Linguistic **I**ssues in **L**anguage **T**echnology – **LiLT**
Submitted, June 29, 2009

Reweaving a Grammar for Wambaya

A Case Study in Grammar Engineering for Linguistic Hypothesis Testing

Emily M. Bender

Submitted, June 29, 2009
Revised, December 28, 2009
Published by CSLI Publications

LiLT volume 3, issue 3 February 2010

Reweaving a Grammar for Wambaya

A Case Study in Grammar Engineering for Linguistic Hypothesis Testing

Emily M. Bender, *University of Washington*

Abstract

This paper presents a case study in grammar engineering for linguistic hypothesis testing, focusing on the treatment of second position auxiliaries in an HPSG grammar for Wambaya. A detailed comparison of two versions of this grammar highlights the interconnetedness of linguistic phenomena, the model-dependence of linguisic analyses, and the value of computational support in calculating the consequences of differing analytical choices.

LiLT Volume 3, Issue 3, February 2010.
Reweaving a Grammar for Wambaya.
Copyright © 2010, CSLI Publications.

1 Introduction

Grammars of natural languages are highly complex objects, in which analyses of many distinct phenomena must interact in order to correctly map even the simplest sentence to its semantic representation (Bender, 2008b). Furthermore, working syntacticians are faced with the dilemma that the available data generally vastly underdetermine the theory: It is typically the case that many possible formal analyses are compatible with the same set of basic data (intuitions about the grammaticality and meaning of some set of strings). At the point of analyzing any particular phenomenon, the currently accepted (or, in our case, implemented) analyses of interacting phenomena serve to constrain the space of possible analyses to some extent. Thus our *models* of grammars of languages are themselves complex objects, in which analytical choices are made in light of previous analytical choices, but usually could be revised provided we are willing to do the work of following the cascade of required changes to other analyses.

Grammar engineering, as a subdiscipline, enlists the aid of computers to handle this complexity. In the first instance, this allows us to verify that the various analyses indeed function together as intended and to check that each additional analysis added to the grammar does not break previous functionality (see Bender et al. 2009). Grammar engineering typically proceeds along a course of *incremental development*, in which new phenomena are incorporated into the grammar, tested, and documented in the test suite for further iterations. Because the grammar must be able to handle at least some complete sentences in order to enable this testing, and/or because a practical application domain requires quick start-up, sometimes place-holder analyses are put in for various phenomena and then refined at a later date.

In this context, hypothesis testing is usually confined to the exploration of alternative analyses of new phenomena being added. Assuming equivalent or near-equivalent coverage over known data, whichever analysis is most easily incorporated into the grammar, with the least (or most fixable) loss of existing coverage, is chosen. Occasionally, this incremental development leads to fundamental reworkings of core pieces of the grammar, but this is rarely carried out in a strictly comparative fashion. That is, despite the potential of grammar engineering for linguistic hypothesis testing, it is rarely used in this way.

This paper reports on a case study comparing two different analyses of a fundamental piece of a grammar (the auxiliary system) through parallel development of two branches of a grammar project. The goal was to explore, via the implementation of the grammars, the ramifications of each choice for other parts of the interrelated web of the model.

The rest of the paper is structured as follows: In §2, I introduce the language studied (Wambaya), and the initial grammar. §3 describes the competing hypotheses and §4 details the methodology that was used to implement and compare them. §5 presents the results of the study: both quantitative and qualitative comparisons of the two grammar branches.

2 Background

Wambaya [wmb] is a non-Pama-Nyungan language of the Mirndi family (Green and Nordlinger, 2004) from the West Barkly Tablelands region of the Northern Territory of Australia.[1] It was originally described and documented by Rachel Nordlinger (1998). The formal analyses described here were developed on the basis of the descriptive analyses in Nordlinger 1998 and the annotated (IGT format) examples from that work. These examples (numbering 804) became the development test suite for the implemented grammar. None of this work would have been possible without the extensive prior analytical work of Nordlinger.

The implemented Wambaya grammar was originally developed as a test of the typological breadth of the LinGO Grammar Matrix (Bender et al., 2002, Bender and Flickinger, 2005). It is situated within the framework of HPSG (Pollard and Sag, 1994), and uses Minimal Recursion Semantics (Copestake et al., 2005) for its semantic representations. In the initial grammar development work (Bender, 2008a), I brought the grammar to 91% coverage over the 804 development examples, where an example was counted as "covered" if among the analyses assigned to it by the grammar was one that provided a predicate-argument argument structure and set of variable properties (person, number, gender, tense, aspect and illocutionary force) that matched the gloss as provided by Nordlinger. I then tested the grammar against a held-out test set of 72 examples: every sentence from a naturally occurring narrative included in Nordlinger 1998 that wasn't already included in the development set. With additional vocabulary mapped to existing lexical types but no further changes to the grammar, it achieved 76% coverage of the held-out data. In addition, the grammar has relatively low ambiguity, assigning on average 12.56 parses to each item in the test set.

In order to achieve this coverage, the grammar had to include analyses of a wide variety of phenomena, including: word order (allowing grammatical combinations while disallowing ungrammatical ones);

[1] The last fluent speakers have passed away since the late 1990s (Nordlinger, p.c.).

the linking of syntactic to semantic arguments, including case assignment; agreement between auxiliaries and verbal dependents and between nouns and their modifiers; modifiers including lexical adverbs, nominals in particular cases used as adverbials, demonstrative, numeral and possessive lexical adjectives and derived adjectives; subordinate clauses including clausal complements and modifiers such as purposives; verbless clauses; markers of illocutionary force; coordination and other sundry phenomena including an inalienable possession construction, secondary predicates, and causatives. Between the initial grammar development work and the start of the hypothesis testing explored here, I also extended the grammar to include semantic quantifier relations associated with each nominal position.

2.1 Salient linguistic features

This section gives a very quick sketch of some relevant linguistic features of Wambaya. Further details will be introduced in the discussion of particular analyses below. Perhaps the most prominent feature of Wambaya is its non-configurational word order. Aside from the constraint that the clitic cluster (analyzed here as an auxiliary) be in second position, the word order within a clause is free, to the point that noun phrases may be discontinuous. An example is shown in (1).[2]

(1) Babaga-yi nyi-n jundurra mirnda
sister.II-LOC 2.SG.A.PRES-PROG dust.IV.ACC 1.DU.INC.OBL
bajbaga yardi.
big.IV.ACC put
'Sister you're making lots of dust for us.' [wmb]
(Nordlinger, 1998, 315)

[2] Glosses are as given in the original. This paper uses the following abbreviations:

	person		*case*		*grammatical role*
1	first person	ACC	accusative	A	agent
2	second person	DAT	dative	S	subject
3	third person	LOC	locative/ergative		*phonological*
INC	inclusive	NOM	nominative	TH	thematic consonant
	gender	OBL	oblique		
I	noun class I		*tense/aspect*	EP	epenthetic vowel
II	noun class II	PRES	present		
III	noun class III	PROG	progressive		*other*
IV	noun class IV	PST	past	NEG	negation
	number	NF	non-future	RR	reflexive/reciprocal
SG	singular	INF	infinitival	AWY	direction away
DU	dual				
PL	plural				

In (1), *jundurra* 'dust.IV.ACC' is modified by *bajbaga* 'big.IV.ACC', even though they are not contiguous. The linking of these two elements is achieved through the agreement marking indicating gender (IV) and case (accusative).

The second position of this example is occupied by an auxiliary (or clitic cluster), as is typical for clauses with verbal predicates.[3] The auxiliary registers agreement in person, number, and gender (third person only) with the subject and with the object (with zero affixes for third person objects). It can also register tense and aspect information.[4]

The case marking follows a split-ergative pattern, with third person noun phrases marked according to the ergative-absolutive pattern and first and second person pronouns marked according to the nominative-accusative pattern. Following Nordlinger (1998), I analyze this as underlyingly a tripartite case system. Non-core cases include dative, allative, ablative, comitative, and genitive. Some verbs idiosyncratically select for non-core cases. In addition, many of the cases are also correlated with modifier roles. For example, ergative-marked nominals can serve as locatives and dative-marked nominals as beneficiaries.

3 Competing hypotheses

3.1 Argument composition

The core linguistic facts that this paper focuses on pertain to the auxiliaries. In particular: the nominals in a sentence are dependents of the verb, but they are ordered with respect to the auxiliary,[5] and furthermore, it is the auxiliary and not the verb that bears the agreement morphology. In the initial implemented grammar, these facts were taken to motivate an argument-composition analysis (Bender, 2008c) in the style of Hinrichs and Nakazawa 1990. On this analysis, the auxiliaries are lexically specified to take the verb as their first complement, and then 'adopt' the subject and complements of the verb as their own arguments. In the HPSG notation, this is as shown in (2).[6]

---

[3] Subordinate clauses, coordinands, and clauses headed by adjectives or nouns do not need and in fact in some cases cannot have an auxiliary.

[4] The second position is defined in terms of constituents, in that the auxiliary must follow the first constituent of the sentence. Note that verbs can appear in the pre-auxiliary position, but not with their arguments.

[5] More specifically, the ordering constraints concern the auxiliary, which must be in second position, and not the verb, which is ordered freely with respect to its arguments.

[6] For a general introduction to HPSG-style grammars, see Sag et al. 2003.

(2) *arg-comp-aux:*

$$\begin{bmatrix} \textsc{head} & \begin{bmatrix} \textit{verb} & \\ \textsc{aux} & + \end{bmatrix} \\ \textsc{val} & \begin{bmatrix} \textsc{subj} & \langle\ \boxed{1}\ \rangle \\ \textsc{comps} & \left\langle \begin{bmatrix} \textsc{opt} & - \\ \textsc{head} & \begin{bmatrix} \textit{verb} & \\ \textsc{from} & \textit{fin} \end{bmatrix} \\ \textsc{subj} & \langle\ \boxed{1}\ \rangle \\ \textsc{comps} & \boxed{2} \end{bmatrix} \right\rangle \oplus \boxed{2} \end{bmatrix} \end{bmatrix}$$

To take a concrete example, the verb in (1) above (*yardi* ‘put’) is associated with the information in (3).[7] The auxiliary *nyi-n* starts off with the underspecified values for SUBJ and COMPS shown in (2), but when it combines with the verb *yardi*, they are (partially) resolved. In particular, the tags $\boxed{1}$ and $\boxed{2}$ in (2) serve to ‘transmit’ the subject and complement requirements of the verb to the auxiliary, resulting in a structure like the one in (4).

(3)
$$\begin{bmatrix} \textsc{head} & \begin{bmatrix} \textit{verb} & \\ \textsc{aux} & - \end{bmatrix} \\ \textsc{val} & \begin{bmatrix} \textsc{subj} & \langle\ \mathrm{NP}[\textsc{case}\ \textit{erg}]\ \rangle \\ \textsc{comps} & \langle\ \mathrm{NP}[\textsc{case}\ \textit{acc}],\ \mathrm{NP}[\textsc{case}\ \textit{all}]\ \rangle \end{bmatrix} \end{bmatrix}$$

(4)
$$\begin{bmatrix} \textsc{head} & \begin{bmatrix} \textit{verb} & \\ \textsc{aux} & + \end{bmatrix} \\ \textsc{val} & \begin{bmatrix} \textsc{subj} & \langle\ \boxed{1}\mathrm{NP}[\textsc{case}\ \textit{erg}]\ \rangle \\ \textsc{comps} & \left\langle \begin{matrix} \mathrm{V} \\ \begin{bmatrix} \textsc{subj} & \langle\boxed{1}\rangle \\ \textsc{comps} & \boxed{2}\langle\boxed{3},\ \boxed{4}\rangle \end{bmatrix} \end{matrix} \right\rangle \\ & \oplus\ \boxed{2}\langle\ \boxed{3}\mathrm{NP}[\textsc{case}\ \textit{acc}],\ \boxed{4}\mathrm{NP}[\textsc{case}\ \textit{all}]\ \rangle \end{bmatrix} \end{bmatrix}$$

[7] In fact, it is associated with much more information, including the mapping of these syntactic arguments to the appropriate semantic arguments. Very little of this information is stipulated in the lexical entry itself: only the stem form, the name of the semantic predicate, and the lexical type that the entry instantiates and from which it inherits the rest of the information.

This combination could be licensed by an instance of the head-complement rule, giving the subtree shown in (5):

(5) V[AUX +] = (4)

V[AUX +] = (2) V[AUX −] = (3)

nyin yardi

In standard HPSG analyses, the head-complement rule combines a head looking for a complement with a constituent matching the constraints on the first element of the head's COMPS list, and creates a larger constituent, which is (syntactically) quite similar to the head daughter except that its COMPS list has been shortened by one. The Wambaya grammar instead follows a non-cancellation analysis, as described in Bender 2008c,[8] so that elements remain on the valence lists even after they have been realized. This facilitates the analysis of discontinuous NPs.

In order to capture the word order facts of the language, this version of the Wambaya grammar includes multiple types of rules for realizing dependents of a head:

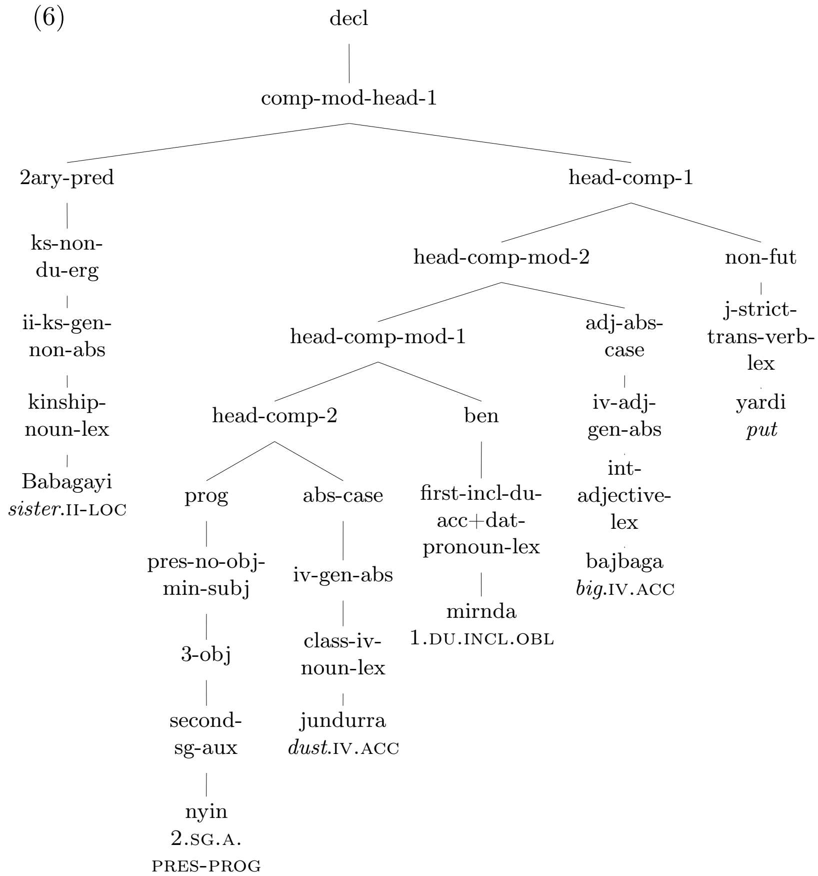
- A series of head-complement rules, which can realize the first, second or third element of the COMPS list
- Head-subject rules
- Head-modifier rules
- Head-arg-mod rules, which attach modifiers which semantically modify some argument of the syntactic head, rather than the syntactic head itself. Like the head-complement rules, these come in a series, for targeting different positions of the COMPS list, plus one for subject modifiers.

These rule types are cross-classified with two other dimensions: One dimension distinguishes clauses headed by auxiliaries from those headed by main verbs or nominal or adjectival predicates. The other dimension distinguishes head-initial from head-final rules. For non-auxiliary clauses, these simply allow the dependent to attach to the left or to the right of the head. In clauses with auxiliaries, however, they also serve the function of ensuring that the auxiliary is in second position: The auxiliary-clause rules are constrained such that the auxiliary must first

[8] For similar proposals, see Meurers 1999 and Müller 2008 on German and Przepiórkowski 1999 on Polish.

pick up all dependents to its right before combining with one single dependent to its left.[9] This leads to trees with a fairly uniform structure. As an example, (6) presents the phrase structure tree assigned to (1) on the intended reading.[10]

(6)



[9] There are a couple of constructions that allow for yet one more sentence-initial position, before the pre-aux position. These are handled by rules which combine an appropriate constituent with a complete auxiliary-headed clause.

[10] In these trees, the preterminal nodes indicate the lexical types of the lexical entries used. The strings of non-branching nodes above the lexical types are chains of lexical rules, some of which have morphophonological effects (not shown).

This parse, and the ones shown in (11) and (12) are in fact not available in the respective versions of the grammar distributed with this paper, due to a bug with the secondary predicate phrase (2ary-pred) that was discovered as we went to press. Structurally analogous parses with the initial noun serving as a locative modifier instead of a secondary predicate are available. The bug is fixed in the most current version of the grammar.

The argument composition analysis appears to present a fairly elegant account of the word order facts, while also accommodating the fact that the auxiliary needs access to the verbal complements in order to register agreement. However, there is no reason to believe it is the only possible analysis of this set of facts. Furthermore, it suffers from inefficiencies in processing, particularly in generation. The heart of the problem is the underspecified valence lists, and especially the COMPS of indeterminate length: until the auxiliary combines with the verbal complement, its COMPS list is almost completely underspecified, and most problematically, unbounded. The grammar is specifically designed so that the auxiliary does not have to pick up the verbal complement first, because the word order facts of Wambaya are such that the auxiliary and the verb do not have to be adjacent. Therefore, the grammar allows the auxiliary to combine with whatever is next to it. The resulting constituents place constraints on the COMPS list of the verbal complement, ensuring that they will only contribute to a successful parse of the sentence if a verb looking for those dependents is found. This, in turn, means that the search space for the parsing algorithm includes constituents combining the auxiliary and any possible constituents next to it, one to the left, and as many as possible to the right. The problem is compounded in generation, as there are many different auxiliary forms and each one needs to be explored by the generator since it might be required in a successful realization.[11]

3.2 Auxiliary+verb cluster

This section describes the alternative analysis explored in this work, which avoids argument composition (and unbounded COMPS lists) and instead posits auxiliaries which select only for verbs as complements together with special constructions (phrase structure rules) for combining the auxiliaries with verbs or verbal projections.[12] These constructions differ from the head complement rule in that they construct the valence information on the mother node by combining information from the two daughter nodes. The auxiliary type is shown in (7) and a type for aux+verb constructions is illustrated in (8):[13,14]

---

[11] The auxiliary forms are distinguished by their agreement features, and the current generation algorithm in the LKB (Carroll et al., 1999, Carroll and Oepen, 2005) does not filter potential lexical entries based on agreement.

[12] This key idea in this analysis—constructions for combining auxiliaries with verbs/verbal clusters—was suggested to me by Dan Flickinger.

[13] As elsewhere in this paper, these displays omit details of the constructions not relevant to the immediate point.

[14] It may seem unnecessary to include the verbal complement of the auxiliary on the COMPS list of the mother. In the current grammar, the verbal complement is

(7) *non-arg-comp-aux:*

$$\begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{AUX} & + \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SUBJ} & \langle\ \rangle \\ \text{COMPS} & \left\langle \begin{bmatrix} \text{CAT.HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{AUX} & - \end{bmatrix} \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

(8) *aux+verb-rule:*

$$\begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{HEAD} & \boxed{1} \\ \text{VAL} & \begin{bmatrix} \text{SUBJ} & \boxed{2} \\ \text{COMPS} & \langle\ \boxed{3}\ \rangle \oplus \boxed{4} \end{bmatrix} \end{bmatrix} \\ \text{HEAD-DTR} & \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{HEAD} & \boxed{1} \\ \text{VAL.COMPS} & \langle\ \boxed{3}\ \rangle \end{bmatrix} \end{bmatrix} \\ \text{NON-HEAD-DTR} & \boxed{3} \begin{bmatrix} \text{CAT.VAL} & \begin{bmatrix} \text{SUBJ} & \boxed{2} \\ \text{COMPS} & \boxed{4} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

As shown in (7), on this analysis, the auxiliary has a complements (COMPS) list of length one. The construction (8) constrains the mother's complements list to contain the auxiliary's complement requirement (the verb, $\boxed{3}$) followed by any complements the verb requires ($\boxed{4}$). The subject requirement ($\boxed{2}$) comes from the verb, while the HEAD properties ($\boxed{1}$) come from the auxiliary.

On this analysis, when the verb appears to the left of the auxiliary, the verb and auxiliary combine first, and then pick up the rest of the dependents to the right. On the other hand, when the verb appears to the right of the auxiliary, the verb picks up all of the other dependents and the auxiliary combines with the resulting verbal cluster, before combining with one constituent to its left. To this end, there are two subtypes of *aux+verb-rule*: *aux+verb-left*, shown in (9) and *aux+verb-group-right*, shown in (10).[15]

---

included on the COMPS of the mother in order to facilitate the analysis of certain adverbs. These adverbs are constrained to modify only non-auxiliary verbs, but can attach to constituents headed by an auxiliary to modify the verbal complement.

[15] These inherit the general constraints on order from the types *head-final* and *head-initial*. In (9) and (10) these constraints are shown incorporated directly into the rule, as the relationship between the features HEAD-DTR, NON-HEAD-DTR, and ARGS.

(9) *aux+verb-left:*

$$\begin{bmatrix} \textsc{cat.mc} & \textit{bool} \\ \textsc{head-dtr} & \boxed{1} \\ \textsc{non-head-dtr} & \boxed{2}\begin{bmatrix}\textsc{cat.head.sat} & - \end{bmatrix} \\ \textsc{args} & \langle\ \boxed{2}, \boxed{1}\ \rangle \end{bmatrix}$$

(10) *aux+verb-group-right:*

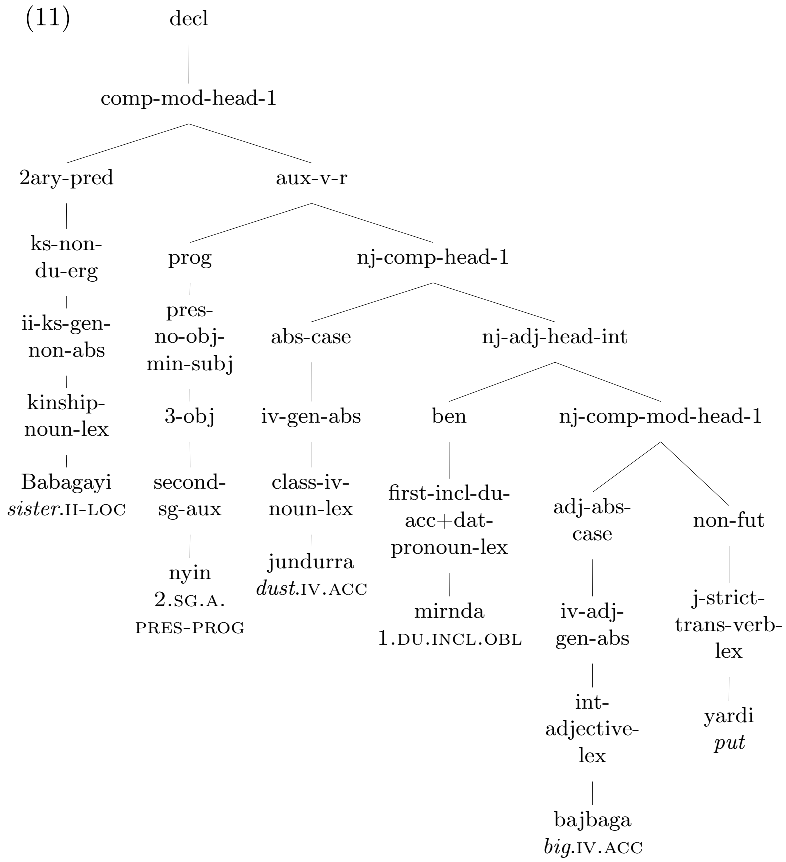
$$\begin{bmatrix} \textsc{cat.mc} & \textit{na} \\ \textsc{head-dtr} & \boxed{1} \\ \textsc{non-head-dtr} & \boxed{2} \\ \textsc{args} & \langle\ \boxed{1}, \boxed{2}\ \rangle \end{bmatrix}$$

The constraint [SAT −] on the non-head daughter interacts with constraints on the rules that combine verbs and their dependents to ensure that only bare verbs can fill this position.[16] The constraints on the feature MC ('main clause') interact with constraints on the rules for combining aux+verb clusters with other dependents to make sure that auxiliaries that have picked up a verb cluster to the right pick up exactly one more dependent to the left, and auxiliaries that have picked up a verb to the left only pick up further dependents to the right.

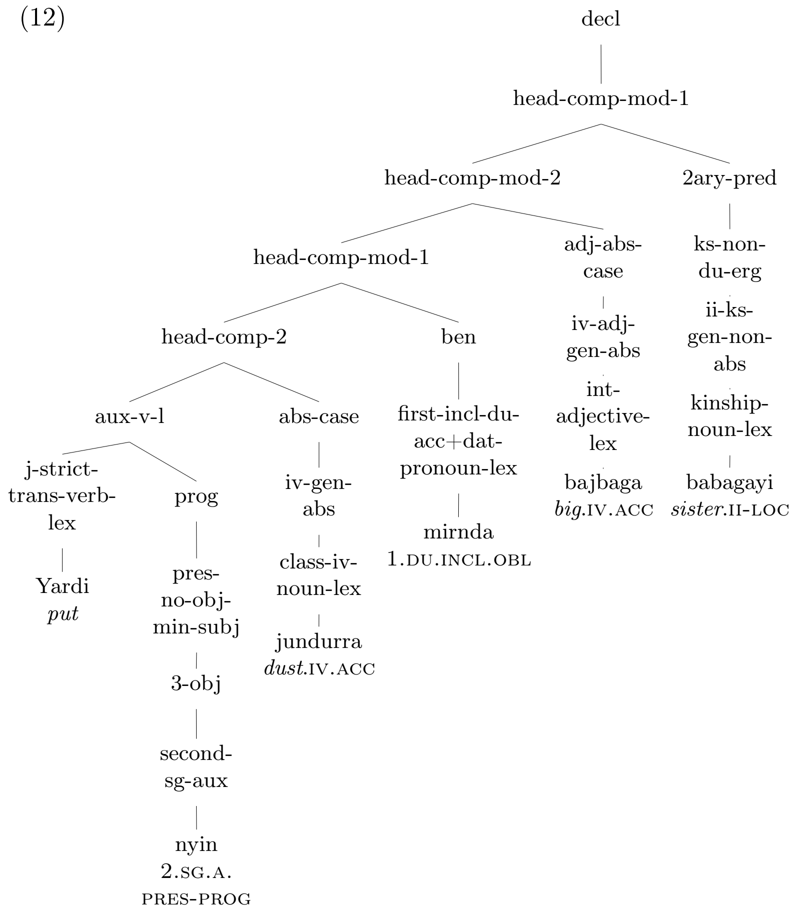
The net result is low attachment of the pre-aux constituent when it is a verb and high attachment of the constituent in that position otherwise. This is illustrated (11) and (12), where (11) is the analysis assigned to (1) and (12) is the analysis assigned to a similar (constructed) example with the verb in sentence-initial position.

[16] Recall that verbs cannot appear together with their arguments in initial position.

(11)


decl
comp-mod-head-1
2ary-pred
aux-v-r
ks-non-du-erg
ii-ks-gen-non-abs
kinship-noun-lex
Babagayi
sister.II-LOC
prog
pres-no-obj-min-subj
3-obj
second-sg-aux
nyin
2.SG.A.PRES-PROG
nj-comp-head-1
abs-case
iv-gen-abs
class-iv-noun-lex
jundurra
dust.IV.ACC
nj-adj-head-int
ben
first-incl-du-acc+dat-pronoun-lex
mirnda
1.DU.INCL.OBL
nj-comp-mod-head-1
adj-abs-case
iv-adj-gen-abs
int-adjective-lex
bajbaga
big.IV.ACC
non-fut
j-strict-trans-verb-lex
yardi
put

(12)



As in the other version of the grammar, this version employs a non-cancellation analysis to handle the discontinuous noun phrases. As a result, even though the auxiliary doesn't select for the arguments it is agreeing with (subject and object), it has access to them through the valence lists of the verb, even if they have been realized already within the verbal cluster. As an example, (13) shows the lexical rule which produces auxiliaries inflected for 1st person objects:[17]

[17] The constraint on the daughter, inherited from a supertype, shows that this rule only applies to non-imperative auxiliaries. Other supertypes guarantee that the information provided in the daughter is copied up to the mother.

(13) *first-person-obj-mark-lex-rule*:

$$\begin{bmatrix} \textsc{comps} & \left\langle \begin{bmatrix} \textsc{comps}\ \left\langle \begin{bmatrix}\textsc{index.png.pn}\ \textit{first}\end{bmatrix}\right\rangle \end{bmatrix} \right\rangle \\ \textsc{dtr} & \textit{non-imp-aux} \end{bmatrix}$$

This analysis is perhaps less elegant than the argument composition analysis in that the tree geometry is less consistent. On the other hand, the lack of unbounded valence lists is an advantage. As discussed in §5, the different tree geometry did not lead to large differences in predictions (over the available data set). In particular, though one might expect the change in constituent structure to interact with coordination, the analysis of coordination continues to function as expected. The only difference in predictions that can currently be traced to this different geometry has to do with the semantics of scopal modifiers, discussed briefly in §5.3.

3.3 Summary

This section has briefly sketched the two competing hypotheses that are at the heart of the experiment reported in this paper. Looking only at these aspects of the grammar (position of the auxiliary within the sentence, agreement marking on the auxiliary, etc), no differences in predictions between the analyses are apparent. However, by implementing the grammar first one way and then the other, it is possible to explore how and where the analyses lead to differences in their interactions with other aspects of the grammar.

4 Methodology

The starting point for this project was the grammar described in §2 above. The grammar development was done with the LKB grammar development environment (Copestake, 2002). Comparison of different grammar versions was carried out with the [incr tsdb()] competence and performance profiling environment (Oepen, 2001), using the PET parser (Callmeier, 2002) for efficient processing of test suite examples.[18] The test suite used consisted of the 804 sentences used as examples in the descriptive chapters of Nordlinger 1998. This is the same data as was used as the development set in Bender 2008a, and in fact represents most of the available data for Wambaya.[19]

[18] These tools are all available as open source software from the DELPH-IN consortium: http://www.delph-in.net/

[19] Nordlinger 1998 also includes a small number of transcribed, glossed and translated narratives, one of which was used for the test data in Bender 2008a.

The initial goal of the project was to start with the baseline grammar, edit it to implement the non-argument composition analysis of auxiliaries, and then explore what further modifications were needed to bring the new grammar in line with the baseline in terms of the number of analyses per input sentence and the semantic representation assigned. However, the baseline grammar, while interesting and reasonably complex, is nonetheless imperfect: it does not have complete coverage over the development set, and furthermore, it admits some spurious ambiguity and faulty semantic representations.[20] Thus some of the differences between the developing aux+verb-cluster grammar and the baseline grammar in fact represented improvements.

Rather than deliberately add bugs to the aux+verb-cluster grammar, I opted to branch the grammar development, and port improvements from the aux+verb-cluster branch to the argument-composition branch. To make sure that these improvements do not introduce regressions as a side effect, I used [incr tsdb()] to compare the semantic representations of the new versions on both branches to the semantic representations from the baseline grammar, and in particular, to those annotated as the preferred analyses of each input item in earlier work.[21] Thus fewer analyses is okay, so long as the preferred analysis is maintained. In addition, as discussed further in §5.3 below, some of the improvements involved redesign of semantic representations. In this case, the relevant examples were checked by hand against the gold standard semantic representations to ensure that this redesign was in fact the cause of the difference.

4.1 Empricist grammar engineering

An assumption here is that the sentences in the development set are sufficiently representative of the phenomena in the language to catch analyses that are off the mark. This is especially important because I do not have access to speaker intuitions about the examples or about how the various linguistic elements behave in other examples, beyond was is documented in Nordlinger 1998 and the glosses she provides. The need for more data is perhaps particularly acute because much of the work involved considering the range of dispreferred analyses available. That is, I would classify ambiguity as spurious ambiguity if the constraints

[20]No complete precision grammar exists for any language. Grammar engineering is always a matter of incremental development.

[21]This annotation was done by comparing the glosses in Nordlinger 1998 to the semantic representations produced by the baseline grammar, and selecting the analysis which matched the gloss in terms of semantic dependencies, illocutionary force, tense, aspect, mood, person, number and gender. If no analysis was a good match for the gloss, then the item was rejected.

that ruled it out did not also rule out any of the preferred analyses of any of the sentences in the development set. It would of course be better to have more data. On the other hand, there is not much more data available for this language, and most of what is left is being held out to serve as test data in future work. Furthermore, as example sentences from a thorough descriptive grammar, this collection of 804 sentences is more diverse in the phenomena illustrated than might otherwise be expected.

Development work progressed according to what Oepen et al. (2002) referred to as a "strongly empiricist style of grammar engineering" (p. 18), making heavy use of [incr tsdb()] to identify test items of interest and verify the results of changes to the grammar. In particular, this involved processing the test sentences with the current version of each branch of the grammar and comparing the resulting grammar 'profiles' to discover which sentences had different numbers of analyses, and which sentences had different semantic representations.[22] For example, Figure 1 shows the result of a comparison between the two branches mid-way through the process. The column labeled '<' indicates the number of semantic representations for each item found only in the old profile (in this case, the profile processed with the argument-composition grammar), the column labeled '=' indicates the number of semantic representations shared between the two grammars, and the column labeled '>' indicates the number found only in the new profile (aux+verb-cluster grammar). These numbers are all clickable, bringing up displays of the semantic representations in question.

The overall development strategy was to first focus on examples that failed to parse at all in the new grammar despite parsing in the old (increasing coverage), then to focus on examples that were more ambiguous according to the new grammar (reducing ambiguity), and finally on examples that had the same number of analyses but differed in some or all of the semantic representations. In most cases, it was efficient to focus on relatively unambiguous examples first, as the ambiguity in the items with many analyses tends to be the product of many different sources of ambiguity illustrated separately in the simpler examples. The simpler examples were easier to work with, and solving problems there could sometimes resolve additional discrepancies in the more complicated examples, without the latter having to be investigated by hand.

Another very important tool in this process is the treebank anno-

[22] Due to the change in the analysis of auxiliaries, the majority of examples have different syntactic derivations, so comparing at that level was not informative.

i-id	i-input	(g)old readings	new readings	mrs <	mrs =	mrs >
1	Gurda ngiyingga.	1	0	1	0	0
4	Ngurruwani ngurrun mirra gili ngarlini.	6	12	6	0	12
9	Guyala wurlungguji daguma.	1	0	1	0	0
15	Gulugbi ga, yandu yangaji nanga najbarda.	274	242	274	0	242
17	Yarru gany yanybarda manganyminka.	16	14	16	0	14
21	Alagbulu wurlungga nyurrunyurru.	1	0	1	0	0
25	Ini juguli dagumajinka.	1	0	1	0	0
27	Gayinirna=miji nayida ngu yanyba agardinka.	10	28	10	0	28
28	Didbidbunga ngirringgan	1	0	1	0	0
29	Gurijbi gaji mirra, ngarabaji.	21	15	6	15	0
30	Wugbardi ngurlunggu gunju.	1	0	1	0	0
42	Yabu ngiya gijilulu jiyajinka marndanginka.	51	36	51	0	36
49	Alangmiminji irringga daguma alangmiminji.	2	0	2	0	0
51	Nana ngiyinggan manku gurijbirna, ngaba ngin ngawurniji gurijbirna mirra.	276	0	276	0	0
52	Mawulajinka gamany yarru	1	1	1	0	1

FIGURE 1 Sample [incr tsdb()] comparison window

tation software (Oepen et al., 2004). This software takes the 'parse forest' assigned to an input item by a grammar and calculates the set of 'discriminants' (Carter, 1997), or derivation features which distinguish subsets of the parse trees in the parse forest. Though the original purpose of this tool was to assist in annotation for grammar-based treebanks, it is also very handy for grammar development in that it allows the grammar developer to explore the sources of ambiguity. A sample screen shot is shown in Figure 2. In the course of this project, I frequently used this tool to isolate which (syntactic) analyses were 'extra' in one branch or the other.[23]

When the new (aux+verb-cluster) branch had extra analyses, I could use the treebanking tool to discover the source of those extra analyses and from there determine what to change in the grammar. When the extra analyses were on the argument-composition branch—i.e., the new analysis was undergenerating—the strategy was to instead examine the parse chart for the first expected edge (node) that was missing, and then use the LKB's interactive unification functionality to discover where the point(s) of unification failure are. Again, this helps inform what to change in the grammar.

The overall goal of this process is to discover whether the two approaches to the auxiliaries could interact with the rest of the gram-

[23]It was not possible to do this automatically with [incr tsdb()] in this case. First, as noted above, the syntactic derivations are almost all different, so just asking directly which ones did not match is not informative. Second, it is unfortunately not possible to link back to the syntactic derivation for a semantic representation identified as different.

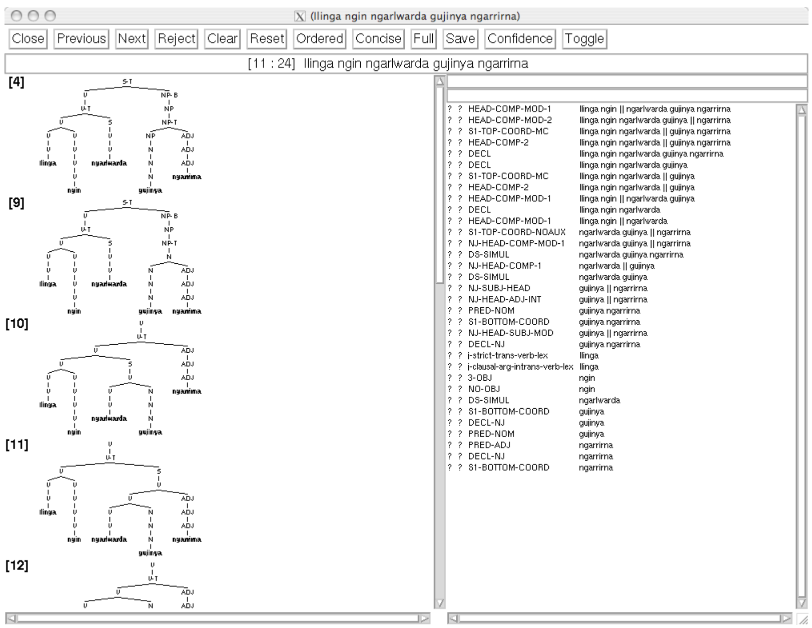


FIGURE 2 Sample tree comparison window

mar to produce the same (semantic) analyses of the development data, to measure the extent to which the rest of the grammar needs to be changed to accommodate the new analysis, and to find particularly interesting interactions between the analysis of auxiliaries and other aspects of the grammar. Of course, the choice of analyses to compare (here, the two approaches to auxiliaries) is somewhat arbitrary, though the analyses chosen target relatively fundamental aspects of the grammar. To carry out similar (exhaustive) comparisons for all of the choice points in the grammar development simply isn't practical.

Instead, I pursued a 'greedy' grammar development strategy: Once I identified the source of a particular difference using the techniques described above, I would think of possible solutions until I found one that seemed plausible. I would then implement that solution and test it against the entire test suite, and examine what changes it caused. If the changes were too great, or showed the analysis to be obviously flawed, I would back it out and try something different. If the changes were minor, I would look to see if they could be addressed with further changes (perhaps to other parts of the grammar). Thus all along the way there are analytical choices that could have been done differently, and it is unlikely that two grammar engineers working with the same data and same framework would come up with exactly the same grammars. At the same time, the test suite data do provide a strong check for plausibility of analyses, and the testing process turns up many unexpected side-effects.

4.2 Example: Secondary predicates

This process can be illustrated with the history of the analysis of example (14).

(14)	Yangula	ng-a	yarru	alanga	gunya-ni.
	NEG	1.SG.S-PST	go	girl.II(NOM)	other.IV-LOC

I didn't move to another (place) (as a) little girl. [wmb]
(Nordlinger, 1998, 178)

This example illustrates the secondary predicate construction: The noun *alanga* 'girl.II(NOM)' is functioning as a modifier of the (unexpressed) 1st-person singular subject, meaning "as a little girl". It can't be picked up as the subject directly, as that would lead to an agreement mismatch, not to mention incorrect semantics. This is handled in the grammar by a construction that takes a noun and creates a predicative nominal that can serve as a modifier of another noun (or of nominal position in a verb's argument structure). This construction, as it was in the baseline grammar, is illustrated in (15).

(15) *secondary-predicate-phrase:*

$$
\left[\begin{array}{ll}
\text{CAT.HEAD} & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textit{noun}} \\
\multicolumn{2}{l}{\text{CASE } \boxed{1}} \\
\text{MOD} & \left\langle \left[\begin{array}{ll}
\text{CAT.HEAD} & \left[\begin{array}{ll} \textit{noun} & \\ \text{CASE} & \boxed{1} \\ \text{INFMOD} & - \end{array}\right] \\
\text{CONT.HOOK.INDEX} & \boxed{2}\left[\begin{array}{ll} \text{PNG.PN} & \textit{non-third} \end{array}\right]
\end{array}\right] \right\rangle
\end{array}\right] \\
\text{C-CONT.RELS} & \langle\ \textbf{pred\_nom\_rel}(\boxed{2}, \boxed{3})\ \rangle \\
\text{ARGS} & \left\langle \left[\begin{array}{ll}
\text{CAT.HEAD} & \left[\begin{array}{ll} \textit{noun} & \\ \text{CASE} & \boxed{1}\textit{nom+erg+acc} \end{array}\right] \\
\text{CONT.HOOK.INDEX} & \boxed{3}\left[\begin{array}{ll} \text{PNG.GEND} & \textit{anim} \end{array}\right]
\end{array}\right] \right\rangle
\end{array}\right]
$$

This construction takes a single daughter (sole element of the ARGS list), which is an animate noun, in nominative, ergative or accusative case. The mother node is still a noun, but now has a non-empty MOD value, making it a modifier seeking to modify a noun with a matching case value. The feature C-CONT is the mechanism by which constructions can introduce semantic information, in this case **pred_nom_rel**, which is a two place relation providing a connection between the two nouns.

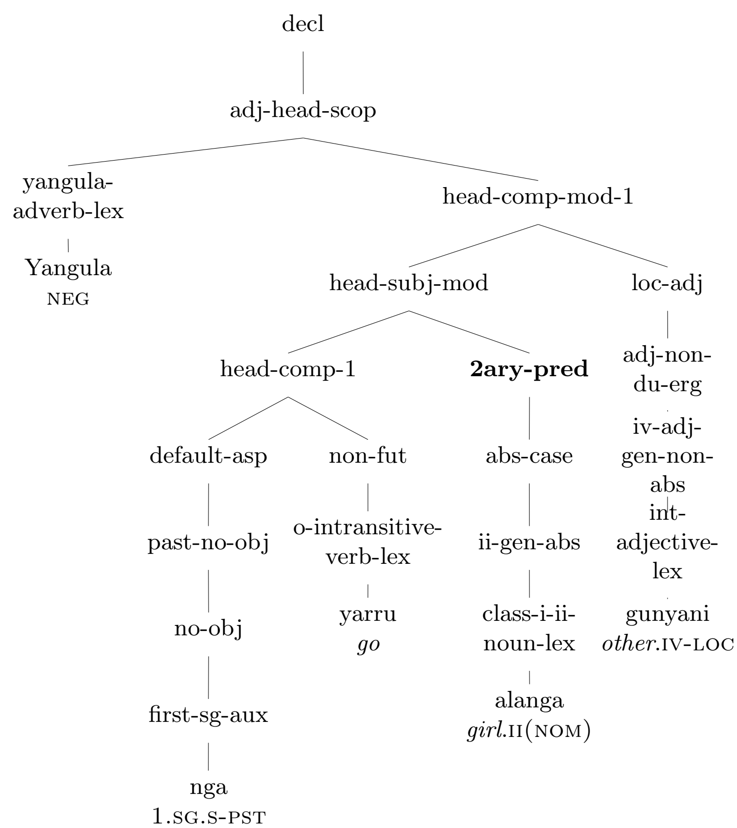
When this construction was initially added to the grammar, it led to a massive increase in ambiguity: in general, head nouns do not need to be overtly expressed. This meant that any noun could serve as a secondary predicate for the position it was supposed to be filling directly. In order to keep a lid on this ambiguity, the construction was constrained in various ways consistent with the available data: it was restricted to animate nouns and non-third person modifiees. In addition, following Nordlinger (1998), the construction is restricted (by the CASE value) to targeting subject or object position (not obliques), and to finite clauses (via the feature INFMOD, which is checked by the modifier-attaching rules).

Example (14) is two-ways ambiguous in the baseline grammar and in the argument-composition branch. The analyses are shown in (16)–(17), with construction names labeling the nodes.[24] The preferred (and probably intended) parse is (16). The other structure is available due to a combination of factors that in fact underlie much of the ambiguity found by the grammar: (i) coordination is unmarked, and achieved

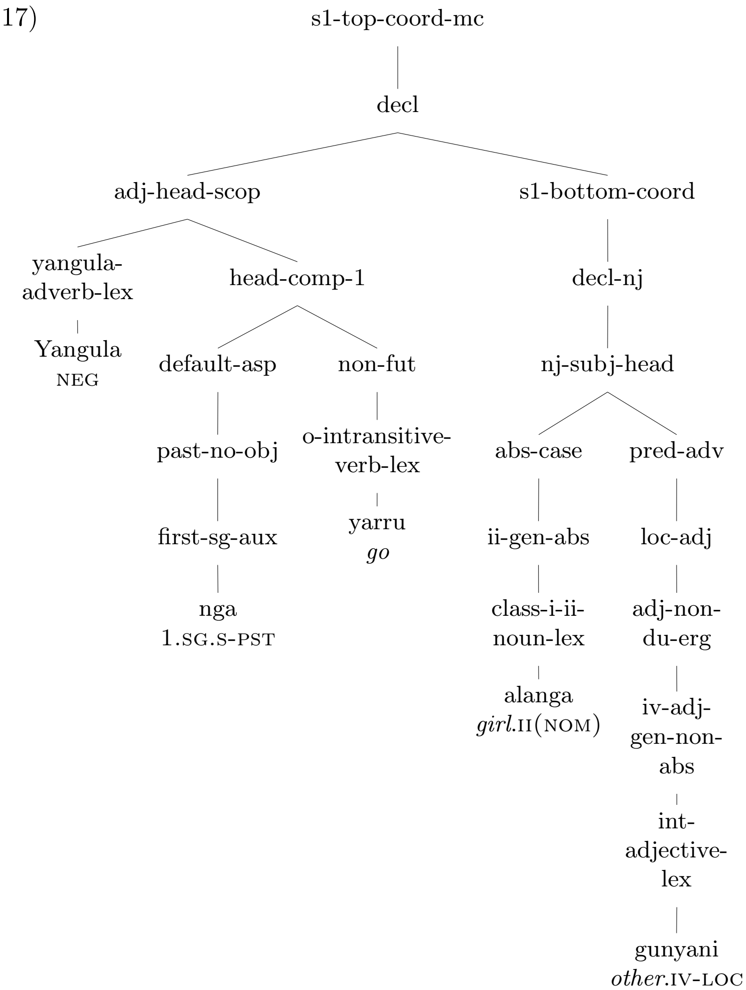
[24]The secondary predicate phrase is called **2ary-pred** and appears in boldface.

through the juxtaposition of clauses and (ii) nouns, adjectives, and adverbials can all head clauses on their own, without any head verb. The semantic representation assigned to (17) can be glossed roughly as 'I didn't move (go) and a girl was in another (place).'.

(16)



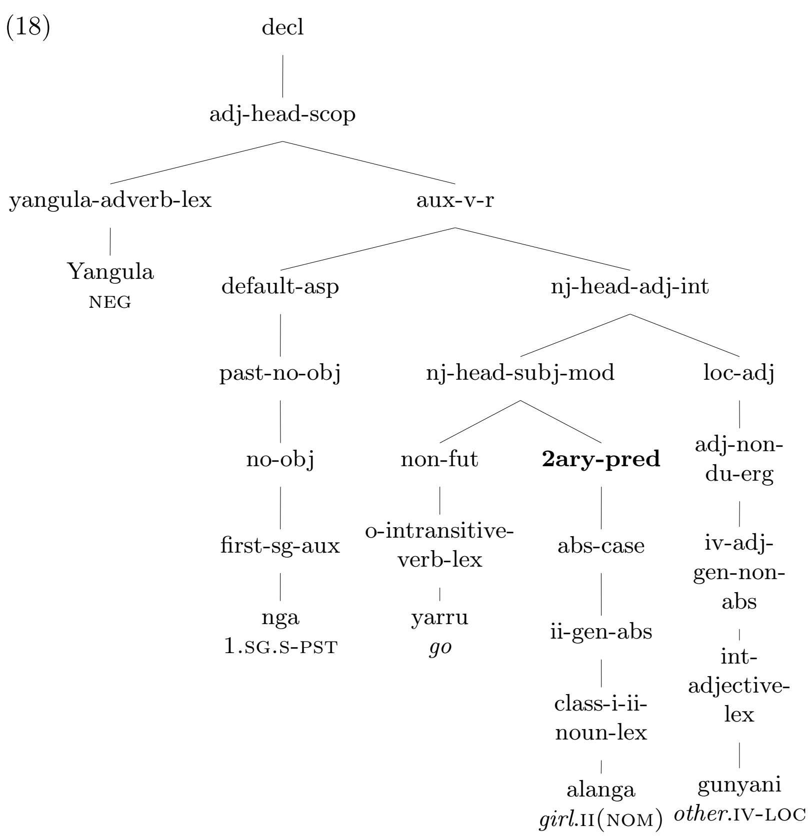
(17)



With the change to the verbal cluster analysis, only the tree analogous to (17) (the dispreferred analysis) was available. Comparison of the profiles turned up this item as a promising one to pursue, as it was only two-ways ambiguous in the (g)old standard. Inspection of the results of the aux+verb-cluster grammar quickly revealed that only the coordination analysis was available. Further investigation of the parse chart showed that the secondary predicate was being constructed, but could not be attached: On the new analysis, it would need to attach within the verbal cluster, i.e., to a non-auxiliary constituent. The LKB's facilities for interactive unification revealed that the source of the failure was the feature INFMOD, which had been posited specifically to restrict secondary predicates to finite (auxiliary-headed) clauses. In particular, the rules for attaching modifiers of arguments of non-auxiliaries (main verbs and non-verbal predicates) check that the modifier is compatible with [INFMOD +], and so weren't firing.

This is still in fact a finite clause, but the new analysis requires that the secondary predicates be able to attach directly to non-auxiliary verbs, as in (18). Lifting the restriction altogether would lead to too much additional ambiguity. Creating a third set of modifier rules for non-auxiliary verbs wasn't desirable. The solution pursued was to replace INFMOD with two binary head features VERBAL and V-MOD. VERBAL distinguishes auxiliaries and main verbs from all other heads. V-MOD allows modifiers to indicate whether they want to attach to verbal projections (independently of whether they are modifying a nominal element within that verbal projection).

(18)



This analysis in fact makes slightly different predictions than the original: allowing secondary predicates on any verbal clause (finite or non-finite), rather than only finite verbal clauses. In terms of the test data, these differences only showed up as an increase of spurious ambiguity: in particular, additional analyses of certain examples where coordinated NPs were serving as the daughter of the secondary predicate construction. This suggested restricting the secondary predicate construction to disallow coordinated NPs. Making that change (via the KEY feature under HEAD) removed those extra parses, but also reduced ambiguity on some other examples. That is, the spurious coordinated-NP-as-secondary-predicate analysis had been in the data all along. Since none of these were the preferred analysis for their parses, this change (disallowing coordinated NPs in the secondary predicate construction) was ported to the argument composition branch.

The purpose of this extended example has been to illustrate the methodology that was used to explore the differences between the grammar branches and then reconcile them, as well as the type of differences that were discovered and the interactions between different parts of the grammar (here, the attachment of verbs and auxiliaries, secondary predicates, and coordination). The example also illustrates the process of incremental development and the ability of representative test suites to shed light on previously unnoticed interactions: By exploring a new analysis of the attachment of secondary predicates (itself required by the new analysis of auxiliaries), and by comparing the two different grammar versions, the (arguably) spurious interaction between NP coordination and secondary predicates was brought to light.

5 Results

5.1 State of the grammars

As of this writing, the two branches of the grammar have the same number of readings in all but four examples. Two of these are cases where the argument composition branch (the old analysis) leads to an error in parsing, while the verb+aux cluster branch (new analysis) successfully completes. The overall 'competence' comparison for the two grammars and the baseline is shown in Table 1.[25] The aux+verb

[25]The example sentences are not evenly distributed across the three length categories. In fact, there are only 3 sentences in the 10–14 word range, and 142 and 659 in the 5–9 and 0–4 ranges, respectively. Table 1 reports raw coverage, rather than 'treebanked' coverage. This means that some of the sentences are only assigned analyses that don't match the gloss given in Nordlinger 1998. On the other hand, since the goal was to ensure that the two grammars have the same coverage, it is appropriate to consider all of the sentences that each can parse, under any analysis.

	sentnces with			**all**
	0-4 words	5-9 words	10-14 words	**sentences**
baseline grammar				
lexical ϕ	5.85	5.69	3.26	**5.78**
analyses ϕ	6.49	37.02	537.00	**12.42**
parsed %	97.1	93.7	33.3	**96.3**
argument-composition grammar				
lexical ϕ	5.85	5.74	3.26	**5.79**
analyses ϕ	5.57	37.37	320.00	**11.47**
parsed %	97.1	94.4	33.3	**96.4**
aux+verb cluster grammar				
lexical ϕ	5.85	5.82	6.23	**5.85**
analyses ϕ	5.57	37.16	454.00	**12.18**
parsed %	97.1	94.4	66.7	**96.5**

TABLE 1 Competence comparison, baseline, argument composition and aux+verb cluster grammars

cluster grammar finds more lexical entries per item successfully parsed and more analyses per item than the argument composition grammar and successfully parses .1% more of the items. In fact, the additional lexical and sentence ambiguity is mostly due to one more (long) example parsing in the aux+verb cluster grammar. Compared to the baseline grammar, both branches show sentence-level ambiguity reduction and a slight improvement in coverage. Again, the improvement in coverage is due to examples previously ending in error receiving parses.

In terms of semantic representations, aside from the four examples with differing numbers of overall readings, there are 16 items with at least partially distinct sets of semantic representations. 14 of these can be attributed to the difference in the representations assigned to non-finite modifier clauses, discussed in §5.3 below.[26] In addition, there are 77 items for which the preferred semantic representation from the baseline grammar is no longer available in the aux+verb cluster branch. 53 of these involve the same non-finite modifier construction mentioned above. An additional 14 can be attributed to a revised analysis of predicative adverbials, intended to improve the semantic representations by including a relation for an unspecified verb that the adverb is modify-

[26]The two remaining items with different sets of assigned semantic representations are still under investigation. One of these two examples did not parse at all in the baseline grammar. The other one did, and the preferred reading is still available in both branches.

ing. Seven more represent a fix to the analysis of the copula which was giving broken semantic representations in the baseline grammar. Only the remaining three represent cases where the aux+verb cluster branch hasn't replicated the coverage of the baseline grammar, and can be fixed with further work on possessive dative nouns used as predicates (two examples) and the exclamative "gubi".

5.2 Performance results

There is a striking difference in performance between the two grammar branches, as shown in Table 2. The aux+verb cluster grammar is far more efficient than the argument-composition branch, requiring 79% fewer parsing tasks, 86.3% less time, and 20.6% less space (memory) to process the same data set.[27] This can be attributed to the difference between bounded and unbounded COMPS lists discussed in §3.1. It is important to note here that the two branches are finding virtually the same set of analyses, as discussed in §5.1. Thus the additional work that the argument-composition branch is doing is almost exclusively the exploration of dead-ends within the search space, i.e., local constituents that never participate in any successful parse.

5.3 Interactions

Overview of changes

With two grammars with (near) identical behavior over the test set, it is now possible to step back and measure what exactly it took to accommodate the new analysis of auxiliaries in the grammar. Starting first with a quantitative analysis, 15 types were added to the aux+verb-cluster branch, and 63 types were modified. This is out of a total of 935 types in the Wambaya-specific portion of the grammar (currently in the aux+verb-cluster branch). Some of these changes were ported to the argument-composition branch, including 1 type added and 22 types changed. Therefore, it can be said that 14 of the added types and 41 of the changed types represent changes to the grammar to implement and/or accommodate the new analysis. The other 23 changes were (somewhat independent) improvements that were suggested by the earlier changes.

These changes touched all parts of the grammar: the definition of phrase structure rules, lexical rules, and lexical entries. They relate to

[27] The apparent slowdown for the aux+verb cluster branch in the longest sentence category can be attributed to it successfully parsing one of the three items in this category whereas the other grammar times out, as the numbers in Table 2 only include successfully parsed examples. Excluding this and the other example which ends in an error for the argument composition branch gives a speed up of 92.1% in both the longest sentence category and overall.

	sentnces with 0-4 words	5-9 words	10-14 words	**all sentences**
argument-composition grammar				
tasks ϕ	2167	27709	157029	**6772**
time ϕ(s)	0.02	0.78	2.66	**0.15**
space ϕ(kb)	4924	13456	48107	**6451**
aux+verb cluster grammar				
tasks ϕ	920	3331	34548	**1423**
time ϕ(s)	0.01	0.03	3.57	**0.02**
space ϕ(kb)	4602	6718	65829	**5124**
reduction				
tasks %	57.5	88.0	78.0	**79.0**
time %	61.6	95.7	-34.2	**86.3**
space %	6.5	50.1	-36.8	**20.6**

TABLE 2 Performance comparison, argument composition v. aux+verb cluster grammars

phenomena including word order, agreement, coordination, the reflexive and reciprocal construction, inherent reflexives, possessives, secondary predicates, predicate adverbials, subject control verbs, various types of adverbs, and imperatives.

Some of the side-effect changes were relatively predictable. For example, the analysis of subject- and object-agreement marking on the auxiliaries needed to be updated. On the argument composition analysis, the auxiliary adopts these arguments as its own, and so can constrain their agreement features (according to its own morphology) in the usual way. On the new analysis, however, the auxiliary takes neither a subject nor any complements beyond the main verb. Therefore, it has to constrain the verb's subject and complement requirements in order to implement agreement. As the only part of the auxiliary which is strictly non-zero in all forms is the subject marker, these are treated as the stems. Object markers are applied as lexical rules (some with zero phonology), as are tense/aspect markers. Some of the tense/aspect markers are sensitive to the person/number of the subject (and the presence/absence of an object) (see Nordlinger 1998:Ch. 5.2). Furthermore, there are separate forms for imperative auxiliaries, which differentiate between intransitive and transitive verbal complements. Thus in addition to the general type for auxiliaries sketched in (7), 15 subtypes defining agreement properties or otherwise constraining the length of the complement's comps list had to be modified, as well as 14 types defining lexical rules which apply to auxiliaries. While these changes were not surprising, the test suite was still useful in identifying all of the types that needed to be modified.

There were also side-effects which were less predictable, and more interesting. I will address two of these here. The first has to do with the reflexive and reciprocal construction, and the second with non-finite subordinate clauses as scopal modifiers.

Reflexive and reciprocal constructions

The Wambaya reflexive and reciprocal construction is marked with a morphological change on the auxiliary. In addition, one of the arguments of the verb is suppressed, and the case of the subject is changed from ergative to nominative (in keeping with the verb being detransitivized). An example is given in (19), where the reflexive/reciprocal morpheme is glossed as 'RR'.

(19) Junmi wurlu-ngg-a jabarrini-ni.
cut 3.DU.A-RR-NF knife.I-LOC

They cut each other with a knife. [wmb] (Nordlinger, 1998, 55)

On the argument composition analysis, this is handled with a lexical rule that applies to the auxiliaries. The rule applies the morphological marker while modifying the auxiliary's valence lists (and constraining the verbal complement to be transitive). This analysis is not available under the aux+verb-cluster grammar: The auxiliaries can still "see" the verb's valence lists, but they can't change them, beyond monotonically adding constraints.

The solution adopted here is a constructional approach. The lexical rule producing reflexive/reciprocal auxiliaries only applies the morphological change and marks them with a diacritic feature [RR +].[28] The morphosyntactic work is done by a new pair of constructions for combining auxiliaries with verbs, which requires [RR +] (while the ordinary ones now require [RR −]). These constructions (one for verb+aux and one for aux+verbal cluster, as before), check that the suppressed argument has not yet been expressed, coindex it with the subject, and construct appropriate valence lists for the mother. The constraints on these constructions are illustrated in (20).

[28] Auxiliaries without the reflexive morpheme go through a non-spelling-changing rule that fills in [RR −].

(20) *rr-aux+verb-rule*

$$\left[\begin{array}{ll}
\text{CAT.VAL} & \left[\begin{array}{ll}
\text{SUBJ} & \left\langle \left[\begin{array}{ll} \text{CONT.HOOK} & \left[\text{INDEX } \boxed{1}\right] \\ \text{CAT.HEAD} & \left[\begin{array}{ll} \textit{noun} & \\ \text{CASE} & \textit{nom}\end{array}\right]\end{array}\right]\right\rangle \\
\text{COMPS} & \langle \boxed{2} \rangle \oplus \boxed{3}
\end{array}\right] \\
\text{HEAD-DTR} & \left[\text{CAT} \left[\begin{array}{ll} \text{HEAD} & \left[\begin{array}{ll} \textit{verb} & \\ \text{AUX} & + \\ \text{RR} & + \end{array}\right] \\ \text{VAL} & \left[\text{COMPS } \langle \boxed{2} \rangle\right]\end{array}\right]\right] \\
\text{NON-HEAD-DTR } \boxed{2} & \left[\text{CAT.VAL} \left[\begin{array}{ll} \text{SUBJ} & \left\langle\left[\text{CONT.HOOK.INDEX } \boxed{1}\right]\right\rangle \\ \text{COMPS} & \left\langle \left[\begin{array}{ll} \text{INST} & - \\ \text{INDEX} & \boxed{1} \\ \text{CASE} & \begin{array}{l}\textit{acc+dat+}\\ \textit{no-case}\end{array}\end{array}\right]\right\rangle \oplus \boxed{3}\end{array}\right]\right]
\end{array}\right]$$

In contrast the non-reflexive aux+verb-rule (see (8)), this rule insists that the non-head-daughter have a COMPS list with at least one element, which is furthermore not expressed ([INST −]). It constructs a COMPS list for the mother that does not include that element. In addition, rather than copying the SUBJ list from the verb, this rule stipulates a SUBJ requirement with a different CASE value. To ensure that the subject is still linked to the right argument position in the verb's semantic contribution, the mother's SUBJ shares its INDEX value with that of the verb ($\boxed{1}$). Furthermore, to capture the reflexive/reciprocal semantics, this index is also linked to the verb's first complement.

The picture is somewhat complicated by a set of inherently reflexive verbs, across a range of valence types (impersonal, intransitive, and subject raising). These are accommodated by marking them as [RR +], and then adding a constraint to the most general type for aux+verb rules that ensures that both daughters have matching RR values. Furthermore, the inherently reflexive verbs must all have dummy syntactic arguments as the first element of their COMPS list in order to meet the constraints on the construction in (20).

This analysis is consistent with all of the data in the development set. However, it makes an unusual prediction, viz., that if the subject is realized within the verbal cluster in a reflexive/reciprocal construction, it will show ordinary case marking (e.g., ergative), because it would be picked up by the verb, which doesn't "know" yet that the clause is marked as reflexive. It seems unlikely that this prediction would be borne out. If it is in fact false, then the solution would be to also have a lexical rule to produce reflexive/reciprocal main verbs (with no morphological effect), which modifies the valence while marking them as [rr +], like the inherent reflexives, so as to ensure the right auxiliary. Curiously, this rule cannot shorten the comps list of the main verb directly, in light of the example in (21).

(21) Garndani-j-ba nyi-ngg-a!
shield-th-fut 2.sg.a-rr-nf
Shield yourself! [wmb] (Nordlinger, 1998, 161)

Here, the transitive imperative auxiliary is used, which requires that the complement verb have a non-empty comps list. Thus if a reflexive lexical rule applied to the main verb and shortened its comps list, then the intransitive auxiliary and not the transitive auxiliary would apply.

To summarize, the reflexive and reciprocal constructions provide an interesting example of the interconnectedness of linguistic analyses within the grammar. The lexical-rule-based analysis of this construction from the baseline grammar was not compatible with the aux+verb cluster analysis of second position auxiliaries. The alternative, construction-based analysis presented here handles the existing data just as well, though as described above, it does make slightly different predictions.

Non-finite modifier clauses

The other example of subtle side-effects that I would like to explore here involves the connection between syntactic structure and semantic composition, particularly in the case of scopal modifiers. This issue came up in connection with non-finite modifier clauses, such as purposives and temporally related subordinate clauses (prior, simultaneous), illustrated in (22)–(23).

(22) Yarru g-any yany-barda manganymi-nka.
go 3.sg.s-np.awy get-inf tucker.iii-dat
He's gone to get some tucker. [wmb] (Nordlinger, 1998, 215)

(23) Ngurruwani ngurru-n mirra gili ngarl-i-ni.
1.pl.inc.nom 1.pl.inc.s(np)-prog sit(nf) here talk-ep-loc
We're sitting here talking. [wmb] (Nordlinger, 1998, 165)

In (22), *yany-barda manganymi-nka* 'get tucker' is marked (through the INF ending on the verb and the dative case on the object) as a subordinate, purposive clause. In (23), *ngarl-i-ni* 'talk' is marked (through the so-called 'locative' ending on the verb) as a subordinate clause expressing simultaneous action.

In both cases, the subordinate clause is analyzed as a scopal modifier. In particular, the constructions which license the subordinate clauses introduce scopal relations (**in+order+to_x_rel** and **temp+overlap-_x_rel**, respectively), which take two scopal arguments, one for the matrix clause and one for the subordinate clause. Ordinarily, once a scopal modifier attaches to a head, any intersective modifiers attaching outside the scopal modifier share its scope (rather than scoping inside) (Copestake et al., 2005). This gives the difference between *Kim happily never left* and *Kim never happily left*.

On the aux+verb-cluster analysis, the subordinate clauses attach either directly to the verb (if they are both to the right of the auxiliary) or to an auxiliary-headed constituent that contains the verb (if either is to the left). In the former case, any intersective modifiers attaching later attach semantically "outside" the scopal relation. In the latter case, the scopal modifier is attached via a head-arg-mod rule which connects it syntactically to the auxiliary-headed constituent but semantically to just the verb. Unlike in ordinary scopal head-modifier rules, this does not change the semantic information available to the next modifier, leaving later attaching modifiers to attach semantically "inside" the scopal relation. In the argument-composition analysis, the subordinate clauses uniformly attach via the head-arg-mod rules.

This remains an area in need of future work in this grammar, beginning with determining what the appropriate semantic representations are. At this point, it seems likely that the analysis in the aux+verb-cluster branch represents an improvement over the argument composition branch, but it needs to be repaired so as not to use the head-arg-mod rules in any of these examples. Whether that is possible, and whether a similar change can be applied in the other branch, remains open to investigation.

5.4 Summary

This section has given an overview of the current state of the grammars, how they differ from each other, and how they differ from the baseline grammar. The overview has highlighted how the analyses of interest—the treatment of auxiliaries—interact with a wide range of other phenomena in the grammar. Furthermore, it has highlighted the role of the development set, the test suite management software, and the ability to compare in detail the analyses of two similar grammars, in both verifying the correctness of different analyses and highlighting the problems they may have.

6 Conclusion

The focus of this paper has been largely methodological, with the details of the Wambaya grammar serving as a case study of how grammar engineering facilitates the formulation, exploration and ultimately the validation of linguistic hypotheses. From a grammar-specific point of view, this experiment has shown that both approaches to Wambaya auxiliaries are in fact (likely) compatible with the other phenomena in the grammar (and in the data set). In addition, the processing data show that the aux+verb cluster approach is much more efficient, and therefore preferable, given comparable grammatical coverage.

The broader, point, however, is that all linguistic hypotheses are situated within some model. Grammar engineering allows us—requires us, actually—to make the models explicit, and this in turn allows for detailed exploration of the ramifications of different hypotheses within those models. The methods used here can generalize to much larger data sets, for languages for which they are available. The larger the data set, the more it can constrain the space of possible analyses.

At the same time, it is important to note that grammar engineering does not replace the ordinary analytical work of syntactic research, but instead augments it: The linguist is still faced with the task of inventing possible alternative analyses. The software merely assists in verifying that the analyses are implemented precisely, that they have the intended behavior on the examples at hand, and that they interact properly with the other analyses already implemented. Also with the linguist rests the responsibility for exploring data types not represented in the development set. The software can assist in calculating the grammar's predictions about these data, but for ground truth, of course, we still need to consult speakers and/or search additional corpora.

Acknowledgments

I would like to thank Rachel Nordlinger for providing access to the data used in this work in electronic form, as well as for answering questions about Wambaya, and helpful comments on the paper itself; Dan Flickinger for the idea behind the aux+verb cluster analysis; and Antske Fokkens, Glenn Slayden, and an anonymous reviewer for helpful comments and discussion. All remaining errors are my own. This material is based upon work supported 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.

References

Bender, Emily M. 2008a. Evaluating a crosslinguistic grammar resource: A case study of Wambaya. In *Proceedings of ACL-08: HLT*, pages 977–985. Columbus, Ohio: Association for Computational Linguistics.

Bender, Emily M. 2008b. Grammar engineering for linguistic hypothesis testing. In N. Gaylord, A. Palmer, and E. Ponvert, eds., *Proceedings of the Texas Linguistics Society X Conference: Computational Linguistics for Less-Studied Languages*, pages 16–36. Stanford: CSLI Publications.

Bender, Emily M. 2008c. Radical non-configurationality without shuffle operators. In S. Müller, ed., *Proceedings of HPSG 2008*. Stanford, CA: CSLI Publications ONLINE.

Bender, Emily M. and Dan Flickinger. 2005. Rapid prototyping of scalable grammars: Towards modularity in extensions to a language-independent core. In *Proceedings of the 2nd International Joint Conference on Natural Language Processing IJCNLP-05 (Posters/Demos)*. Jeju Island, Korea.

Bender, Emily M., Dan Flickinger, and Stephan Oepen. 2002. The Grammar Matrix: An open-source starter-kit for the rapid development of cross-linguistically consistent broad-coverage precision grammars. In J. Carroll, N. Oostdijk, and R. Sutcliffe, eds., *Proceedings of the Workshop on Grammar Engineering and Evaluation at the 19th International Conference on Computational Linguistics*, pages 8–14. Taipei, Taiwan.

Bender, Emily M., Dan Flickinger, and Stephan Oepen. 2009. Grammar engineering and linguistic hypothesis testing: Computational support for complexity in syntactic analysis. Unpublished ms, University of Washington, Stanford University, and University of Oslo.

Callmeier, Ulrich. 2002. Preprocessing and encoding techniques in PET. In S. Oepen, D. Flickinger, J. Tsujii, and H. Uszkoreit, eds., *Collaborative Language Engineering. A Case Study in Efficient Grammar-based Processing*. Stanford, CA: CSLI Publications.

Carroll, John, Ann Copestake, Dan Flickinger, and Victor Poznański. 1999. An efficient chart generator for (semi-)lexicalist grammars. In *Proceedings of the 7th European Workshop on Natural Language Generation*, pages 86–95. Toulouse, France.

Carroll, John and Stephan Oepen. 2005. High efficiency realization for a wide-coverage unification grammar. In *Natural Language Processing – IJCNLP 2005*, pages 165–176. Berling/Heidelberg: Springer.

Carter, David. 1997. The TreeBanker: A tool for supervised training of parsed corpora. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence*, pages 598–603. Providence, RI.

Copestake, Ann. 2002. *Implementing Typed Feature Structure Grammars*. Stanford, CA: CSLI Publications.

Copestake, Ann, Dan Flickinger, Carl Pollard, and Ivan A. Sag. 2005. Minimal Recursion Semantics: An introduction. *Research on Language & Computation* 3(4):281–332.

Green, Ian and Rachel Nordlinger. 2004. Revisiting Proto-Mirndi. In C. Bowern and H. Koch, eds., *Australian Languages: Classification and the Comparative Method*, pages 291–311. Amsterdam: John Benjamins.

Hinrichs, Erhard and Tsuneko Nakazawa. 1990. Subcategorization and VP structure in German. In S. Hughes and J. Salmons, eds., *Proceedings of the Third Symposium on Germanic Linguistics*. Amsterdam: Benjamins.

Meurers, Walt Detmar. 1999. Raising spirits (and assigning them case). In W. Abraham, ed., *Groninger Arbeiten zur Germanistischen Linguistik 43*, pages 173–226.

Müller, Stefan. 2008. Depictive secondary predicates in German and English. In C. Schroeder, G. Hentschel, and W. Boeder, eds., *Secondary Predicates in Eastern European Languages and Beyond*, no. 16 in Studia Slavica Oldenburgensia, pages 255–273. Oldenburg: BIS-Verlag.

Nordlinger, Rachel. 1998. *A Grammar of Wambaya, Northern Australia*. Canberra: Pacific Linguistics.

Oepen, Stephan. 2001. [incr tsdb()] — Competence and performance laboratory. User manual. Technical report, Saarbrücken, Germany.

Oepen, Stephan, Emily M. Bender, Ulrich Callmeier, Dan Flickinger, and Melanie Siegel. 2002. Parallel distributed grammar engineering for practical applications. In *Proceedings of the Workshop on Grammar Engineering and Evaluation at the 19th International Conference on Computational Linguistics*. Taipei, Taiwan.

Oepen, Stephan, Daniel Flickinger, Kristina Toutanova, and Christopher D. Manning. 2004. LinGO Redwoods. A rich and dynamic treebank for HPSG. *Journal of Research on Language and Computation* 2(4):575 – 596.

Pollard, Carl and Ivan A. Sag. 1994. *Head-driven Phrase Structure Grammar*. Chicago: Chicago University Press.

Przepiórkowski, Adam. 1999. *Case Assignment and the Complement-Adjunct Dichotomy: A Non-Configurational Constraint-Based Approach*. Ph.D. thesis, Universität Tübingen, Germany.

Sag, Ivan A., Thomas Wasow, and Emily M. Bender. 2003. *Syntactic Theory: A Formal Introduction*. Stanford, CA: CSLI, 2nd edn.