Coordination in an Axiomatic Grammar*

David Milward
University of Cambridge
Computer Laboratory
New Museums Site, Pembroke Street, Cambridge, CB2 3QG, England
drm1@cl.cam.ac.uk

Abstract

For some time there has been interest in the idea of parsing as deduction. Here we present a grammatical formalism, 'Axiomatic Grammar', which is based upon a small number of linguistically motivated axioms and deduction rules. Each axiom or rule combines a 'category' with a string of words to form a further category. This contrasts with the usual 'tree-structure' approach to syntactic analysis where constituents are combined with each other to form a further constituent.

We describe a grammar for English which has a good coverage of 'non-constituent' coordination. The grammar has been integrated with a toy semantics, and has been implemented in a left-to-right parser with incremental semantic interpretation. The parser does not suffer from spurious ambiguity.

1 Introduction

Coordination is a particularly troublesome phenomenon to account for in theories of syntax based upon phrase structure rules. Acceptable examples of 'non-constituent' coordination such as:

(1) John gave Mary a book and Peter a paper

(2) Ben likes and Fred admires Mary

have led some to abandon a single level of grammatical description, and others to abandon phrase structure rules.

An example of the former approach is Modifier Structure Grammar (Dahl and McCord, 1983), which was justified as follows:

> ... it appears that a proper and general treatment must recognise coordination as a 'metagrammatical' construction, in the sense that metarules, general system operations, or 'second-pass' operations such as transformations, are needed for its formulation.

Modifier Structure Grammar embeds its rules for coordination into the parsing algorithm (there are close parallels with the SYSCONJ system (Woods, 1973)). In order to parse sentence (2), the state of the parser at the point immediately before 'Fred' is matched to the state immediately before 'Ben'. 'Fred admires' is then parsed, and the resulting state is merged with the state after parsing 'Ben likes'.

The alternative approach to dealing with coordination uses a single level of grammatical description, but uses a weaker notion of constituency than phrase structure grammar. It is presently exemplified by proposals to extend Categorial Grammar with Forward Composition, the Product operator, Subject Type-Raising etc.

Categorial Grammar, just like phrase structure grammar, is based upon the combination of one or more constituents to form a further constituent. In order to deal with coordination, the category $(X\backslash X)/X$[1] is assigned to the conjunction, or, more usually, a phrase structure rule is invoked of the form:

$$X \rightarrow X \text{ conj } X$$

In either case, each conjunct has to be assigned a category. Extensions to Categorial Grammar provide a greater coverage of coordination phenomena by allowing a greater number of strings to form categories. For example, to accept both (2) and (3),

(3) Ben likes Mary and admires Jane

an extended grammar must allow 'Ben likes' to form a category which can combine with 'Mary' to form a sentence, and 'likes Mary' to form a category which can combine with 'Ben' to form a sentence. The consequence of this is that the simple sentence 'Ben likes Mary' can be assigned at least two different syntactic structures: (Ben (likes Mary)) or ((Ben likes) Mary), which both correspond to the same reading (the sentence is spuriously ambiguous according to the grammar).

Axiomatic Grammar avoids the problem of spurious ambiguity by avoiding the need to assign categories to conjuncts. Although the formalism was developed during research into extended Categorial Grammar, the separation of grammatical information into axioms and rules makes its treatment of coordination look similar to that in a metalevel approach such as Modifier Structure Grammar. This

*This research was supported by an SERC research studentship.

[1] Capital letters will be used to denote variables throughout this paper.

similarity is easiest to show if we introduce the central notion of 'category transition' through the idea of state transition.

Consider a left-to-right parse of the sentence 'a man sits' based upon a phrase structure grammar including the rules:

$$s \rightarrow np\ vp \qquad np \rightarrow det\ n$$

Initially we start in a state expecting a sentence (which we can encode as a list ⟨s⟩). After absorbing the determiner, 'a', we can move to a new state which expects a noun followed by a verb-phrase (encoded as a list ⟨n,vp⟩). Following the absorption of 'man', we can move to a state expecting just a verb-phrase, and following the absorption of 'sits' we have a successful parse since there is no more input and nothing more expected. The transitions between the encodings of the states are as follows:

$$\begin{array}{ll} \langle s\rangle + \text{“a”} \rightarrow \langle n,vp\rangle & \text{where a:det} \\ \langle n,vp\rangle + \text{“man”} \rightarrow \langle vp\rangle & \text{where man:n} \\ \langle vp\rangle + \text{“sits”} \rightarrow \langle\rangle & \text{where sits:vp} \end{array}$$

('a:det' *means that the word 'a' is a determiner*)
Instead of deriving these transitions from the phrase structure rules, consider directly supplying axioms of the form:

$$\begin{array}{ll} \langle s\rangle + \text{“W”} \rightarrow \langle n,vp\rangle & \text{where W:det} \\ \langle n,vp\rangle + \text{“W”} \rightarrow \langle vp\rangle & \text{where W:n} \\ \langle vp\rangle + \text{“W”} \rightarrow \langle\rangle & \text{where W:vp} \end{array}$$

If constituent names are replaced by category specifications, generalisations become possible. The three axioms:

$$\begin{array}{ll} \langle s\rangle + \text{“W”} \rightarrow \langle n,s\backslash np\rangle & \text{where W:np/n} \\ \langle n,s\backslash np\rangle + \text{“W”} \rightarrow \langle s\backslash np\rangle & \text{where W:n} \\ \langle s\backslash np\rangle + \text{“W”} \rightarrow \langle\rangle & \text{where W:s}\backslash\text{np} \end{array}$$

('np/n' *is an np requiring a noun on its right, and* 's\np' *is a sentence requiring an np on its left*)
are instantiations of the axioms:

$$\begin{array}{ll} \langle X\rangle \bullet R + \text{“W”} \rightarrow \langle Z,X\backslash Y\rangle \bullet R & \text{where W:Y/Z} \\ \langle X\rangle \bullet R + \text{“W”} \rightarrow R & \text{where W:X} \end{array}$$

('X' *is the head and* 'R', *the tail of the list encoding the state.* '•' *denotes concatenation, so* '⟨n,s\np⟩' *is equivalent to* '⟨n⟩ • ⟨s\np⟩')
The 'encoded states' will roughly correspond to 'principal' categories in Axiomatic Grammar, and the axioms above to the axioms of Prediction and Composition.

The rule for coordination in Axiomatic Grammar is stated in terms of principal category transition. For example, the acceptability of sentence (2) is dependent upon a proof that the two strings "Ben likes" and "Fred admires" both take us from the initial category (corresponding to a parsing state expecting a sentence) to a second category (corresponding to a state expecting a noun-phrase). The rule will be stated formally after a general description of the formalism.

2 The Basics

Axiomatic Grammar is mainly lexically based, with lexical entries containing both subcategorisation and order information. An association of a word with a 'lexical' category is given by an expression of the form:

word: LEX-CAT

Each lexical category is a feature valued structure. The features of interest are 'cat', which gives the base type of the category ('s','np', or 'n'), and 'left' and 'right' which contain lists of 'arguments'. Each argument is itself a lexical category. Categories are complete if the argument lists are empty. As an example, consider the lexical entries for the determiner 'the' and the transitive verb 'likes':

$$\text{the:} \begin{bmatrix} \mathtt{cat} & = np \\ \mathtt{left} & = \langle\rangle \\ \mathtt{right} & = \left\langle \begin{bmatrix} \mathtt{cat} & = n \\ \mathtt{left} & = \langle\rangle \\ \mathtt{right} & = \langle\rangle \end{bmatrix} \right\rangle \end{bmatrix}$$

$$\text{likes:} \begin{bmatrix} \mathtt{cat} & = s \\ \mathtt{left} & = \left\langle \begin{bmatrix} \mathtt{cat} & = np \\ \mathtt{left} & = \langle\rangle \\ \mathtt{right} & = \langle\rangle \end{bmatrix} \right\rangle \\ \mathtt{right} & = \left\langle \begin{bmatrix} \mathtt{cat} & = np \\ \mathtt{left} & = \langle\rangle \\ \mathtt{right} & = \langle\rangle \end{bmatrix} \right\rangle \end{bmatrix}$$

We can read the category for 'likes' as follows: given a complete noun-phrase on the left and a complete noun-phrase on the right, we can form a complete sentence. It is worth comparing this category with the category generally assigned to 'likes' by a Categorial Grammar:

$$(S \setminus NP)/\ NP$$

The categories differ in two respects. Firstly, the Categorial Grammar category not only provides information as to what is on the left, and what is on the right, but also determines the order in which each argument is to be absorbed (in the above, the argument on the right must be absorbed first, followed by the argument on the left). Secondly, whereas the Categorial Grammar category would be regarded as having the syntactic type 'np→(np→s)', the Axiomatic Grammar category is regarded as having the base type 's'. This difference has a bearing on the treatment of modifiers (discussed later).

When a string of words is absorbed it causes a transition between principal categories. A principal category is again a feature structure, the feature of interest being the 'right' feature i.e. the list of arguments[2] required on the right. A parse of a sentence consists of a proof that, starting with a principal category which requires a sentence, we can end

[2] Arguments are again lexical categories.

up with a complete principal category. For example, to prove that 'Ben sits' is a sentence we prove the statement[3]

$$\begin{bmatrix} \mathrm{r} = \langle \begin{bmatrix} \mathrm{c} = s \\ \mathrm{l} = \langle\rangle \\ \mathrm{r} = \langle\rangle \end{bmatrix} \rangle \end{bmatrix} \text{"Ben sits"} \begin{bmatrix} \mathrm{r} = \langle\rangle \end{bmatrix}$$ [4]

Henceforth, the convention is adopted that left or right argument lists which are not specified are empty. This allows us to rewrite the statement above rather more compactly as:

$$\begin{bmatrix} \mathrm{r} = \langle \begin{bmatrix} \mathrm{c} = s \end{bmatrix} \rangle \end{bmatrix} \text{"Ben sits"} \begin{bmatrix} \; \end{bmatrix}$$

A proof of a parse is performed using rules and axioms. An axiom declares that a string of words performs a transition between two principal categories. Axioms are either simple statements, or restricted statements of the form:

$$\mathrm{C}_0 \text{ String } \mathrm{C}_1 \quad \text{where ....}$$

Three axioms will be discussed here[5]. The first, Identity, merely declares that an empty string performs the identity transition i.e.

$$\mathrm{C}_0 \text{ ""} \mathrm{C}_0$$

The other axioms, Prediction and Composition, work on strings consisting of a single word. They have the format:

$$\mathrm{C}_0 \text{ "W" } \mathrm{C}_1 \quad \text{where} \quad \text{W:LEX-CAT}$$

The full definitions, given in Figure 1, should become clearer as we work through an example.

A deduction rule in Axiomatic Grammar declares that a string of words performs a transition between two principal categories provided that certain substrings perform certain transitions i.e. rules have the format[6]:

$$\frac{\mathrm{C}_0 \text{ String}_0 \ \mathrm{C}_1, \ \ldots \ ,\mathrm{C}_n \text{ String}_n \ \mathrm{C}_{n+1}}{\mathrm{C}_a \text{ String } \mathrm{C}_b}$$

(*subscripted strings are substrings of* 'String')
The consequent of a rule (the statement under the line) can be proved by proving all the antecedents (the statements above the line).

[3]Feature names are abbreviated in an obvious manner. Simple statements have the general form:

C_0 String C_1

('C_0' *and* 'C_1' *are principal categories, and* 'String' *is a string of words*)

[4]The corresponding state transition would be:

$\langle s \rangle$ + "Ben sits" $\rightarrow \langle\rangle$

[5]A fourth axiom is used for topicalisation.

[6]This is actually the form of simple rules. As with axioms, rules may be restricted using a 'where' clause.

3 An Example Proof

In order to prove that 'Ben sits' is a sentence, we need to use all the axioms, and two rules, Sequencing and Optional Reduction. The relevant proof tree is given in Figure 2[7].

The Prediction Axiom is restricted in English to the case where a category requires a sentence on the right, and the word encountered has a lexical category of base type noun-phrase. Thus starting with the principal category:

$$\begin{bmatrix} \mathrm{r} = \langle \begin{bmatrix} \mathrm{c} = s \end{bmatrix} \rangle \end{bmatrix}$$

we can absorb the proper-name 'Ben', which has the lexical category, $\begin{bmatrix} \mathrm{c} = np \end{bmatrix}$, to form a principal category, 'c0', which requires first an optional noun-phrase modifier (e.g. a non-restrictive relative clause), and then a sentence which requires a noun-phrase (a verb-phrase) i.e.

$$\begin{bmatrix} \mathrm{r} = \langle \begin{bmatrix} \mathrm{c} = (np) \\ \mathrm{l} = \langle \begin{bmatrix} \mathrm{c} = np \end{bmatrix} \rangle \end{bmatrix}, \begin{bmatrix} \mathrm{c} = s \\ \mathrm{l} = \langle \begin{bmatrix} \mathrm{c} = np \end{bmatrix} \rangle \end{bmatrix} \rangle \end{bmatrix}$$

(*the use of parentheses around the base type of the noun-phrase modifier denotes optionality*[8])
Writing this as a statement in the logic, we have a proof that:

$$\begin{bmatrix} \mathrm{r} = \langle \begin{bmatrix} \mathrm{c} = s \end{bmatrix} \rangle \end{bmatrix} \text{"Ben" c0}$$

The Sequencing Rule[9] is used to combine the effects of the absorption of two strings. The rule declares that if one string defines a transition from Category0 to Category1, and another defines a transition from Category1 to Category2, then the combined string defines a transition from Category0 to Category2 i.e.

$$\frac{\mathrm{C}_0 \text{ String}_0 \ \mathrm{C}_1, \ \mathrm{C}_1 \text{ String}_1 \ \mathrm{C}_2}{\mathrm{C}_0 \text{ String}_0 \bullet \text{String}_1 \ \mathrm{C}_2}$$

(*here '•' denotes concatenation of word strings e.g. "Ben"• "sits" is equivalent to "Ben sits"*)

For this example, we can instantiate the Sequencing Rule as follows:

$$\frac{\begin{bmatrix} \mathrm{r} = \langle \begin{bmatrix} \mathrm{c} = s \end{bmatrix} \rangle \end{bmatrix} \text{"Ben" c0, c0 "sits"} \begin{bmatrix} \; \end{bmatrix}}{\begin{bmatrix} \mathrm{r} = \langle \begin{bmatrix} \mathrm{c} = s \end{bmatrix} \rangle \end{bmatrix} \text{"Ben sits"} \begin{bmatrix} \; \end{bmatrix}}$$

[7]At this stage no restrictions have been imposed upon the ordering of the rules, and more than one proof tree is possible. However, it is relatively trivial to prove the existence of a normal proof strategy which supplies a single proof tree for a given sentence and a possible semantics (Milward, 1990).

[8]Parentheses are used as a shorthand. The lexical categories in argument lists actually include a feature 'opt', which is set to an uninstantiated variable when the argument is optional, to 'true' if the argument is compulsory.

[9]The name 'Sequencing Rule' is due to a loose correspondence between the grammar and the Floyd Hoare Rules for Axiomatic Semantics of programming languages

COMPOSITION

$$\begin{bmatrix} \text{r} = \langle \begin{bmatrix} \text{c} = X \\ \text{l} = L \\ \text{r} = R \end{bmatrix} \rangle \bullet R'' \end{bmatrix} \text{ "W" } \begin{bmatrix} \text{r} = R' \bullet \langle \begin{bmatrix} \text{c} = (X) \\ \text{l} = \langle [\,\text{c} = X\,] \rangle \end{bmatrix} \rangle \bullet R'' \end{bmatrix} \quad \text{where W:} \begin{bmatrix} \text{c} = X \\ \text{l} = L \\ \text{r} = R \bullet R' \end{bmatrix}$$

PREDICTION

$$\begin{bmatrix} \text{r} = \langle [\,\text{c} = X\,] \rangle \bullet R'' \end{bmatrix} \text{ "W" } \begin{bmatrix} \text{r} = R' \bullet \langle \begin{bmatrix} \text{c} = (Y) \\ \text{l} = \langle [\,\text{c} = Y\,] \rangle \end{bmatrix}, \begin{bmatrix} \text{c} = X \\ \text{l} = \langle [\,\text{c} = Y\,] \rangle \end{bmatrix} \rangle \bullet R'' \end{bmatrix}$$

where $\text{W:} \begin{bmatrix} \text{c} = Y \\ \text{r} = R' \end{bmatrix}$ and, in English, $X=s$, $Y=np$.

('•' denotes concatenation of lists. Optional arguments have the value of the 'cat' feature in parentheses. X may be instantiated to the base types 's','np', or 'n'; L,R,R' and R" to lists of categories)

Figure 1: Axioms

$$\dfrac{\dfrac{}{\begin{bmatrix} \text{r} = \langle [\,\text{c} = s\,] \rangle \end{bmatrix} \text{ "Ben" c0}}\,\text{PREDICTION} \qquad \dfrac{\dfrac{\dfrac{}{\text{c1 "sits" c2}}\,\text{COMPOSITION} \quad \dfrac{\dfrac{}{[\;] \text{ "" } [\;]}\,\text{IDENTITY}}{\text{c2 "" } [\;]}\,\text{OPTIONAL REDN}}{\text{c1 "sits" } [\;]}\,\text{SEQUENCING}}{\text{c0 "sits" } [\;]}\,\text{OPTIONAL REDN}}{\begin{bmatrix} \text{r} = \langle [\,\text{c} = s\,] \rangle \end{bmatrix} \text{ "Ben sits" } [\;]}\,\text{SEQUENCING}$$

('c0','c1' and 'c2' are principal categories mentioned in the text)

Figure 2: Proof Tree for 'Ben sits'

We can thus obtain a proof of the whole sentence by proving the antecedents to the rule. The first has already been proved, so we are left to prove:

$$\text{c0 "sits" } [\;]$$

The head of the argument list of c0 is an optional noun-phrase modifier. Optional categories at the head of the argument list of a principal category can be deleted by the use of the Optional Reduction Rule which is as follows:

$$\dfrac{\begin{bmatrix} \text{r} = R'' \end{bmatrix} \text{ String C}}{\begin{bmatrix} \text{r} = \langle \begin{bmatrix} \text{c} = (X) \\ \text{l} = L \\ \text{r} = R \end{bmatrix} \rangle \bullet R'' \end{bmatrix} \text{ String C}}$$

We instantiate the Optional Reduction Rule to:

$$\dfrac{\text{c1 "sits" } [\;]}{\text{c0 "sits" } [\;]}$$

in which 'c1' is c0 without the optional modifier i.e.

$$\begin{bmatrix} \text{r} = \langle \begin{bmatrix} \text{c} = s \\ \text{l} = \langle [\,\text{c} = np\,] \rangle \end{bmatrix} \rangle \end{bmatrix}$$

The proof now consists of proving the antecedent of the Optional Reduction Rule i.e.

$$\begin{bmatrix} \text{r} = \langle \begin{bmatrix} \text{c} = s \\ \text{l} = \langle [\,\text{c} = np\,] \rangle \end{bmatrix} \rangle \end{bmatrix} \text{ "sits" } [\;]$$

This can be proved using first the Composition Axiom, then the Sequencing Rule followed by Optional Reduction, and finally the Identity Axiom.

The Composition Axiom[10] absorbs a word which has the same base category as the head of the argument list of a principal category. Since the word 'sits' has the following category:

$$\begin{bmatrix} \text{c} = s \\ \text{l} = \langle [\,\text{c} = np\,] \rangle \end{bmatrix}$$

the Composition Axiom can be used to absorb 'sits' and get us to the category 'c2':

$$\begin{bmatrix} \text{r} = \langle \begin{bmatrix} \text{c} = (s) \\ \text{l} = \langle [\,\text{c} = s\,] \rangle \end{bmatrix} \rangle \end{bmatrix}$$

[10] The name 'Composition' is due to the similarity with the rule of generalised Forward Composition in a Categorial Grammar.

Using the Sequencing Rule once more, we can prove the whole given a proof of

$$\begin{bmatrix} \mathtt{r} = \langle \begin{bmatrix} \mathtt{c} = (s) \\ \mathtt{l} = \langle [\, \mathtt{c} = s \,] \rangle \end{bmatrix} \rangle \end{bmatrix} \text{“”} \begin{bmatrix} \; \end{bmatrix}$$

which can be proved by first invoking the Optional Reduction Rule. The optional sentential modifier is then deleted, leaving us with a proof of

$$\begin{bmatrix} \; \end{bmatrix} \text{“”} \begin{bmatrix} \; \end{bmatrix}$$

which is true by the Identity Axiom.

4 Rules and Lexical Items

So far we have introduced three axioms which are used by the grammar, and two rules. Before considering further rules it is worth discussing the grammar as it stands.

The effect of the axioms, Prediction and Composition, is to absorb a word and to predict an optional modifier for the base type. For example, in parsing 'the girl' a noun-phrase modifier is predicted after parsing 'the' and a noun-modifier is predicted after parsing 'girl'. Thus, given a treatment of non-restrictive relatives, we could parse something like:

(4) The girl outside, who has been waiting a long time, looks frozen

Moreover, after parsing a noun modifier, another noun modifier is predicted (the base type of a noun modifier is, after all, a noun). Thus we could also parse

(5) The girl outside in the red dress with the large man ....

Although the treatment of noun and noun-phrase modification looks reasonably traditional, the treatment of verbal modification is less so. Since the base type of a verb is a sentence, a modifier for the verb has the same type as a sentential modifier. For example, in:

(6) John hit the ball with a racket

the action of the Composition Axiom is to add an optional sentential modifier onto the end of the subcategorisation list of the verb 'hit', and then to add this list onto the list of expected arguments i.e. after absorbing "hit" the principal category becomes:

$$\begin{bmatrix} \mathtt{r} = \langle [\, \mathtt{c} = np \,], \begin{bmatrix} \mathtt{c} = (s) \\ \mathtt{l} = \langle [\, \mathtt{c} = s \,] \rangle \end{bmatrix} \rangle \end{bmatrix}$$

A successful proof of the sentence is achieved by giving 'with' a lexical entry:

$$\begin{bmatrix} \mathtt{c} = s \\ \mathtt{l} = \langle [\, \mathtt{c} = s \,] \rangle \\ \mathtt{r} = \langle [\, \mathtt{c} = np \,] \rangle \end{bmatrix}$$

Sentences such as 'John decided to sack Mary in secret' are correctly treated as being structurally ambiguous, since 'in secret' may modify the 's' introduced by 'decided' or the 's' introduced by 'sack'.

The grammar which has been described so far imposes a strict notion of word order. This seems particularly inappropriate for relative clauses which can be extraposed from a position following the subject noun-phrase to after the verb-phrase. Consider the sentence:

(7) Children arrived who only spoke English

The present grammar treats this case by allowing heavy noun and noun-phrase modifiers to swap places with categories having a base type 's'. Thus the principal category created after absorbing "Children":

$$\begin{bmatrix} \mathtt{r} = \langle \begin{bmatrix} \mathtt{c} = (np) \\ \mathtt{l} = \langle [\, \mathtt{c} = np \,] \rangle \end{bmatrix}, \begin{bmatrix} \mathtt{c} = s \\ \mathtt{l} = \langle [\, \mathtt{c} = np \,] \rangle \end{bmatrix} \rangle \end{bmatrix}$$

can be transformed into:

$$\begin{bmatrix} \mathtt{r} = \langle \begin{bmatrix} \mathtt{c} = s \\ \mathtt{l} = \langle [\, \mathtt{c} = np \,] \rangle \end{bmatrix}, \begin{bmatrix} \mathtt{c} = (np) \\ \mathtt{l} = \langle [\, \mathtt{c} = np \,] \rangle \end{bmatrix} \rangle \end{bmatrix}$$

The possibility is being considered of replacing lists of arguments by sets of arguments associated with linear precedence constraints (along the lines of work done on bounded discontinuous constituency (Reape, 1989)).

Finally, let us consider the particular restriction which was made to the Prediction Rule for English. The effect of the restriction is that the only acceptable lexical entries with left arguments are either of the form

$$\begin{bmatrix} \mathtt{c} = s \\ \mathtt{l} = \langle [\, \mathtt{c} = np \,] \rangle \\ \mathtt{r} = R \end{bmatrix} \quad \text{or} \quad \begin{bmatrix} \mathtt{c} = X \\ \mathtt{l} = \langle [\, \mathtt{c} = X \,] \rangle \\ \mathtt{r} = R \end{bmatrix}$$

i.e. verbs (which require a noun-phrase subject on their left), or modifiers of the base types.

5 The Coordination Rule

The Coordination Rule is as follows:

$$\frac{C_0 \text{ String}_0 \; C_1, \; C_0 \text{ String}_1 \; C_1}{C_0 \text{ String}_0 \bullet \text{“W”} \bullet \text{String}_1 \; C_1}$$

where W ∈ {and,or,but}

This contrasts with the phrase structure rule:

$$\text{X} \rightarrow \text{X conj X}$$

which can be expressed in deduction rule format as:

$$\frac{\text{String}_0 : \text{X}, \;\; \text{String}_1 : \text{X}}{\text{String}_0 \bullet \text{“W”} \bullet \text{String}_1 : \text{X}}$$

where W ∈ {and,or,but}

Both rules allow nested and iterated conjunction, however, whereas the phrase structure rule enforces that conjuncts are of the same category, the Coordination Rule enforces that each conjunct defines the same transition between principal categories.

We can show the expressive power of the Coordination Rule by considering some examples. The first is an example of 'unbounded' Right-Node Raising:

(8) John admires, but Mary thinks he loves, the new teacher

This can be proved by separately proving that both "John admires" and "Mary thinks he loves" perform a transition between the initial principal category, $\begin{bmatrix} \mathrm{r} = \langle \begin{bmatrix} \mathrm{c} = s \end{bmatrix} \rangle \end{bmatrix}$ and the category:

$$\begin{bmatrix} \mathrm{r} = \langle \begin{bmatrix} \mathrm{c} = np \end{bmatrix}, \begin{bmatrix} \mathrm{c} = (s) \\ 1 = \langle \begin{bmatrix} \mathrm{c} = s \end{bmatrix} \rangle \end{bmatrix} \rangle \end{bmatrix}$$

The proof is completed by proving that "the teacher" defines a transition between this category and the complete principal category, [ ].

The second example involves sharing on both the right and the left:

(9) He lent John a book, and Mary a paper, about subjacency

This example, which has been used to argue for the addition of meta-rules to Categorial Grammar (Morrill, 1987), is of interest when the required reading is where the noun modifier 'about subjacency' applies to both the book and the paper. To prove the sentence we first prove that "He lent" performs a transition between the initial principal category and the category:

$$\begin{bmatrix} \mathrm{r} = \langle \begin{bmatrix} \mathrm{c} = np \end{bmatrix}, \begin{bmatrix} \mathrm{c} = np \end{bmatrix}, \begin{bmatrix} \mathrm{c} = (s) \\ 1 = \langle \begin{bmatrix} \mathrm{c} = s \end{bmatrix} \rangle \end{bmatrix} \rangle \end{bmatrix}$$

We then prove separately that "John a book", and "Mary a paper" perform a transition between this category and the category:

$$\begin{bmatrix} \mathrm{r} = \langle \begin{bmatrix} \mathrm{c} = (n) \\ 1 = \langle \begin{bmatrix} \mathrm{c} = n \end{bmatrix} \rangle \end{bmatrix}, \begin{bmatrix} \mathrm{c} = (s) \\ 1 = \langle \begin{bmatrix} \mathrm{c} = s \end{bmatrix} \rangle \end{bmatrix} \rangle \end{bmatrix}$$

Finally we prove that the string "about subjacency" takes us from this category to the complete category, [ ].

The basic grammar does accept some sentences which are generally regarded as unacceptable, and extra features are needed to constrain the rules. The situation with the basic grammar is not, however, as bad as with many extended Categorial Grammars. The Coordination Rule enforces a parallelism between conjuncts in a similar manner to the parallelism enforced by the phrase structure rule mentioned above. This can be contrasted with assigning conjunctions the polymorphic category (X\X)/X, which allows sentences like:

(10) John likes Mary and, or Peter likes Joan and Anne

Further parallelism is enforced by the particular treatment of wh-movement[11] which, for example, predicts the acceptability of (11) but not of (12):

(11) The book arrived which John had shown Mary and given to Peter

(12) *The book arrived which John had shown Mary and to Peter

6 Conclusion

This paper has introduced Axiomatic Grammar, and has given some justification for particular axioms and rules chosen for English. The formalism itself has been left very much underspecified, and further research is required both into its applicability to other languages, and into its formal properties.

A larger grammar for English has been implemented, including a treatment of wh-movement and verbal ellipsis (gapping). The parser works word-by-word from left-to-right, and was designed so that incorporation of the coordination rule does not slow down parsing in general.

Axiomatic Grammar fits in naturally with an incremental approach to semantic interpretation, or with semantics based upon state change. The present grammar is integrated with a toy semantics based upon the incremental (but non-monotonic) accumulation of constraints.

References

Dahl, Veronica and McCord, Michael C. (1983). Treating Coordination in Logic Grammars. *Computational Linguistics*, 9(2):69–91.

Milward, David R. (1990). Ph.D. Thesis. Forthcoming.

Morrill, Glyn (1987). Meta-Categorial Grammar. In Haddock, Nicholas, Klein, Ewan, and Morrill, Glyn (eds.), *Categorial Grammar, Unification Grammar and Parsing*, Centre for Cognitive Science, University of Edinburgh.

Reape, Mike (1989). A Logical Treatment of Semi-Free Word Order and Bounded Discontinuous Constituency. In *European ACL*, pages 103–110.

Woods, W. (1973). An Experimental Parsing System for Transition Network Grammars. In Rustin, R. (ed.), *Natural Language Processing*, Algorithmics Press.

[11] The rules for wh-movement involve the use of a feature on principal categories which 'stacks' extracted elements, and the use of further features to control extraction sites.