclitic
  verb-slot3_order=after
    verb-slot3_input1_type=verb-slot1
    verb-slot3_morph1_orth==naš
      verb-slot3_morph1_feat1_name=pernum
      verb-slot3_morph1_feat1_value=1sg
      verb-slot3_morph1_feat1_head=subj
    verb-slot3_morph2_orth==na
      verb-slot3_morph2_feat1_name=pernum
      verb-slot3_morph2_feat1_value=1du_incl, 1pl_incl
      verb-slot3_morph2_feat1_head=subj
    verb-slot3_morph3_orth==nataš
      verb-slot3_morph3_feat1_name=pernum
      verb-slot3_morph3_feat1_value=1du_excl, 1pl_excl
      verb-slot3_morph3_feat1_head=subj
    verb-slot3_morph4_orth==nam
      verb-slot3_morph4_feat1_name=pernum
      verb-slot3_morph4_feat1_value=2sg
      verb-slot3_morph4_feat1_head=subj
    verb-slot3_morph5_orth==pam
      verb-slot3_morph5_feat1_name=pernum
      verb-slot3_morph5_feat1_value=2du, 2pl
      verb-slot3_morph5_feat1_head=subj
      verb-slot3_morph6_feat1_name=pernum
      verb-slot3_morph6_feat1_value=3rd
      verb-slot3_morph6_feat1_head=subj
  verb-slot4_name=trans-enclitic
  verb-slot4_order=after
    verb-slot4_input1_type=verb-slot2
    verb-slot4_morph1_orth==naš
      verb-slot4_morph1_feat1_name=pernum
      verb-slot4_morph1_feat1_value=1sg
      verb-slot4_morph1_feat1_head=higher
      verb-slot4_morph1_feat2_name=pernum
      verb-slot4_morph1_feat2_value=3rd
      verb-slot4_morph1_feat2_head=lower
    verb-slot4_morph2_orth==na
      verb-slot4_morph2_feat1_name=pernum
      verb-slot4_morph2_feat1_value=1du_incl, 1pl_incl
      verb-slot4_morph2_feat1_head=higher
  verb-slot4_morph2_feat2_name=pernum
  verb-slot4_morph2_feat2_value=3rd
  verb-slot4_morph2_feat2_head=lower
verb-slot4_morph3_orth==nataš
  verb-slot4_morph3_feat1_name=pernum
  verb-slot4_morph3_feat1_value=1du_excl, 1pl_excl
  verb-slot4_morph3_feat1_head=higher
  verb-slot4_morph3_feat2_name=pernum
  verb-slot4_morph3_feat2_value=3rd
  verb-slot4_morph3_feat2_head=lower
verb-slot4_morph4_orth==nam
  verb-slot4_morph4_feat1_name=pernum
  verb-slot4_morph4_feat1_value=2sg
  verb-slot4_morph4_feat1_head=subj
verb-slot4_morph5_orth==nam
  verb-slot4_morph5_feat1_name=pernum
  verb-slot4_morph5_feat1_value=3rd
  verb-slot4_morph5_feat1_head=subj
  verb-slot4_morph5_feat2_name=pernum
  verb-slot4_morph5_feat2_value=2sg
  verb-slot4_morph5_feat2_head=obj
verb-slot4_morph6_orth==pam
  verb-slot4_morph6_feat1_name=pernum
  verb-slot4_morph6_feat1_value=2du, 2pl
  verb-slot4_morph6_feat1_head=subj
verb-slot4_morph7_orth==pam
  verb-slot4_morph7_feat1_name=pernum
  verb-slot4_morph7_feat1_value=3rd
  verb-slot4_morph7_feat1_head=subj
  verb-slot4_morph7_feat2_name=pernum
  verb-slot4_morph7_feat2_value=2du, 2pl
  verb-slot4_morph7_feat2_head=obj
verb-slot4_morph8_orth==maš
  verb-slot4_morph8_feat1_name=pernum
  verb-slot4_morph8_feat1_value=1sg
  verb-slot4_morph8_feat1_head=subj
  verb-slot4_morph8_feat2_name=pernum
  verb-slot4_morph8_feat2_value=2sg
  verb-slot4_morph8_feat2_head=obj
verb-slot4_morph9_orth==mataš
  verb-slot4_morph9_feat1_name=pernum
```

```
      verb-slot4_morph9_feat1_value=1st
      verb-slot4_morph9_feat1_head=subj
      verb-slot4_morph9_feat2_name=pernum
      verb-slot4_morph9_feat2_value=2du, 2pl
      verb-slot4_morph9_feat2_head=obj
    verb-slot4_morph10_orth==mataš
      verb-slot4_morph10_feat1_name=pernum
      verb-slot4_morph10_feat1_value=1du, 1pl
      verb-slot4_morph10_feat1_head=subj
      verb-slot4_morph10_feat2_name=pernum
      verb-slot4_morph10_feat2_value=2sg
      verb-slot4_morph10_feat2_head=obj
      verb-slot4_morph11_feat1_name=pernum
      verb-slot4_morph11_feat1_value=3rd
      verb-slot4_morph11_feat1_head=subj
      verb-slot4_morph11_feat2_name=pernum
      verb-slot4_morph11_feat2_value=3rd
      verb-slot4_morph11_feat2_head=obj
  verb-slot5_name=subjprox
  verb-slot5_order=after
    verb-slot5_input1_type=verb-slot4
      verb-slot5_morph1_feat1_name=pernum
      verb-slot5_morph1_feat1_value=3sg
      verb-slot5_morph1_feat1_head=subj
      verb-slot5_morph1_feat2_name=proximity
      verb-slot5_morph1_feat2_value=proximate
      verb-slot5_morph1_feat2_head=subj
      verb-slot5_morph1_feat3_name=topicality
      verb-slot5_morph1_feat3_value=non-topic
      verb-slot5_morph1_feat3_head=obj
      verb-slot5_morph1_feat4_name=direction
      verb-slot5_morph1_feat4_value=inv
      verb-slot5_morph1_feat4_head=verb
      verb-slot5_morph2_feat1_name=pernum
      verb-slot5_morph2_feat1_value=3sg
      verb-slot5_morph2_feat1_head=subj
      verb-slot5_morph2_feat2_name=topicality
      verb-slot5_morph2_feat2_value=topic
      verb-slot5_morph2_feat2_head=subj
      verb-slot5_morph2_feat3_name=pernum
      verb-slot5_morph2_feat3_value=3sg
    verb-slot5_morph2_feat3_head=obj
    verb-slot5_morph2_feat4_name=proximity
    verb-slot5_morph2_feat4_value=obviative
    verb-slot5_morph2_feat4_head=subj
    verb-slot5_morph3_feat1_name=pernum
    verb-slot5_morph3_feat1_value=3sg
    verb-slot5_morph3_feat1_head=subj
    verb-slot5_morph3_feat2_name=pernum
    verb-slot5_morph3_feat2_value=3sg
    verb-slot5_morph3_feat2_head=obj
    verb-slot5_morph3_feat3_name=topicality
    verb-slot5_morph3_feat3_value=topic
    verb-slot5_morph3_feat3_head=obj
    verb-slot5_morph3_feat4_name=proximity
    verb-slot5_morph3_feat4_value=neutral
    verb-slot5_morph3_feat4_head=subj
    verb-slot5_morph4_feat1_name=pernum
    verb-slot5_morph4_feat1_value=3sg
    verb-slot5_morph4_feat1_head=subj
    verb-slot5_morph4_feat2_name=pernum
    verb-slot5_morph4_feat2_value=3du, 3pl
    verb-slot5_morph4_feat2_head=obj
    verb-slot5_morph4_feat3_name=proximity
    verb-slot5_morph4_feat3_value=neutral
    verb-slot5_morph4_feat3_head=subj
    verb-slot5_morph5_feat1_name=pernum
    verb-slot5_morph5_feat1_value=1st, 2nd, 3du, 3pl
    verb-slot5_morph5_feat1_head=subj
    verb-slot5_morph5_feat2_name=proximity
    verb-slot5_morph5_feat2_value=neutral
    verb-slot5_morph5_feat2_head=subj

section=test-sentences
```

Appendix H
SAHAPTIN TEST SENTENCES

The following list contains many, but not all, of the sentences used to test the Sahaptin grammar discussed in Chapter 6—in particular, approximately 6000 automatically-generated transitive sentences have been omitted to save space. This list contains both grammatical and ungrammatical examples; the ungrammatical examples are preceded by a "*". The sentences have been separated into labeled groups to make it easier to see all the variations that were tested.

In order to save space, I have not glossed the sentences below, but they are all made up of a small vocabulary of stems and affixes. The sentences are all in VSO order. All the intransitive sentences contain the verb stem *wína* 'go' and all transitive sentences, the stem *q̓ínun* 'see'. Verbs may optionally have a prefix and/or an enclitic as described in §6.1.2. The subject and object are always pronouns from Figure 6.5; additionally, third-person subjects and objects may be marked with the -*TOP* suffix to mark topicality.

Intransitive (grammatical):
wína=naš ín
wína=na napiiní
wína=na náma
wína=nataš napiiní
wína=nataš náma
wína=nam ím
wína=pam imiiní
wína=pam imáy
i-wína pɨ́n
pa-wína piiní
pa-wína pmáy

Intransitive (missing enclitic):
*wína ín
*wína napiiní
*wína náma
*wína ím
*wína imiiní
*wína imáy

Intransitive (missing prefix):
*wína pɨ́n
*wína piiní
*wína pmáy

Intransitive (wrong enclitic):
*wína=na ín
*wína=nataš ín
*wína=nam ín
*wína=pam ín
*wína=maš ín
*wína=mataš ín
*wína=naš napiiní
*wína=nam napiiní
*wína=pam napiiní
*wína=maš napiiní
*wína=mataš napiiní
*wína=naš náma
*wína=nam náma
*wína=pam náma
*wína=maš náma
*wína=mataš náma
*wína=naš ím
*wína=na ím
*wína=nataš ím
*wína=pam ím
*wína=maš ím
*wína=mataš ím
*wína=naš imiiní
*wína=na imiiní
*wína=nataš imiiní
*wína=nam imiiní
*wína=maš imiiní

*wína=mataš imiiní
*wína=naš imáy
*wína=na imáy
*wína=nataš imáy
*wína=nam imáy
*wína=maš imáy
*wína=mataš imáy
*i-wína=naš pɨ́n
*i-wína=na pɨ́n
*i-wína=nataš pɨ́n
*i-wína=nam pɨ́n
*i-wína=pam pɨ́n
*i-wína=maš pɨ́n
*i-wína=mataš pɨ́n
*pa-wína=naš piiní
*pa-wína=na piiní
*pa-wína=nataš piiní
*pa-wína=nam piiní
*pa-wína=pam piiní
*pa-wína=maš piiní
*pa-wína=mataš piiní
*pa-wína=naš pmáy
*pa-wína=na pmáy
*pa-wína=nataš pmáy
*pa-wína=nam pmáy
*pa-wína=pam pmáy
*pa-wína=maš pmáy
*pa-wína=mataš pmáy

Intransitive (wrong prefix):
*i-wína=naš ín
*pa-wína=naš ín
*á-wína=naš ín
*pá-wína=naš ín
*patá-wína=naš ín
*i-wína=nataš napiiní
*pa-wína=nataš napiiní
*á-wína=nataš napiiní
*pá-wína=nataš napiiní
*patá-wína=nataš napiiní
*i-wína=nataš náma

*pa-wína=nataš náma
*á-wína=nataš náma
*pá-wína=nataš náma
*patá-wína=nataš náma
*i-wína=nam ím
*pa-wína=nam ím
*á-wína=nam ím
*pá-wína=nam ím
*patá-wína=nam ím
*i-wína=pam imiiní
*pa-wína=pam imiiní
*á-wína=pam imiiní
*pá-wína=pam imiiní
*patá-wína=pam imiiní
*i-wína=pam imáy
*pa-wína=pam imáy
*á-wína=pam imáy
*pá-wína=pam imáy
*patá-wína=pam imáy
*pa-wína pɨ́n
*á-wína pɨ́n
*pá-wína pɨ́n
*patá-wína pɨ́n
*i-wína piiní
*á-wína piiní
*pá-wína piiní
*patá-wína piiní
*i-wína pmáy
*á-wína pmáy
*pá-wína pmáy
*patá-wína pmáy

Transitive (grammatical):
q̓ínun=maš ín imanáy
q̓ínun=mataš ín imiinamanáy
q̓ínun=mataš ín imaamanáy
q̓ínun=mataš napiiní imanáy
q̓ínun=mataš náma imanáy
q̓ínun=mataš napiiní imiinamanáy
q̓ínun=mataš napiiní imaamanáy
q̓ínun=mataš náma imiinamanáy

q̓ínun=mataš náma imaamanáy
á-q̓ínun=naš ín paanáy
á-q̓ínun=naš ín piinamanáy
á-q̓ínun=naš ín paamanáy
á-q̓ínun=na náma paanáy
á-q̓ínun=na napiiní paanáy
á-q̓ínun=na napiiní piinamanáy
á-q̓ínun=na napiiní paamanáy
á-q̓ínun=na náma piinamanáy
á-q̓ínun=na náma paamanáy
á-q̓ínun=nataš napiiní paanáy
á-q̓ínun=nataš náma paanáy
á-q̓ínun=nataš napiiní piinamanáy
á-q̓ínun=nataš napiiní paamanáy
á-q̓ínun=nataš náma piinamanáy
á-q̓ínun=nataš náma paamanáy
pá-q̓ínun=nam ím ináy
q̓ínun=nam ím napiinamanáy
q̓ínun=nam ím naamanáy
q̓ínun=pam imiiní ináy
q̓ínun=pam imáy ináy
q̓ínun=pam imiiní napiinamanáy
q̓ínun=pam imiiní naamanáy
q̓ínun=pam imáy napiinamanáy
q̓ínun=pam imáy naamanáy
á-q̓ínun=nam ím paanáy
á-q̓ínun=nam ím piinamanáy
á-q̓ínun=nam ím paamanáy
á-q̓ínun=pam imiiní paanáy
á-q̓ínun=pam imáy paanáy
á-q̓ínun=pam imiiní piinamanáy
á-q̓ínun=pam imiiní paamanáy
á-q̓ínun=pam imáy piinamanáy
á-q̓ínun=pam imáy paamanáy
i-q̓ínun=naš pnɨ́m ináy
i-q̓ínun=na pnɨ́m napiinamanáy
i-q̓ínun=na pnɨ́m naamanáy
i-q̓ínun=nataš pnɨ́m napiinamanáy
i-q̓ínun=nataš pnɨ́m naamanáy
pa-q̓ínun=naš pmáy ináy
pa-q̓ínun=naš piiní ináy

pa-q̓ínun=na piiní napiinamanáy
pa-q̓ínun=na piiní naamanáy
pa-q̓ínun=na pmáy napiinamanáy
pa-q̓ínun=na pmáy naamanáy
pa-q̓ínun=nataš piiní napiinamanáy
pa-q̓ínun=nataš piiní naamanáy
pa-q̓ínun=nataš pmáy napiinamanáy
pa-q̓ínun=nataš pmáy naamanáy
i-q̓ínun=nam pnɨ́m imanáy
i-q̓ínun=pam pnɨ́m imiinamanáy
i-q̓ínun=pam pnɨ́m imaamanáy
pa-q̓ínun=nam piiní imanáy
pa-q̓ínun=nam pmáy imanáy
pa-q̓ínun=pam piiní imiinamanáy
pa-q̓ínun=pam piiní imaamanáy
pa-q̓ínun=pam pmáy imiinamanáy
pa-q̓ínun=pam pmáy imaamanáy
i-q̓ínun piiní-TOP paanáy
pá-q̓ínun pɨ́n paanáy-TOP
i-q̓ínun pɨ́n piinamanáy
i-q̓ínun pɨ́n paamanáy
pa-q̓ínun piiní-TOP paanáy
pa-q̓ínun pmáy-TOP paanáy
patá-q̓ínun piiní paanáy-TOP
patá-q̓ínun pmáy paanáy-TOP
pa-q̓ínun piiní piinamanáy
pa-q̓ínun piiní paamanáy
pa-q̓ínun pmáy piinamanáy
pa-q̓ínun pmáy paamanáy

Transitive (missing enclitic):
*q̓ínun ín imanáy
*q̓ínun ín imiinamanáy
*q̓ínun ín imaamanáy
*q̓ínun napiiní imanáy
*q̓ínun náma imanáy
*q̓ínun napiiní imiinamanáy
*q̓ínun napiiní imaamanáy
*q̓ínun náma imiinamanáy
*q̓ínun náma imaamanáy
*á-q̓ínun ín paanáy

*á-q̓ínun ín piinamanáy

*á-q̓ínun ín paamanáy

*á-q̓ínun náma paanáy

*á-q̓ínun napiiní paanáy

*á-q̓ínun napiiní piinamanáy

*á-q̓ínun napiiní paamanáy

*á-q̓ínun náma piinamanáy

*á-q̓ínun náma paamanáy

*pá-q̓ínun ím ináy

*q̓ínun ím napiinamanáy

*q̓ínun ím naamanáy

*q̓ínun imiiní ináy

*q̓ínun imáy ináy

*q̓ínun imiiní napiinamanáy

*q̓ínun imiiní naamanáy

*q̓ínun imáy napiinamanáy

*q̓ínun imáy naamanáy

*á-q̓ínun ím paanáy

*á-q̓ínun ím piinamanáy

*á-q̓ínun ím paamanáy

*á-q̓ínun imiiní paanáy

*á-q̓ínun imáy paanáy

*á-q̓ínun imiiní piinamanáy

*á-q̓ínun imiiní paamanáy

*á-q̓ínun imáy piinamanáy

*á-q̓ínun imáy paamanáy

*i-q̓ínun pnɨ́m ináy

*i-q̓ínun pnɨ́m napiinamanáy

*i-q̓ínun pnɨ́m naamanáy

*pa-q̓ínun pmáy ináy

*pa-q̓ínun piiní ináy

*pa-q̓ínun piiní napiinamanáy

*pa-q̓ínun piiní naamanáy

*pa-q̓ínun pmáy napiinamanáy

*pa-q̓ínun pmáy naamanáy

*i-q̓ínun pnɨ́m imanáy

*i-q̓ínun pnɨ́m imiinamanáy

*i-q̓ínun pnɨ́m imaamanáy

*pa-q̓ínun piiní imanáy

*pa-q̓ínun pmáy imanáy

*pa-q̓ínun piiní imiinamanáy

*pa-q̓ínun piiní imaamanáy
*pa-q̓ínun pmáy imiinamanáy
*pa-q̓ínun pmáy imaamanáy

Transitive (missing prefix):
*q̓ínun=naš ín paanáy
*q̓ínun=naš ín piinamanáy
*q̓ínun=naš ín paamanáy
*q̓ínun=na náma paanáy
*q̓ínun=na napiiní paanáy
*q̓ínun=na napiiní piinamanáy
*q̓ínun=na napiiní paamanáy
*q̓ínun=na náma piinamanáy
*q̓ínun=na náma paamanáy
*q̓ínun=nataš napiiní paanáy
*q̓ínun=nataš náma paanáy
*q̓ínun=nataš napiiní piinamanáy
*q̓ínun=nataš napiiní paamanáy
*q̓ínun=nataš náma piinamanáy
*q̓ínun=nataš náma paamanáy
*q̓ínun=nam ím ináy
*q̓ínun=nam ím paanáy
*q̓ínun=nam ím piinamanáy
*q̓ínun=nam ím paamanáy
*q̓ínun=pam imiiní paanáy
*q̓ínun=pam imáy paanáy
*q̓ínun=pam imiiní piinamanáy
*q̓ínun=pam imiiní paamanáy
*q̓ínun=pam imáy piinamanáy
*q̓ínun=pam imáy paamanáy
*q̓ínun=naš pnɨ́m ináy
*q̓ínun=na pnɨ́m napiinamanáy
*q̓ínun=na pnɨ́m naamanáy
*q̓ínun=nataš pnɨ́m napiinamanáy
*q̓ínun=nataš pnɨ́m naamanáy
*q̓ínun=naš pmáy ináy
*q̓ínun=naš piiní ináy
*q̓ínun=na piiní napiinamanáy
*q̓ínun=na piiní naamanáy
*q̓ínun=na pmáy napiinamanáy
*q̓ínun=na pmáy naamanáy

*q̓ínun=nataš piiní napiinamanáy
*q̓ínun=nataš piiní naamanáy
*q̓ínun=nataš pmáy napiinamanáy
*q̓ínun=nataš pmáy naamanáy
*q̓ínun=nam pnɨ́m imanáy
*q̓ínun=pam pnɨ́m imiinamanáy
*q̓ínun=pam pnɨ́m imaamanáy
*q̓ínun=nam piiní imanáy
*q̓ínun=nam pmáy imanáy
*q̓ínun=pam piiní imiinamanáy
*q̓ínun=pam piiní imaamanáy
*q̓ínun=pam pmáy imiinamanáy
*q̓ínun=pam pmáy imaamanáy
*q̓ínun piiní-TOP paanáy
*q̓ínun pɨ́n paanáy-TOP
*q̓ínun pɨ́n piinamanáy
*q̓ínun pɨ́n paamanáy
*q̓ínun pmáy-TOP paanáy
*q̓ínun piiní paanáy-TOP
*q̓ínun pmáy paanáy-TOP
*q̓ínun piiní piinamanáy
*q̓ínun piiní paamanáy
*q̓ínun pmáy piinamanáy
*q̓ínun pmáy paamanáy

VITA

Scott Drellishak was born and raised in Southern California. He attended the University of California, Berkeley, where he earned a Bachelor of Science in Electrical Engineering and Computer Sciences. After working for several years in the software industry, he returned to academics, studying Linguistics at the University of Washington and earning first a Master of Arts and then a Doctor of Philosophy. He currently lives in Seattle, WA.