A Categorial Grammar fragment for Lushootseed

Deryle Lonsdale
BYU Linguistics Department

This paper establishes a basic framework for a Categorial Grammar (CG) description of Lushootseed morphosyntax and semantics. After a brief review of English CG principles, a preliminary attempt is made to apply these fundamentals to Lushootseed free and bound morphemes. Several sample derivations are presented, including one rather complex syntactic ambigiuty previously noted in the literature with its alternative descriptions. Once the syntactic elements have been put in place a short overview is presented of how semantic interpretation can be established in tandem with syntactic composition principles. Finally, the implementation of this framework within the Attribute Logic Engine environment is discussed, including a sample rule and parser output. Finally ongoing efforts, future work, and possible applications are mentioned.

1 Introduction

A categorial grammar (CG) is a type of grammatical formalism that presents an alternative explanation of constituency to that presented by more traditional phrase-structure rule-based grammars. CG's have been useful in describing languages with complex morphology and syntax, particularly where the division between the two areas is somewhat blurred. For example, CG descriptions have been proposed for Turkish (Bozşahin and Göçmen, 1995) and Korean (Cha et al., 1999), both of which are agglutinative languages.

This paper proposes a preliminary analysis of Lushootseed in CG, providing examples of the basic operations and showing how the principles can be applied to analyze the language's syntax. It then mentions how semantic analysis can also be carried out. Finally, an implementation of a CG parser for Lushootseed is discussed.

2 Background

The Chomskyan approach to syntactic description is based on the notion of rule-based phrase-structure grammars. An alternative set of grammatical formalisms is called lexicalized grammars, and one type of lexicalized grammar involves the use of categorial grammars. The basic element of a categorial grammar is the lexicon; once it has been established, universal syntactic principles operate on increasingly complex phrases and clauses to determine the complete set of constituents of a sentence. Categorial grammars were introduced in the 1940's (Ajdukiewicz, 1935; Bar-Hillel, 1953) and have enjoyed somewhat of a resurgence in the last decade. The nature of categorial grammars is such that they rely on a highly compositional, type-driven approach to syntax similar to that used in some computer programming languages. CG's are also particularly useful in performing semantic analysis and in describing the syntax/semantics interface (Carpenter, 1997); in this paper, we first propose a syntactic approach, and then relate it to a semantic description.

Lushootseed is a Coast Salish language whose syntax and semantics has been described to some degree in a number of theoretical frameworks: X-bar syntax, traditional grammatical approaches, cognitive grammar, HPSG, and LFG. Given this language's nontrivial morphosyntax, challenges in describing word and phrase-level structure are abundant. This paper presents a preliminary account of Lushootseed syntax in CG. In the next section, an overview of CG as it relates to English is presented as background; in subsequent sections attention is turned to the Lushootseed language description.

2.1 English categories

Categories in CG have been traditionally referred to as the set of basic categories, BASCAT, which for English includes the members np, n, and s (for noun phrase, noun, and sentence respectively). While rather minimal, this set can be bootstrapped to create an infinite number of more complex categories. Complex categories are the result of composing together more basic categories in a particular type of operation called function composition. The set of complex categories generated over BASCAT we will call CAT(BASCAT), and its elements are derived three ways:

- directly from BASCAT (i.e. $\alpha \in$ CAT(BASCAT) if $\alpha \in$ BASCAT)
- via forward application (i.e. $\alpha/\beta \in$ CAT(BASCAT) if $\alpha,\beta \in$ CAT(BASCAT))
- via backward application (i.e. $\alpha\backslash\beta \in$ CAT(BASCAT) if $\alpha,\beta \in$ CAT(BASCAT))

One observation is important: as categories are composed together via concatenation, a delimiting slash (either forward or backward) specifies the appropriate directionality of application.

By way of illustration, it should be noted that BASCAT does not contain a category for determiners. In other words, the simple category label for determiners (e.g. Det in other grammatical approaches) does not exist in CG.

$$\frac{\begin{array}{cc}\text{lex}_1 & \text{lex}_2 \\ \alpha/\beta & \beta\end{array}}{\alpha} \quad\Bigg|\quad \frac{\begin{array}{cc}\textit{the} & \textit{dog} \\ \text{np/n} & \text{n}\end{array}}{\text{np}} \quad\Bigg\|\quad \frac{\begin{array}{cc}\text{lex}_1 & \text{lex}_2 \\ \beta & \alpha\backslash\beta\end{array}}{\alpha} \quad\Bigg|\quad \frac{\begin{array}{cc}\textit{dogs} & \textit{bark} \\ \text{np} & \text{s}\backslash\text{np}\end{array}}{\text{s}}$$

Figure 1: Two examples of derivations: on the left is the forward application schema with an English example, and on the right is the backward application schema with an example.

This is because this category can be specified via forward application; we will consider it to be a derived category with the specification np/n. This means that we consider determiners to be (potential) noun phrases which must be satisfied by finding a noun (n) to the right. This is called a derivation (based on the way derivations are shown in formal logic), and the construction of a syntactic constituent is shown by drawing a line beneath the words which are involved in the function application.

For example, the left-hand side of Figure 1 shows the derivation of the English noun phrase *the dog*. The words *the* and *dog* are involved in forward application, since the np/n looks forward for the n it is seeking. A line is drawn underneath both words, and under the line we indicate the category which results from the forward application, in this case np.

For an example of backward application, we begin by observing that there is no basic category for a verb phrase in BASCAT. In fact, the very existence of such a node has been the topic of extended debate (i.e. the configurationality debate) among linguistic theorists, some claiming it must exist, and others denying its existence or at least its usefulness. In a CG account for English the notion of an intransitive verb phrase can be represented as the derived category s\np, meaning that it is a (potential) sentence (s) which must seek "backward" for an np (which will be the subject). This process of arriving at a sentence involves backward application, as shown in the right-hand portion of Figure 1. Here the verb "bark" (of category s\np) looks backward for its subject, "dogs" (an np).

Note that there is also no label for conjunctions in BASCAT; this, too, is a derived category (in this case np\np/np since it looks to both sides for an np, and yields an np. Similarly, complementizers (e.g. *that, whether*) have the category s/s since they look to the right for a sentence.

Basic adjectives will have the category n/n, yielding a noun when they combine with a noun to the right (see Figure 2, top). Prepositions are another type of noun modifier; they receive an np to the right and then combine with a noun to the left (Figure 2, bottom).

Increasingly complex categories can be used in successive derivations to account for further methods of syntactic description. For example, English transitive verbs (see Figure 3) can be represented as (s\np)/np, meaning they are sentences which first look forward for an np (the object), and then backward for another np (the subject). Other English categories include auxiliary verbs, modal verbs, copulas, and infinitives. In all cases, these can be assigned the complex

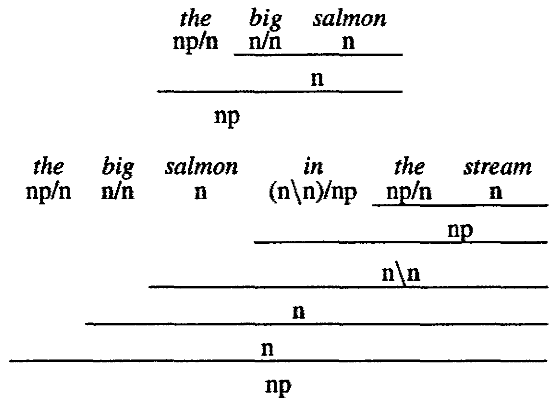


Figure 2: Derivations for a simple noun phrase with adjectival modification (top) and for one with a prepositional phrase modifier (bottom).

category (s\np)/(s\np). Derivations using these categories are left to the reader but follow straightforwardly from the principles enumerated so far.

Many verbs subcategorize for certain types of complements, and this information is captured via their lexical categories; verbs such as *know* which take a sentential complement are of category s\np/s, equi verbs (e.g. *want, promise, persuade, appeal*) are of category s\np/(s\np/np), subject raising verbs (e.g. *tends, seems*) are of category s\np/(s\np), and object raising verbs are of category s\np/(s\np/np).

Much more could be discussed with respect to English CG analysis, but what has been covered establishes the necessary groundwork for the next sections. Further details for English are available elsewhere (Carpenter, 1992; Carpenter, 1997).

3 Describing Lushootseed

In this section we address various aspects of the Lushootseed language within CG including categorizing the basic lexical items, morphemes and syntactic structures. Adaptations are mentioned when necessary to address properties of the language that are not relevant to English. Throughout this section only the basics of CG description are referred to; more technical work is possible but beyond the scope of this paper. For the most part, sentences and glosses are from some of the standard Lushootseed sources (Hess, 1995; Hess, 1998; Bates et al., 1994)[1].

[1] whose authors are not responsible for any errors in this paper

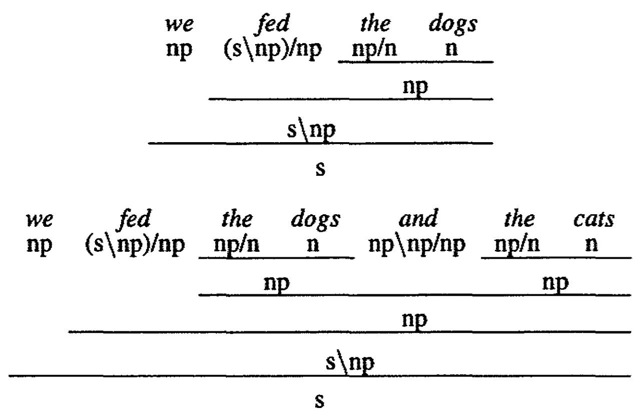


Figure 3: Derivations for a simple transitive sentence (top) and for one with a conjoined object (bottom).

3.1 Basic lexical entries

As with the English examples above, individual lexical items are assigned categories, either basic or complex. In many cases they are somewhat equivalent to English. Following are a few of the most elementary categories:

- nouns [n]: bad, č̓ƛ̓aʔ, etc. (English: *father, rock, etc.*)
- pronouns [np]: čəd, čəxʷ, čələp, ʔəlgʷəʔ, ʔəca, dəgʷi, etc. (English: *I, you, they, etc.*)
- determiners [np/n]: tiʔiɬ, kʷi, kʷsi, etc. (English: *the, etc.*)
- adjectives [n/n]: hikʷ, etc. (English: *big, etc.*)
- conjunctions [np\np/np]: ʔi (English: *and*)
- sentential adverbs [s/s]: gʷəl, etc. (English: *then, etc.*)

As shown in Figure 4, with these categories it is already possible to analyze simple noun phrases.

To verbal roots, which are usually inherently intransitive, the following category will be assigned:

- verbs [s/np]: saq̓ʷ, kʷatač, etc. (English: *fly, climb, etc.*)

Note that, as in English, no vp node is required, a desirable situation for Lushootseed description given its canonical clausal constituent order. Instead, the category s/np will represent basic (intransitive) verbal roots; this category's interpretation is that these verbs are sentences looking for a subject to the right.

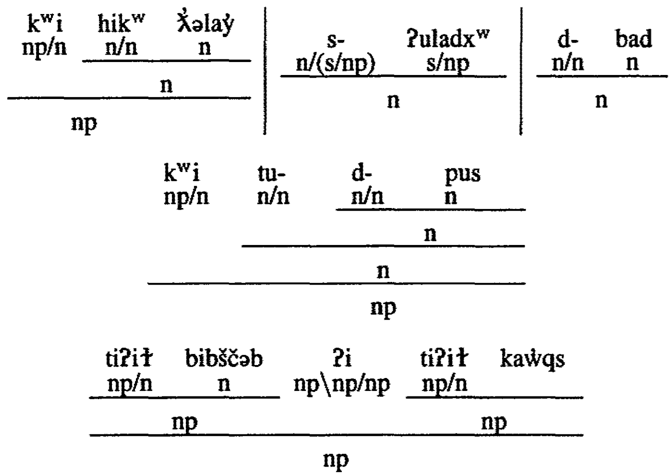
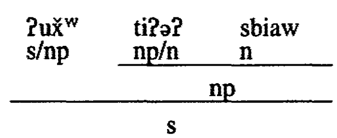


Figure 4: Four simple nominal derivations (top: English *the big canoe*, *salmon*, *my father*; middle: English *my late aunt*), and derivation of conjoined noun phrases (bottom: English *Mink and Raven*).

With the inventory of lexical categories as discussed so far it is already possible to analyze simple sentences. In the following example the determiner combines via forward application with the noun, and then the verb combines via forward application with the resultant noun phrase to create a sentence (English *The coyote goes.*):



An important point to note is that cross-categorial derivation is very common in Lushootseed; much discussion has taken place on whether there is in fact a verb/noun distinction in this and related languages. Consider, for example, the word "hikʷ", which can function as a predicate ("hikʷ ti spaʔc") or as a modifier ("ʔux̌ʷ ti hikʷ spaʔc"). In the former case, the word would have the category s/np, whereas in the latter it would have the category n/n. Creation of multiple lexical entries in such situations (e.g. the English homograph "fish" as a noun and a predicate) is common in CG (and in some circles is somewhat criticized); we assume this approach for Lushootseed lexical roots where the category is unclear. This is less of a problem than would appear at first: the categories of adjacent words and morphemes tend to conspire to only allow appropriately categorized variants of such homographs. The immediate context significantly pares down the search space of all possible category combinations.

Another example of category-based multiplicities in lexical encoding is apparent in prepositions. Depending on their function, prepositions (ʔə, ʔal, liɬ, tul̓, etc.) have several possible categories, two of which are:

- (n\n)/np when attached to a noun
- (s\s)/np when attached to a completed predicate

Figure 5 shows sample prepositional attachments.

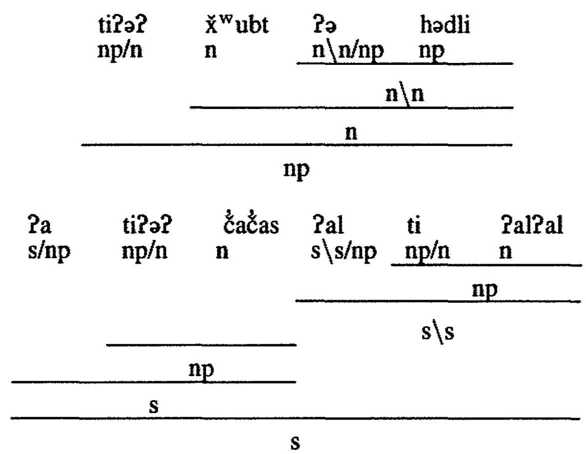


Figure 5: Sample prepositional phrase attachments to a noun (top: English *Henry's paddle*) and to a verbal predicate (bottom: English *There is a/the child at the house.*).

3.2 Lushootseed morphosyntax

Of course, few Lushootseed sentences are created only with free lexical items: morphological marking is almost always present in sentences. In this section affixation and other morphological processes are discussed, as well as certain related syntactic principles. As was the case in the previous section, CG descriptions of English morphology will motivate most of the category assignments for Lushootseed.

3.2.1 Affixation

We first consider the morphological structure of Lushootseed nominal complexes. Lushootseed nouns can take various prefixes and suffixes, which in CG can be assigned derived categories corresponding to their role in word formation. Nominal-modifying affixes, analogously to their full lexical counterparts, are assigned the category n/n. Accordingly, the possessive affixes and particles look righward or leftward for a noun, and return a noun. In addition, the nominalizing bound prefix *s-* is obligatory in connection with some nominal roots; it could be assigned the category n/n. We will assume that lexical

q̓^{w}ibi	-d	čəd	ti	d-	ʔiišəd
s/np	(s/np/np)\(s/np)	np	np/n	n/n	n
	s/np/np	np	np/n	n	
				np	
	s/np				
		s			

Figure 6: The parse for a transitive predicate (English: *I am disembarking my friend.*).

suffixes also produce nominals when attached to a nominal root. Thus we have the following assignments:

- nominalizer [n/n]: s-
- possessive prefixes [n/n]: d-, ad- (English: *my, your*)
- lexical suffixes [n\n]: =alʔtx^{w}, =šəd, etc. (English: *house, foot, etc.*)
- possessive suffixes/predicates [n\n]: -s, -čəɬ, -lap (English: *his/hers/theirs, our, your, etc.*)

Lushootseed has rich verbal morphology, with aspectual and tense marking primarily via prefixation, and valency markings via suffixation. As above, we apply affixes to the base with appropriate forward or backward function application. Assuming that the basic category of a predicate is s/np, we can assign a category to tense and aspectual prefixes accordingly:

- tense/aspect markers [(s/np)/(s/np)]: ʔu-, ʔəs-, tu-, ɬu-, lə-, bə-, etc.

Note that many of these prefixes can also attach to nouns and to nominalized verbal complexes, necessitating the following additional lexical categories:

- tense/aspect markers [n/n]: ʔəs-, tu-, ɬu-, lə-, bə-, etc.
- nominalizer [n/(s/np)]: s-

Assignment of valency-related suffixes is more complicated. Consider first the transitivizer suffix, -d/t. It has the effect of changing an intransitive predicate to a transitive one, and in the process adding another nominal argument to the verb frame. The same is true for related suffixes which entail transitivity: reflexives (-cut), and out-of-control constructions (-dx^{w} and related forms). The appropriate category to assure transitivity-inducing operations is as follows:

- transitivizer morphemes [(s/np/np)\(s/np)]: -(ə)d, -cut, -dx^{w}, etc.

This means that such suffixes look backward for an intransitive predicate, and after combining become a transitive predicate, one that looks forward for two np's (the agent and patient/theme arguments). Figure 6 shows a transitive clause parse.

The case of the middle voice marker, -(ə)b, is somewhat problematic. For the purposes of this discussion, we will assume that the middle takes a transitive predicate and reduces its arity by one argument; if the reduced argument (the agent) needs to be expressed, it will be done obliquely via a prepositional phrase and hence would require another derived category (which we do not address here):

- middle voice marker [(s/np)\(s/np/np)]: -(ə)b

Secondary suffixes and stem-extenders are also common in verbal morphology. These allow normally intransitive verb frames to take another argument, whether a direct object, benefactive, resultative, etc. Their form is -i, and they almost always combine with other suffixes (e.g. the transitive suffix that it precedes). In this paper we will assume that these affixes act analogously to the other mentioned verbal ones:

- stem extenders [(s/np)\(s/np)]: -i, -yi-, -bi-, etc.

Clitics such as -əxʷ that occur after the verb valency suffixes will require encodings for different predicate categories:

- after intransitive, middle verb complexes [(s/np)\(s/np)]: -əxʷ
- after transitive verb complex [(s/np/np)\(s/np/np)]: -əxʷ

as shown in the following example (English: *He chews on the heart of the whale.*):

```
ƙaw    -dxʷ          -əxʷ          ∅     tiʔiɬ   sčaliʔ  ʔə         tiʔiɬ   čxʷəluʔ

       (s/np/np)\    (s/np/np)\
s/np   (s/np)        (s/np/np)     np    np/n    n       (n\n)/np   np/n    n
-------------------
      s/np/np                                            ------------------------
                                                                     np
---------------------------------------          ----------------------------------
                s/np/np                                         n\n
                                          s/np   ----------------------------------
                                                                 n
-----------------------------------------------------------------------------------
                                                   np
-----------------------------------------------------------------------------------
                                          s
```

Taking stock of these various categories, it is possible to parse increasingly complex phrases, clauses and sentences. Figure 7 shows the derivation for a sentence with a nontrivial predicate.

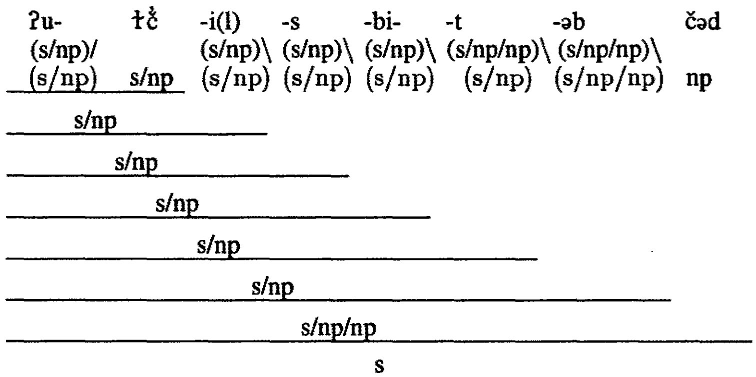


Figure 7: The parse for a sentence with a highly inflected verbal predicate (English: *He came to my place to see me.*).

3.2.2 Empty categories

The use of empty categories in CG is relatively widespread, and allows for processing of such phenomena as ellipsis, node raising, discontinuous constituents, and anaphora. This could be a problem, as unrestricted positing of empty categories is undesirable; in practice, though, the combinatorics are significantly reduced via constraints on the surrounding elements' categories.

Treatment of zero or null anaphors, which are frequent in Lushootseed, is typically handled in CG grammars by positing the presence of a null constituent of category np. Accordingly, we assume this technique for Lushootseed sentences.

Relative clauses in Lushootseed do not always require complementizers or relative pronouns, which can be problematic. Note that they are sometimes optional in English, as in the sentences *I think _ I will go home.* and *The dog _ I chased bit me.*, respectively. For this paper we assume that an empty category of type (s/np)\(s/np)/s or n\n/s (respectively) is used in such cases in Lushootseed.

Up to now in this paper the assumption has been that Lushootseed clauses consist of a verbal complex followed by arguments and/or adjuncts; this is, of course, an oversimplification. Instead, a general schema for clauses is: predicate, followed by arguments and adjuncts. The predicate can in fact be of any category.

Accounting for such structures is not an insurmountable problem for CG, but one which would involve more CG machinery than appropriate for this paper. We will instead assume the presence of an empty copula-like item between the predicate and the other material in the clause, but only when the predicate is non-verbal. This has the effect of coercing the category of the non-verbal predicate to one that is verbal. The category of such linking elements will vary:

- empty linker for phrasal nominal predicate [(s/np)\np]:

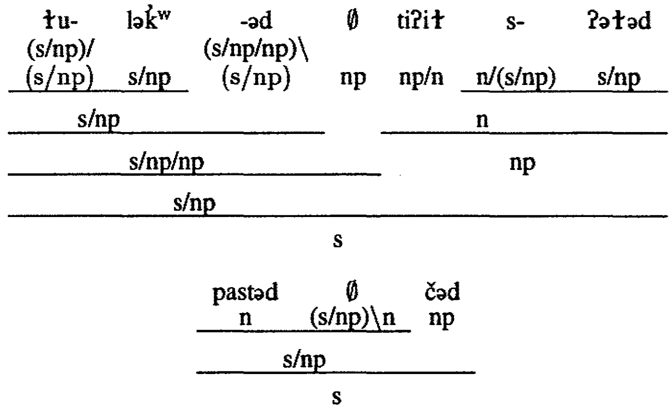


Figure 8: Positing an empty category: for zero anaphors (top: English *He intended to eat that food (and he did so).*) and nominal predication (bottom: English *I am a white person.*).

- empty linker for bare nominal predicate [(s/np)\n]:
- empty linker for adjectival predicate [(s/np)\(n/n)]:

Figure 8 shows the derivation of two sentences using empty categories: one for zero anaphor and the other for predicating nominals.

3.3 Lushootseed semantics

A commonly adopted paradigm for semantics holds that it is established interpretively; that is, syntax is prior, and therefore an understanding of the syntax is necessary for semantic understanding, the latter being established by associating conceptual structures which have been mapped from the syntax. Categorial Grammar differs from the interpretive approach in that it assumes that syntax and semantics are established in tandem; that syntactic and semantic descriptions can be developed in parallel based on the same operations introduced above.

Following the standard practice in CG, we will adopt a predicate calculus foundation for representing the semantics of Lushootseed expressions. Concepts will be expressed as variablized expressions with appropriate arity: intransitive predicates take one argument, transitives take two, and so forth. Lexical items and morphemes in CG, as we have seen earlier, each are assigned a syntactic category. Usually, though, there is another component to each category—a semantic one—that is used in describing the semantics of an expression.

As constituents are combined in CG according to the syntactic principles discussed above, semantic representations are built up compositionally. To accomplish this, function application (forward or backward) is applied to the semantics of each constituent when combining the two together, just as it is in the syntax[2].

For example, consider the sentence "Dave ate a clam." It discusses two entities, Dave and some clam; we represent each with a variable, x for Dave and y for the clam. We also need to describe each of these entities in terms of their membership in some semantic class; for our purposes we will simply assume the use of two class predicates, "dave" and "clam", which include all people of named "Dave" and all clams, respectively. Therefore the entity Dave can be described semantically as dave(x), and the other one clam(y). Both are expressions containing open veriables; these variables will eventually be instantiated with real-world entities. The action of eating (ignoring tense) is a transitive predicate represented by its lexical form and two arguments: eat(y)(x). Note that for English the most prominent predicate argument (e.g. agent/subject) is listed last, as is standard practice in CG. Given these conventions, we can represent the semantic content of the sentence as:

dave(x) & clam(y) & eat(y)(x)

meaning "entity x is Dave, entity y is a clam, and entity x eats entity y".

In this paper we include two simplifying assumptions in displaying and manipulating semantic information for convenience in presenting the data. First, we will use combinator (or curried or Schönfinkelized) forms of predicate terms. These forms are simply standard representations re-cast without free variables. For example, instead of writing dave(x) we will use the combinator DAVE. The second assumption is that we will use English predicate names in Lushootseed semantic expressions. Of course, this can only be an approximation to the real Lushootseed meaning, but is necessary for ease of presentation. Figure 9 shows sample Lushootseed expressions and their variablized and combinator terms, as specified by these assumptions. Note that in representing Lushootseed semantics we will list arguments according to their thematic importance, with agent first (a slight departure from traditional CG work).

Figure 10 shows a derivation illustrating tandem development of both syntactic and semantic structures. At each stage of the derivation the category consists of two colon-delimited components: the syntax (on the left) and the semantics (on the right) for the constituent in question. Function application in both domains is completely parallel.

3.4 Ambiguity

Ambiguity at the morphological and syntactic levels is possible in Lushootseed, though not as frequently as in English with its pervasive fusional morphology and morphological part-of-speech conversions. A principled

[2] In fact, the process is much more formal and detailed than this, involving the calculus of operators and application schemas for lambda expressions. We will not explore the details in this paper.

Lushootseed expression	Predicate expression	Combinator
hədli	henry(x)	HENRY
sbiaw	coyote(x)	COYOTE
ləƛ̕ʷəd čəd	eat(I)	EAT(I)
ləƛ̕ʷəd čəd ləpəskʷi	biscuit(y) & eat(I)(y)	EAT(I)(BISCUIT)
pastəd	whiteperson(x)	WHITEPERSON
pastəd ti hədli	henry(x) & whiteperson(x)	WHITEPERSON(HENRY)

Figure 9: Semantic expressions: Lushootseed text, English predicates, and combinators (some resulting from function application).

```
      ʔu-            ʔux̌ʷ        tiʔəʔ       sbiaw
  (s/np)/
  (s/np):PERF    s/np:GO     np/n:DEF    n:COYOTE
 ____________________________
      s/np:PERF(GO)             ________________________
                                    np:DEF(COYOTE)
 _______________________________________________________
      s:PERF(GO(DEF(COYOTE)))
```

Figure 10: Semantic derivation showing syntax:semantic categories at each stage of the process (English: *The coyote went.*).

CG-based approach permits description of all possible readings when ambiguity exists.

Previous work (Bennett and Beck, 1998) has documented a remarkable and impressive example of syntactic ambiguity in Lushootseed for a narrative containing the utterance(s):

huyutəbəxʷ dxʷʔal kʷi gʷəsəsčəbaʔs tiʔəʔ cədiɬ kʷagʷičəd tiʔəʔ sɬiltəbs.

There are arguably two possible syntactic construals for this utterance fragment; prosodic cues (which are not addressed in this paper) presumably serve to disambiguate them. The first construal involves one sentence with a subject relative clause and a manner adjunct clause, translated into English as: *This elk which had been given him was fixed up so it could be backpacked.*. The second reading involves two separate sentences, rendered into English as: *It was fixed up so that it could be backpacked. What had been given him [was] this elk.*.

Both syntactic readings for this utterance fragment can be derived given the categories posited above. Figures 11 and 12 show the two respective parses. Derivation of the semantics (which is left to the reader) will show that the different syntactic parses lead to semantically divergent interpretations consistent with the compositional structure derived in tandem from the syntax.

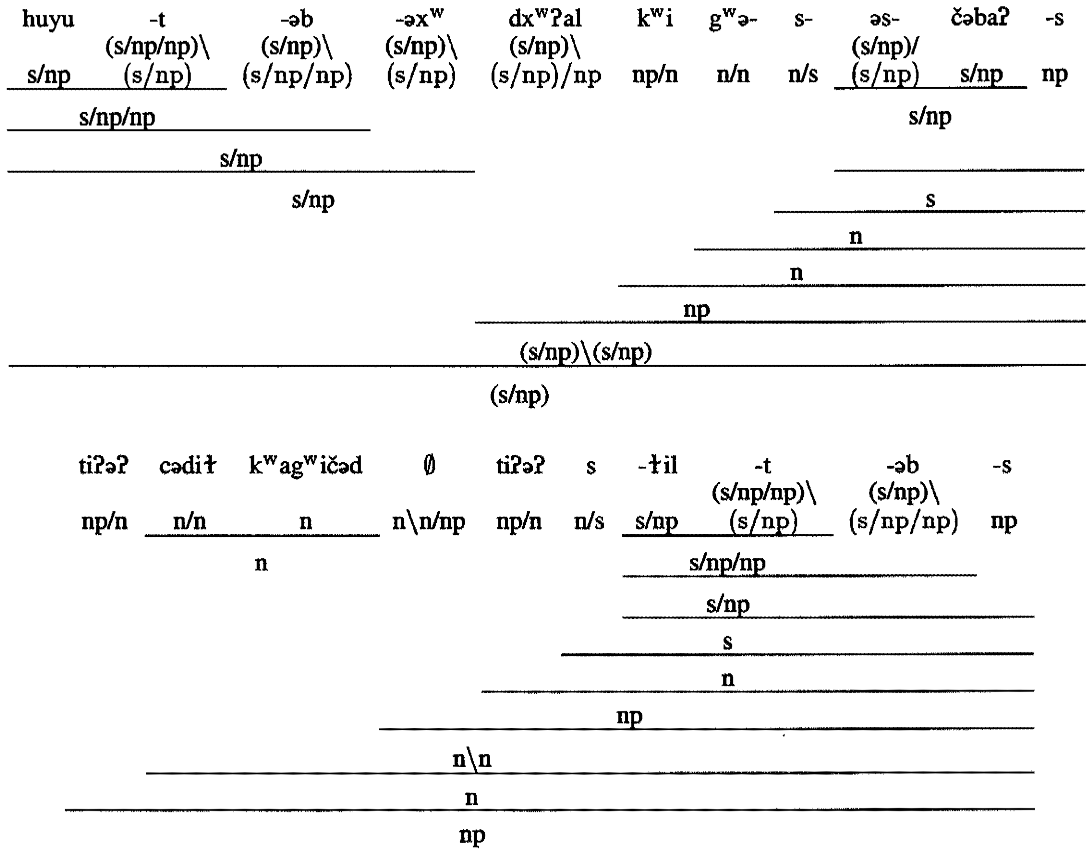


Figure 11: The two halves of one alternative parse for the syntactically ambiguous utterance discussed in Section 3.4 (English: *This elk which had been given him was fixed up so it could be backpacked.*). Both combine via forward application to create one sentence.

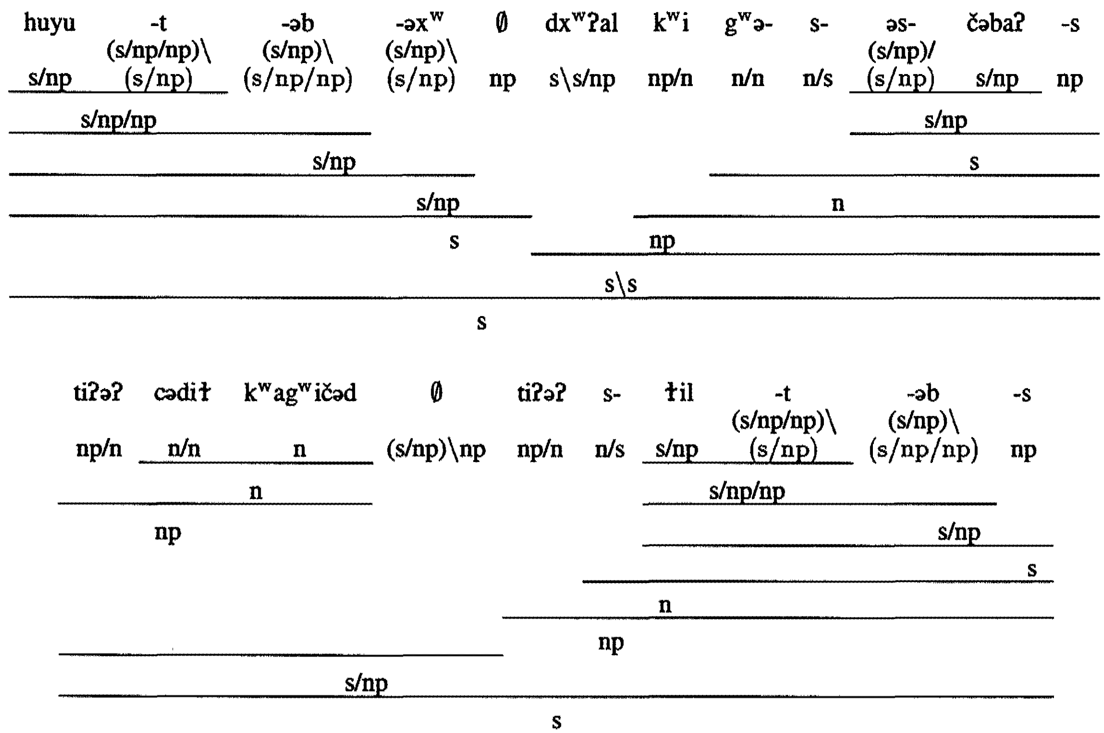


Figure 12: Another alternative parse for the syntactically ambiguous utterance discussed in Section 3.4 (English: *It was fixed up so that it could be backpacked. What had been given him [was] this elk.*). Two separate sentences result.

```
tv(Rel) macro
 synsem:(forward,
   arg:(syn:np,
        sem:Y),
   res:(forward,
        arg:(syn:np,
             sem:X),
        res:(syn:s,
             sem:(Rel,
                  agent:Y,
                  theme:X)))),
 @ quantifier_free.

rec[7ugWECEd,CEL,ti7E7,hikW,spa7c].
QSTORE e_list
SYNSEM basic
   SEM DEF
       RESTR and
             CONJ1 BEAR
                   ARG1 [0] individual
             CONJ2 BIG
                   ARG1 [0]
       SCOPE SEEK
             AGENT pro1p
                   THEME [0]
       VAR [0]
   SYN s
```

Figure 13: A Lushootseed grammar rule for transitive verbs (top), and one parse returned for a sentence (bottom).

3.5 Implementing the grammar

All of the rules described above, and several more not mentioned in this paper, have been implemented in a computer system that can parse the syntactic and semantic content of Lushootseed sentences. The system was developed using ALE, the Attribute Logic Engine (Carpenter and Penn, 2001). ALE has been widely used for implementing computational grammars in such formalisms as CG, HPSG, and LFG. Developed within the Prolog programming language, ALE exhibits multiple-goal-directed problem solving, pattern matching, and backtracking capabilities. This is useful when processing ambiguous structure since all possible interpretations can be pursued and generated by ALE.

Input is entered into the system according to a romanization scheme used in previous computational work with Lushootseed (Lonsdale, 2001). Figure

13 (top) shows an example of a Lushootseed grammar rule for a transitive verb, declaring its syntactic category (s/np/np), and its semantic structure (agent as the first argument, patient as the second one). The bottom portion of Figure 13 shows one of the output parses for the sentence "ʔugʷəč̓əd čəɬ tiʔəʔ hikʷ spaʔc." (English: *We looked for the big bear.*). It shows that the syntactic category of the input is s, and that the semantics involves a definite (DEF) entity (or individual) with predicated properties of being a BEAR and being BIG. It also mentions that there is a SEEK action whose agent is the first-person plural pronominal and whose theme is the entity described (i.e. the BEAR). In predicate representation, the output represents could be expressed: def(y) & bear(y) & big(y) & we(x) & seek(x,y), and in combinator form it would be: SEEK(WE,DEF(BIG(BEAR))). Another parse is returned but not shown here; it deals with the quantificational properties of the determiner.

The current version of the parser assumes that morphology has already been performed, that the input for each word includes a breakdown of the morphemes in their canonical forms. A separate morphological engine has already been developed for the Lushootseed language (Lonsdale, 2001), and future work will involve integrating that morphological processor as a front-end to the parser described in this paper. This will allow for a wholly integrated treatment of sentences in the language.

4 Conclusions and future work

This paper has sought to establish the groundwork for pursuing a categorial grammar account of Lushootseed. It has been necessarily sketchy on the details, especially where complex issues of morphology and syntax are concerned. Still, the basic CG framework seems to be applicable, and exploring further complexities should be possible.

One major aspect of Lushootseed grammar is reduplication. This paper has not addressed the issue of how to codify reduplication in a categorial manner. Though languages with reduplication have been treated in a CG framework, it remains unclear to what extent the work carried out in this area transfers to Salish languages. This is an item for further study.

Other problems not addressed, but where there is more optimism for satisfactory resolution, is in syntactic complexity. Many more word orders than those discussed in this paper exist for Lushootseed reflecting negation, Wackernagel clitic placement, topicalization, and so on. Many of these problems have been addressed for other languages, and further work beyond this basic introduction should reveal techniques and procedures applicable to Lushootseed description.

Of course, though the focus has been on Lushootseed in this paper, the approach should be just as easily implementable for other Salish languages. Language-specific details would need to be worked out, but this could result in insights into both areas: linguistic description of these languages, and the basic CG approach. Describing and leveraging the semantics of items with lexical suffixes, for example, seems like an area very suitable for exploration within CG. Other work might include treating quantification, aspectual types and coercion, and so forth.

Engines such as the one described here for have been used in various applications including speech recognition systems, language learning environments, language generators, corpus interfaces, and text understanding systems. As the evolution of the CG grammar engine continues, a powerful component will be available for processing Lushootseed in such applications.

References

Ajdukiewicz, K. (1935). Die syntaktische Konnexität. *Studia Philosophica*, 1:1–27.

Bar-Hillel, Y. (1953). A quasi-arithmetical notation for syntactic description. *Language*, 25.

Bates, D., Hess, T., and Hilbert., V. (1994). *Lushootseed Dictionary*. University of Washington Press.

Bennett, D. and Beck, D. (1998). Extending the prosodic hierarchy: Evidence from lushootseed narrative. *Proceedings of the 33rd International Conference on Salishan and Neighboring Languages.*

Bozşahin, H. C. and Göçmen, E. (1995). A categorial framework for composition in multiple linguistic domains. *Proceedings of the Fourth International Conference on Cognitive Science of NLP.*

Carpenter, B. (1992). Lexical rules, categorial grammar, and the English predicative. Unpublished manuscript.

Carpenter, B. (1997). *Type-Logical Semantics.* MIT Press.

Carpenter, B. and Penn, G. (2001). The attribute logic engine user's guide: Version 3.2.1. Technical report, SpeechWorks Research and the University of Toronto Department of Computer Science.

Cha, J., Lee, G., and Lee, J.-H. (1999). Modeling and parsing of Korean with K-CCG. *Proceedings of the 18th International Conference on Computer Processing of Oriental Languages (ICCPOL99)*, pages 67–74.

Hess, T., editor (1995). *Lushootseed reader with introductory grammar, Vol. 1.* University of Montana Occasional Papers in Linguistics. Summer Institute of Linguistics.

Hess, T., editor (1998). *Lushootseed reader with introductory grammar, Vol. 2.* University of Montana Occasional Papers in Linguistics. Summer Institute of Linguistics.

Lonsdale, D. (2001). A two-level morphology engine for Lushootseed. *Proceedings of the 36th International Conference on Salishan and Neighbouring Languages.*