Canad. Math. Bull. Vol. **32** (3), 1989

GRAMMAR AS MATHEMATICS

BY

J. LAMBEK

Dedicated to the memory of William Lloyd Garrison Williams.

ABSTRACT. While there are a small number of reasonably deep theorems in mathematical linguistics, I wish to argue that grammar is mathematics at a very basic level, albeit "trivial" mathematics. Linguistic activities such as the production and recognition of sentences are quite analogous to the mathematical activities of proving theorems or making calculations, while learning a language involves something akin to the discovery or invention of postulates.

Preamble. I feel greatly honoured to have been chosen to give the Jeffery-Williams lecture, particularly as I was a friend of both Ralph Jeffery, who used to look over my shoulder at the summer research institute in Kingston, and Lloyd Williams, who had encouraged me to become a professional mathematician in the first place and often invited me to his home when I was but a humble undergraduate. I take this opportunity to talk on a subject which not everyone believes to be a part of mathematics (e.g. the NSERC Grant Selection Committee), but which Lloyd would have approved of, as he himself had gone to Oxford as a Rhode scholar after having majored in Classics.

0. Intoduction.

Pythagoras, who coined the word "mathematics", was supposed to have said "everything is number". Most mathematicians would concede that at least all things in mathematics are numbers, although they might mumble under their breath: "or sets of numbers or sets of sets of numbers ...". Anyway, according to Pythagoras, "mathematics" (the word is related to the English word "mind") covered four subjects: arithmetic, geometry, astronomy and music. While we might have our doubts today about music, we could be persuaded that astronomy is "applied" mathematics. Moreover, ever since Descartes, we accept the reduction of geometry to "real" numbers, and these, according to Eudoxus and Dedekind, may be viewed as sets of rational numbers, which are the numbers Pythagoras had in mind.

Conspicuously absent from his classification is grammar. But wait, the four mathematical subjects of Pythagoras re-appear in the medieval university curriculum as the four liberal arts, the so-called "quadrivium". They are preceded by the "trivium" consisting of the three subjects: grammar, logic and rhetoric, hence labelled "trivial". No one would claim that rhetoric is mathematics, but logic has by now been

Received by the editors June 13, 1988.

AMS (MOS) Subject Classification Scheme (1970) G8A30.

accepted as respectable mathematics almost everywhere and grammar is beginning to gain recognition at certain European universities as "mathematical linguistics" and, even in Montreal, I have been teaching an undergraduate course "computability and mathematical linguistics" for the last twenty years or so.

Actually, grammar deals with finite strings of symbols taken from a finite alphabet. According to Cantor, these strings are in one-to-one correspondence with the natural numbers, an idea which has been exploited by Gödel in the proof of his famous incompleteness theorem. However, my claim that grammar is mathematics is not based on Gödel's arithmetization, but rather on the observation that proof and computation are as fundamental in language as in mathematics.

If my position is controversial, at least my interest in language has been shared by many mathematicians, including Lloyd Williams, and I know of three who have made contributions to linguistics in the past: Eratosthenes, Wallis and Grassmann.

The kind of grammar I favour nowadays has been called by various names: semi-Thue systems, rewriting systems, productions grammars. In mathematics, semi-Thue systems were studied in connection with the word problem for semigroups (see e.g. Kleene 1952).

For our purposes, a *semi-Thue system* is a finitely generated free monoid together with a finitely generated pre-order on it. Thus, we are given a finite set $\mathcal{V}$ of symbols, usually called the *vocabulary* or *alphabet*, and a finite subset $\mathcal{P}$ of $\mathcal{V}^* \times \mathcal{V}^*$, called the set of *productions*. Here $\mathcal{V}^*$ is the free monoid generated by $\mathcal{V}$; it consists of finite strings of elements of $\mathcal{V}$, including the empty string 1, multiplication being just concatenation. From $\mathcal{P}$ we generate a pre-order relation on $\mathcal{V}^*$ with the help of the following axioms and rules of inference:

$$\Gamma \to \Delta \text{ when } (\Gamma, \Delta) \in \mathcal{P} \quad \text{(productions)},$$

$$\Gamma \to \Gamma \quad \text{(reflexive law)},$$

$$\frac{\Gamma \to \Delta \quad \Delta \to \Lambda}{\Gamma \to \Lambda} \text{(transitive rule)},$$

$$\frac{\Gamma \to \Delta \quad \Gamma' \to \Delta'}{\Gamma\Gamma' \to \Delta\Delta'} \text{(substitution rule)}.$$

Thus, we may write $\Gamma \leqq \Delta$ to mean that $\Gamma \to \Delta$ is provable in the above deductive system. We have used capital Greek letters to denote strings of symbols.

The substitution rule is easily seen to be equivalent to the following:

$$\frac{\Gamma \to \Delta}{\Phi\Gamma\Psi \to \Phi\Delta\Psi}.$$

As the free monoid also contains the empty string, special cases of this rule are:

$$\frac{\Gamma \to \Delta}{\Phi\Gamma \to \Phi\Delta}, \qquad \frac{\Gamma \to \Delta}{\Gamma\Psi \to \Delta\Psi}.$$

A *production grammar* $\mathcal{G} = (\mathcal{V}, \mathcal{P}, \mathcal{V}_i, \mathcal{V}_t)$ is a semi-Thue system in which two subsets of $\mathcal{V}$ have been specified: the *initial* vocabulary $\mathcal{V}_i$ and the *terminal* vocabulary $\mathcal{V}_t$. Note the symmetry of our definition: if $\mathcal{P}$ is replaced by its converse, hence all arrows are reversed, and if $\mathcal{V}_i$ and $\mathcal{V}_t$ are interchanged, we get another production grammar, called the *dual* of $\mathcal{G}$.

For example, if we wish to generate English sentences, we might take $\mathcal{V}_t$ to be the set of all English words, say those listed in the Oxford dictionary, together with all their inflected forms, and $\mathcal{V}_i$ as the set $\{S,Q,C, \ldots\}$, where S = statement, Q = question, C = command et cetera are the different types of sentences we wish to generate. Assuming $\mathcal{V}$ and $\mathcal{P}$ to have been chosen judiciously, if Γ is a string of English words, we expect to be able to prove $S \longrightarrow \Gamma$ then and only then when Γ is a grammatical declarative sentence.

The dual of the grammar just considered will serve to analyze certain strings of English words as different types of sentences. For example, in this dual grammar we might expect to prove

$$\textit{does he come} \longrightarrow \mathrm{Q}.$$

If we are interested in translating from French to English, we might construct a production grammar in which $\mathcal{V}_i$ is the French vocabularly and $\mathcal{V}_t$ the English vocabulary. In such a grammar we would expect to be able to prove

$$\textit{il va} \longrightarrow \textit{he goes}.$$

The translation grammar will of course be much more diffcult to construct than the separate grammars of the two languages.

In this brief exposition we shall discuss in detail three examples of languages and their grammars. The first is a formal language which has been specially constructed to illustrate a technical point, namely that the intersection of two contextfree languages need not be contextfree. The second is a fragment of English dealing with consanguineous kinship terminology, along lines which anthropologists have adopted to look at more exotic kinship systems. The third is a fragment of English which exhibits the structure of basic sentences such as *he likes her*. Our grammar should account not only for the syntax, but also for the morphology. In particular, it should not be able to generate the ungrammatical **he like she*. (Linguists usually put a star on incorrect forms.)

Let me emphasize that my main interest lies in the grammars of natural languages and the "trivial" theorems they contain, even though there exist many nontrivial mathematical theorems, or should I say "metatheorems", about formal languages. For the latter, the reader is referred to the beautiful exposition by Cohn (1975), who also discusses our first example below.

1. **A formal language which is not contextfree.** In examples of formal languages, one usually assumes that $\mathcal{V}_i = \{S\}$ consists of a single symbol S. The sentences of such a language are then all strings $\Gamma \in \mathcal{V}_t^*$ for which $S \longrightarrow \Gamma$ is provable in the deductive system. For example, consider the terminal vocabulary $\mathcal{V}_t = \{a, b, c\}$, from which we want to generate the language

$$\mathcal{L} = \{\mathrm{a}^k\mathrm{b}^k\mathrm{c}^k \mid k \geqq 1\}.$$

We take $\mathcal{V}_i = \{\mathrm{S}\}$ and $\mathcal{V} = \{\mathrm{a,b,c,S,B}\}$, with one auxiliary symbol B, and the following productions:

$$\begin{aligned} \mathrm{S} &\longrightarrow \mathrm{abc}, \\ \mathrm{S} &\longrightarrow \mathrm{aBSc}, \\ \mathrm{Ba} &\longrightarrow \mathrm{aB}, \\ \mathrm{Bb} &\longrightarrow \mathrm{bb}. \end{aligned}$$

Indeed, here is a proof that S ⟶ aabbcc:

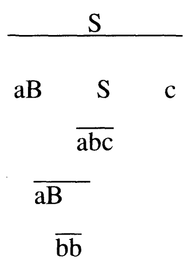
$$\cfrac{\cfrac{\mathrm{S}\longrightarrow\mathrm{aBSc}\qquad\cfrac{\cfrac{\cfrac{\mathrm{S}\longrightarrow\mathrm{abc}}{\mathrm{BS}\longrightarrow\mathrm{Babc}}\qquad\cfrac{\mathrm{Ba}\longrightarrow\mathrm{aB}}{\mathrm{Babc}\longrightarrow\mathrm{aBbc}}}{\mathrm{BS}\longrightarrow\mathrm{aBbc}}}{\mathrm{aBSc}\longrightarrow\mathrm{aaBbcc}}}{\mathrm{S}\longrightarrow\mathrm{aaBbcc}}\qquad\cfrac{\mathrm{Bb}\longrightarrow\mathrm{bb}}{\mathrm{aaBbcc}\longrightarrow\mathrm{aabbcc}}}{\mathrm{S}\longrightarrow\mathrm{aabbcc}}$$

Such proofs in tree-form are a bit cumbersome and, in practice, they are replaced by so-called *derivations*, as follows:

$$\mathrm{S}\longrightarrow\mathrm{aBSc}\longrightarrow\mathrm{aBabcc}\longrightarrow\mathrm{aaBbcc}\longrightarrow\mathrm{aabbcc}.$$

Here, the first step is the production S ⟶ aBSc, the second step uses the production S ⟶ abc in the context aB–c, the third step uses the production Ba ⟶ aB in the context a-bcc ad the fourth step uses the production Bb ⟶ bb in the context aa-cc.

Alternatively, one may depict such a proof by a diagram:



Linguists are fond of so-called *contextfree* productions, namely productions of the form $A \longrightarrow \Gamma$ where the left side consists of a single symbol. (Sometimes one also insists that Γ is not the empty string.) Of the productions listed above, only the first two are contextfree. Still, one might hope that another set of contextfree productions

)

will generate the same language. That this is not so follows from a famous result by Bar-Hillel, Perlis and Shamir, which we shall not prove here, as it is easily found in the literature (see e.g. Cohn 1975).

LEMMA (Bar-Hillel et al.) A necessary condition for an infinite language to have a contextfree grammar is that there should exist strings $\Gamma, \Delta, \Lambda, \Phi$ and Ψ, where Φ and Ψ are not both empty, such that $\Gamma\Phi^n\Delta\Psi^n\Lambda$ is a sentence for each $n \geqq 1$.

To illustrate this condition, consider the fragment of English built from the terminal vocabulary $\{I, know, and\}$ and the productions

$$\mathrm{S} \longrightarrow \textit{I know}, \qquad \mathrm{S} \longrightarrow \mathrm{S}\ \textit{and}\ \mathrm{S}.$$

Now take $\Gamma = \textit{I know}$ and $\Phi = \Delta = \Psi = \Lambda = \textit{and I know}$. Let us apply the lemma to the language $\mathcal{L}$ under consideration in this Section and assume that it is contextfree. The cases $n = 1$ and $n = 2$ of the lemma require that there exist strings $\Gamma, \Phi, \Delta, \Psi$ and Λ in $\{a,b,c\}^*$, with Φ and Ψ not both empty, and integers $p, q \geqq 1$ such that

$$\Gamma\Phi\Delta\Psi\Lambda = \mathrm{a}^p\mathrm{b}^p\mathrm{c}^p,$$

$$\Gamma\Phi\Phi\Delta\Psi\Psi\Lambda = \mathrm{a}^q\mathrm{b}^q\mathrm{c}^q.$$

A straight-forward but tedious calculation shows that this leads to a contradiction.

It follows that the language $\mathcal{L}$ is not contextfree. On the other hand, it is easily seen to be the intersection of two contextfree languages:

$$\mathcal{L}_1 = \{\mathrm{a}^m\mathrm{b}^m\mathrm{c}^n | m, n \geqq 1\},$$
$$\mathcal{L}_2 = \{\mathrm{a}^m\mathrm{b}^n\mathrm{c}^n | m, n \geqq 1\}.$$

For example, $\mathcal{L}_1$ is generated by the produtions

$$\mathrm{U}_1 \longrightarrow \mathrm{ab}, \mathrm{U}_1 \longrightarrow \mathrm{aU}_1\mathrm{b},$$
$$\mathrm{V}_1 \longrightarrow \mathrm{c}, \mathrm{V}_1 \longrightarrow \mathrm{V}_1\mathrm{c},$$
$$\mathrm{S}_1 \longrightarrow \mathrm{U}_1\mathrm{V}_1,$$

where S_1 is the initial symbol.

2. **English consanguineous kinship terminology.** Kinship terminologies have been studied by anthropologists, who have been concentrating largely on languages spoken by isolated societies (see e.g. Buchler and Selby 1968). The usefulness of production grammars in this connection was first pointed out by Lounsbury (1965). The original emphasis was on reduction and equivalence rules, which may be illustrated in English by noting that a second cousin is viewed simply as a cousin and that the brother's wife is denoted by the same term as the husband's sister. These rules of primary interest to anthropologists were integrated with structural and morphological rules in

a series of articles written in collaboration with Michael Lambek (1981), who initiated the project, and Mira Bhargava (1983, 1985). Some fruitful ideas were contributed by George Bergman.

We shall here concentrate on the kinship terminology in the idiolect of Goodenough (1965); but, for brevity, we shall ignore any relations through marriage. The present treatment will differ slightly from that in my 1986 paper, by placing more emphasis on contextfree production, at the cost of increasing the number of productions.

We shall adopt the vocabulary:

$$\begin{aligned}\mathcal{V} &= \mathcal{V}_i \smile \mathcal{V}_t \smile \{\mathrm{M},\mathrm{F},\mathrm{P},\mathrm{C},\mathrm{S},\mathrm{U},\mathrm{V},\mathrm{W},[,],\}\\ \mathcal{V}_i &= \{\mathrm{R}\},\\ \mathcal{V}_t &= \{\textit{father}, \textit{mother}, \textit{son}, \textit{daughter}, \textit{brother}, \textit{sister}, \textit{uncle}, \textit{aunt}, \textit{nephew}, \textit{niece},\\ &\qquad \textit{cousin}, \textit{grand}, \textit{great}, \#\}\end{aligned}$$

The reader may think of some of the auxiliary symbols as abbreviations:

$$\begin{aligned}\mathrm{R} &= \text{relation},\\ \mathrm{M} &= \text{male},\\ \mathrm{F} &= \text{female},\\ \mathrm{P} &= \text{parent (of)},\\ \mathrm{C} &= \text{child (of)},\\ \mathrm{S} &= \text{sibling (of)}.\end{aligned}$$

Also we shall use U for descendant, V for ancestor and W for any relation which involves S. The terminal vocabulary includes the prefix *grand* and the symbol #, which is supposed to denote a space between words.

Before we postulate the productions, we take a look at the data which they should account for. The following table summarizes the consanguineous data discussed by Goodenough in 1965 for a certain dialect of American English, with one minor difference: we have spelled *grandnephew* in one word.

TABLE I

MSP^{n+1}	$\longrightarrow$	$(\textit{great}\#)^n\textit{uncle}$	FSP^{n+1}	$\longrightarrow$	$(\textit{great}\#)^n\textit{aunt}$
MP^{n+2}	$\longrightarrow$	$(\textit{great}\#)^n\textit{grandfather}$	FP^{n+2}	$\longrightarrow$	$(\textit{great}\#)^n\textit{grandmother}$
MP	$\longrightarrow$	*father*	FP	$\longrightarrow$	*mother*
MS	$\longrightarrow$	*brother*	FS	$\longrightarrow$	*sister*
$\mathrm{MC}^{m+1}\mathrm{SP}^{n+1}$	$\longrightarrow$	*cousin*	$\mathrm{FC}^{m+1}\mathrm{SP}^{n+1}$	$\longrightarrow$	*cousin*
MC	$\longrightarrow$	*son*	FC	$\longrightarrow$	*daughter*
MCS	$\longrightarrow$	*nephew*	FCS	$\longrightarrow$	*niece*
MC^{m+2}	$\longrightarrow$	$(\textit{great}\#)^m\textit{grandson}$	FC^{m+2}	$\longrightarrow$	$(\textit{great}\#)^m\textit{granddaughter}$
$\mathrm{MC}^{m+2}\mathrm{S}$	$\longrightarrow$	$(\textit{great}\#)^m\textit{grandnephew}$	$\mathrm{FC}^{m+2}\mathrm{S}$	$\longrightarrow$	$(\textit{great}\#)^m\textit{grandniece}$

The entries in Table I are self-explanatory. For example, MSP^2 is a kinship description to be read as 'male sibling of parent of parent" and the table tells us that it is rendered in English as *great # uncle*.

Concerning the kinship description of *cousin*, I have avoided the more precise rendering

$$GC^{m+1}SP^{n+1} \longrightarrow \textit{i-th cousin j times removed,}$$

where $G = \mathrm{M}$ or F and

$$i = \min(m, n) + 1, \quad j = |m - n|,$$

as I find it difficult to calculate these two primitive recursive functions in my head.

We wish to postulate productions which will account for the data in Table I. For example, the productions should predict that

$$MSP^2 \longrightarrow \textit{great\#uncle} \text{ in the context } \# - \#,$$

assigning a kinship term to a kinship description. They should also predict

$$R \longrightarrow MSP^2 \text{ in the context } \# - \#,$$

to account for the fact that MSP^2 is a kinship description in the first place and not, for example, *MPSP. Furthermore, we hope that they will explain the asymmetry between *grandnephew* and *great # uncle* and also account for the absence of **grandcousin*. Here then are our productions:

(1) *Structure rules.*
$U \longrightarrow C$, $U \longrightarrow CU$;
$V \longrightarrow P$, $V \longrightarrow VP$;
$W \longrightarrow S$, $W \longrightarrow CW$, $W \longrightarrow WP$;
$R \longrightarrow GU, GV, GW$ in the context $\# - \#$, where $G =$M or F.

(2) *Reduction rules.*
$C^2SP \longrightarrow CSP$,
$CSP^2 \longrightarrow CSP$.

(3) *Word assignments,* before # and after # or [*grand*].

MSP	$\longrightarrow$ *uncle*	FSP	$\longrightarrow$ *aunt*,
MP	$\longrightarrow$ *father*,	FP	$\longrightarrow$ *mother*,
MS	$\longrightarrow$ *brother*,	FS	$\longrightarrow$ *sister*,
MCSP	$\longrightarrow$ *cousin*,	FCSP	$\longrightarrow$ *cousin*,
MC	$\longrightarrow$ *son*,	FC	$\longrightarrow$ *daughter*,
MCS	$\longrightarrow$ *nephew*,	FCS	$\longrightarrow$ *niece*.

(4) *Prefix assignments*, where $G = \mathrm{M}$ or F.

$$G\mathrm{P}^2 \longrightarrow [\mathit{grand}]G\mathrm{P};$$
$$G\mathrm{C}^2 \longrightarrow [\mathit{grand}]G\mathrm{C};$$
$$[\mathit{grand}] \longrightarrow \begin{cases} \mathit{great}\ \#\ \text{before } \mathit{grand} \text{ or } \mathit{great}, \\ 1 \text{ before cousin}, \\ \text{grand before other English words;} \end{cases}$$
$$G\mathrm{SP}^2 \longrightarrow \text{great}\ \#\ \mathrm{GSP}.$$

We shall now discuss these productions.

(1) The structure rules account for:

$$\mathrm{U} \longrightarrow \mathrm{C}^{m+1}, \quad \mathrm{V} \longrightarrow \mathrm{P}^{n+1}, \quad \mathrm{W} \longrightarrow \mathrm{C}^m\mathrm{SP}^n;$$
$$\mathrm{R} \longrightarrow G\mathrm{C}^{m+1}\mathrm{SP}^{n+1}, G\mathrm{C}^{m+1}(\mathrm{S}), G(\mathrm{S})\mathrm{P}^{n+1}, G\mathrm{S} \text{ in the context } \# - \#;$$

where $m, n, \geqq 0$ and $G = \mathrm{M}$ or F.

(2) The reduction rules account for

$$\mathrm{C}^{m+1}\mathrm{SP}^{n+1} \longrightarrow \mathrm{CSP}.$$

Rules of this kind had been first emphasized by Lounsbury (1965); they are more prominent in some exotic languages than in English.

(3) The word assignments allow us e.g. to rewrite MSP as *uncle* in the context # – #, but not, for example, MSP^2 as *uncle*P.

(4) Structural linguists might consider [*grand*] as a morpheme with three allomorphs: *great#*, *grand* and 1, the empty string.

We shall present some sample calculations, assuming everywhere the context # – #.

$$\begin{aligned} \mathrm{MC}^4 &\longrightarrow [\mathit{grand}]\mathrm{MC}^3 \longrightarrow [\mathit{grand}]^2\mathrm{MC}^2 \longrightarrow [\mathit{grand}]^3\mathrm{MC} \\ &\longrightarrow [\mathit{grand}]^3 \mathit{son} \longrightarrow [\mathit{grand}]^2 \mathit{grandson} \\ &\longrightarrow [\mathit{grand}]\ \mathit{great\#}\ \mathit{grandson} \longrightarrow\ \mathit{great\#}\ \mathit{great\#}\ \mathit{grandson}. \\ \mathrm{MC}^3\mathrm{S} &\longrightarrow [\mathit{grand}]\mathrm{MC}^2\mathrm{S} \longrightarrow [\mathit{grand}]^2\mathrm{MCS} \\ &\longrightarrow [\mathit{grand}]^2\ \mathit{nephew} \longrightarrow [\mathit{grand}]\ \mathit{grandnephew} \\ &\longrightarrow \mathit{great\#}\ \mathit{grandnephew}. \\ \mathrm{MC}^2\mathrm{SP}^2 &\longrightarrow [\mathit{grand}]\ \mathrm{MCSP}^2 \longrightarrow [\mathit{grand}]^2\ \mathrm{MCSP} \\ &\longrightarrow [\mathit{grand}]^2\ \mathit{cousin} \longrightarrow [\mathit{grand}]\ \mathit{cousin} \longrightarrow \mathit{cousin}; \end{aligned}$$

or more quickly:

$$MC^2SP^2 \longrightarrow MCSP^2 \longrightarrow MCSP \longrightarrow \textit{cousin}.$$

$$MSP^3 \longrightarrow \textit{great}\#MSP^2 \longrightarrow \textit{great}\#\textit{great}\#MSP \longrightarrow \textit{great}\#\textit{great}\#\textit{uncle}.$$

Note that *grandnephew* is obtained from the same rule which already accounts for *grandson*, but that the kinship description of *great# uncle* cannot be evaluated by the rule which yields *grandfather*; it requires a new rule.

Webster's New Ideal Dictionary lists *great-nephew* as an alternative to *grandnephew* and, in place of Goodenough's *great#uncle*, it offers a choice between *great-uncle* and *granduncle*. Webster's idiolect could be accounted for by altering the last two rules in (4) as follows:

$$[\textit{grand}] \longrightarrow \textit{grand} \text{ or } \textit{great-} \text{ before other English words;}$$

$$GSP^2 \longrightarrow [\textit{grand}]\, GSP.$$

Why are there no grandcousins? We could try to introduce them by deleting the rule

$$[\textit{grand}] \longrightarrow 1 \qquad \text{before } \textit{cousin}$$

in (4) above. We could then compute

$$MC^2SP \longrightarrow [\textit{grand}]MCSP \longrightarrow [\textit{grand}]\textit{cousin} \longrightarrow \textit{grandcousin}.$$

However, there is no way of converting $MCSP^2$ to *grandcousin*, and this would lead to the undesirable situation where A could be the grandcousin of B whereas B is not the grandcousin of A.

We should stress again that there is no unique set of postulated productions to account for the data in Table 1. In fact the productions offered here differ from those in my 1986 paper, for which Bill Anglin (1986) had given a soundness proof.

3. **Basic sentence structure in English**. Our third example deals with a small fragment of English, which we hope contains all basic declarative sentences subject to certain simplifying restrictions: noun phrases are taken to be personal pronouns, adverbs and adverbial phrases are avoided and only the simplest verb phrases are admitted, namely those built from transitive or intransitive verbs. Some such restrictions are necessary to keep this section within reasonable bounds, yet I hope that the sample is large enough to make a convincing case for our way of presenting English grammar.

The method advocated here is of course greatly influenced by the pioneering work of Chomsky. It differs from the latter by sticking to pure production grammars without additional machinery and by placing syntax and morphology on the same footing from the start, as I had advocated for French (1975), Latin (1979) and more recently for English (1987). I expect to expand the fragment of English treated here in later publications.

Our terminal vocabulary will be a subset of the set of all English words, including inflected forms. The initial vocabulary will consist of a single symbol S. In addition, we shall make use of an auxiliary vocabulary borrowed from traditional grammar, which will be introduced gradually. To start with, we make the following abbreviations:

Subj	=	subject,
Pred	=	predicate,
NP	=	noun phrase,
P_k	=	k-th person ($k = 1, 2, 3$),
Tens	=	tense,
Inf	=	infinitive,
T_i	=	i-th simple tense ($i = 1, 2$),
Neg	=	negation,
Asp	=	aspect,
Pass	=	passive of,
VP	=	nuclear verb phrase,
Perf	=	perfect participle of,
Part	=	present participle of,
Obj	=	object,
Acc	=	accusative of.

We begin with a list of contextfree productions, which determine the structure of basic declarative sentences.

(1.1) $\mathrm{S} \longrightarrow \mathrm{Subj\ Pred}$;
(1.2) $\mathrm{Subj} \longrightarrow \mathrm{NP}_k\ \mathrm{P}_k\ (k = 1, 2, 3)$;
(1.3) $\mathrm{NP}_1 \longrightarrow I$;
$\mathrm{NP}_2 \longrightarrow$ *we, you, they, ...*;
$\mathrm{NP}_3 \longrightarrow$ *he, she, it, one, ...*;
(1.4) $\mathrm{Pred} \longrightarrow \mathrm{Tens\ Inf}$;
(1.5) $\mathrm{Tens} \longrightarrow \mathrm{T}_i\ (\mathrm{Neg})\ (V)\ (i = 1, 2;\ V$ any modal verb);
(1.6) $\mathrm{Inf} \longrightarrow (\mathrm{Asp})\ (\mathrm{Pass})\ \mathrm{VP}$;
(1.7) $(\mathrm{Asp}) \longrightarrow$ (*have* Perf) (*be* Part):

$$(1.8) \qquad \mathrm{VP} \longrightarrow \begin{cases} V \text{ if } V \text{ is an intransitive verb} \\ \text{or a transitive verb with deletable object;} \\ V\ \mathrm{Obj} \text{ if } V \text{ is a transitive verb;} \end{cases}$$

(1.9) $\mathrm{Obj} \longrightarrow \mathrm{Acc\ NP}_k$.

We shall make some comments on these productions.

(1.1) A statement consists of a subject followed by a predicate.

(1.2) The subject consists of a noun phrase followed by a matching person marker: P_1, P_2, P_3 denote the first, second and third person respectively. These markers will act on the verb later. We have taken advantage of the fact that in modern English the three persons of the plural affect the verb in the same way as the second person of the singular. Other European languages still require six persons. In some languages, e.g. Latin, the subject may consist of a person marker alone.

(1.3) There is only one noun phrase which may precede P_1. P_2 may also be preceded by noun phrases built from plurals and P_3 by noun phrases built from count nouns or mass nouns, all with appropriate determiners. All these and some others will not be considered here.

(1.4) The predicate consists of a tense, rather widely conceived, followed by an infinitival verb phrase.

(1.5) The tense must contain T_1 = present or T_2 = past. It may contain a modal verb: *shall, will, can, may* or *must*, two of which traditionally also express the future tense. It has been found convenient to include an optional negation, as its appropriate place is just after T_i. We might just mention that two verbs, *need* and *dare*, become modal verbs when preceded by Neg.

(1.6) The infinitival verb phrase consists of an optional aspect, an optional passive transformation and a nuclear verb phrase.

(1.7) The aspect, if present at all, may be *have* Perf, *be* Part or *have*Perf*be*Part. Here Perf and Part are what I wish to call "inflectors", they act on the verb in a way to be specified later.

(1.8) The nuclear verb phrase consists of an intransitive verb, e.g. *come, go, work, sleep*, ..., or a transitive verb with deletable object, e.g. *eat, call, kill*, ...; it may also consist of any transitive verb followed by an object. The object cannot be deleted after e.g. *like, receive, resemble*, ...

(1.9) The object consists of the accusative inflector followed by a noun phrase. We have retained the subscript k, although it serves no purpose here.

With the help of the productions listed so far we can, for example, prove the following:

$$S \longrightarrow \textit{he}\ P_3T_1\ \textit{call}\ \mathrm{Acc}\ \textit{she},$$

$$S \longrightarrow \textit{he}\ P_3T_1\ \mathrm{Neg}\ \textit{go},$$

$$S \longrightarrow \textit{he}\ P_3T_1\ \mathrm{Neg}\ \textit{will}\ \mathrm{Pass}\ \textit{call}\ \mathrm{Obj},$$

$$S \longrightarrow \textit{he}\ P_3T_1\ \textit{have}\ \mathrm{Perf}\ \textit{be}\ \mathrm{Part}\ \textit{come}.$$

The expressions on the right of the arrows are not yet English sentences; they indicate the structure of certain declarative sentences, which may be computed with the help of further productions, however not contextfree ones.

Before stating these productions, we enlarge our auxiliary vocabulary by including the inflector

$$C_{ik} = \text{conjugation for } i\text{-th tense and } k\text{-th person}$$

and the morphemes

$$+not,\ +s,\ +ed,\ +ing.$$

We also assume that the dictionary lists the past tense V^p and the perfect participle V^q of any so-called irregular verb V. For example, $go^{\mathrm{p}} = went$, $go^{\mathrm{q}} = gone$.

We are now ready to adopt the following new productions.

(2.1)
$$\mathrm{P}_k\mathrm{T}_i \longrightarrow \mathrm{C}_{ik};$$
$$\mathrm{C}_{11}V \longrightarrow \begin{cases} am & \text{if } V = be, \\ V & \text{otherwise;} \end{cases}$$
$$\mathrm{C}_{12}V \longrightarrow \begin{cases} are & \text{if } V = be, \\ V & \text{otherwise;} \end{cases}$$
$$\mathrm{C}_{13}V \longrightarrow \begin{cases} is & \text{if } V = be, \\ has & \text{if } V = have, \\ V & \text{if } V \text{ is a modal verb,} \\ V+s & \text{otherwise;} \end{cases}$$
$$\mathrm{C}_{2k}\ be \longrightarrow \begin{cases} was & \text{if } k = 1 \text{ or } 3, \\ were & \text{if } k = 2; \end{cases}$$
$$\mathrm{C}_{2\mathrm{k}}V \longrightarrow \begin{cases} V^{\mathrm{p}} & \text{if } V \text{ is irregular other than } be, \\ V+ed & \text{otherwise.} \end{cases}$$

(2.2)
$$\mathrm{Neg}\ V \longrightarrow \begin{cases} V+not & \text{if } V \text{ is an auxiliary verb,} \\ do+not\ V & \text{otherwise.} \end{cases}$$

(2.3)
$$\begin{aligned} &\mathrm{Acc}\ I \longrightarrow me, \\ &\mathrm{Acc}\ he \longrightarrow him, \\ &\mathrm{Acc}\ she \longrightarrow her, \\ &\mathrm{Acc}\ we \longrightarrow us, \\ &\mathrm{Acc}\ they \longrightarrow them, \\ &\mathrm{Acc}\ X \longrightarrow X \text{ otherwise.} \end{aligned}$$

(2.4)
$$\mathrm{Perf}\ V \longrightarrow \begin{cases} V^{\mathrm{q}} & \text{if } V \text{ is an irregular verb,} \\ V+ed & \text{otherwise.} \end{cases}$$

(2.5)
$$\mathrm{Part}\ V \longrightarrow V+ing.$$

(2.6) Pass V Obj $\longrightarrow$ *be* Perf V -, where V is any transitive verb which has a passive.

Here are some comments.

(2.1) C_{ik} is an inflector which will act on any verb to produce its conjugation matrix, e.g. $C_{ik}go$ yields:

$$\begin{pmatrix} go & go & goes \\ went & went & went \end{pmatrix}.$$

In English, if we ignore the almost obsolete subjunctive, this matrix has $2 \times 3 = 6$ entries, of which usually only three are distinct. (The corresponding matrix in German has $4 \times 6 = 24$ entries, in French $7 \times 6 = 42$, in Latin $3 \times 5 \times 6 = 90$, in Hebrew $7 \times 2 \times 10 = 140$, in Arabic and Sanskrit many more.)

(2.2) Auxiliary verbs are the modal verbs, the verb *be* and the verb *have* when followed by the inflector Perf. (When followed by Obj, *have* counts as a transitive verb.) The negation discussed here is the grammatical negation, which may or may not coincide with the logical negation. For example, *can* + *not* is both the grammatical and the logical negation of *can*, whereas the grammatical negation *must* + *not* of *must* differs from its logical negation *need* + *not*.

(2.3) English has only few surviving accusative cases. In German, the inflector Acc acts visibly not only on pronouns, but also on nouns, adjectives and determiners.

(2.4) We assume the dictionary lists $be^q = been$, $have^q = had$, $go^q = gone$, etc. The inflector Perf will never appear before a modal verb.

(2.5) The inflector Part never appears before a modal verb either.

(2.6) Not all transitive verbs have passives, e.g. *resemble* does not. The reader may ignore the dash –, which denotes what Chomsky calls a "trace"; it plays no rôle in this brief exposition.

With the help of the new productions (2.1) to (2.6), we may almost compute our four sample sentences:

$$\begin{aligned} &S \longrightarrow he\ C_{11}\ call\ her \longrightarrow he\ call + s\ her, \\ &S \longrightarrow he\ C_{13}\ do + not\ go \longrightarrow he\ do + s + not\ go, \\ &S \longrightarrow he\ C_{13}\ will + not\ be\ \mathrm{Perf}\,call- \\ &\quad \longrightarrow he\ will + not\ be\ call + ed-, \\ &S \longrightarrow he\ C_{13}\ have\ be^q\ come + ing \longrightarrow he\ has\ been\ come + ing. \end{aligned}$$

All that remains is to compute the correct form of the morphemes +*not*, +*s*, +*ed*, +*ing*. We do this for the written forms; the spoken forms may be calculated in an analogous manner, but that would require a special notation for English phonemes.

Here are some sample spelling rules for +*not* and +*s*:

(3.1)
$$\begin{aligned} can + not &\longrightarrow cannot,\ can't; \\ will + not &\longrightarrow will\ not,\ won't; \\ do + not &\longrightarrow do\ not,\ don't; \\ does + not &\longrightarrow does\ not,\ doesn't;\ \text{etc.} \end{aligned}$$

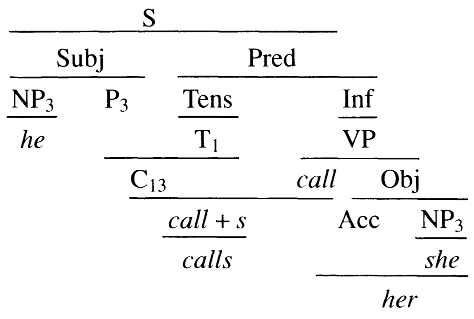
(3.2)
$$X + s \longrightarrow \begin{cases} Xes & \text{if } X \text{ ends in } z, s, x, sh \text{ or } ch, \text{ or in } o \text{ after a consonant,} \\ Yies & \text{if } X = Yy \text{ and } Y \text{ ends in a consonant or } qu, \\ Xs & \text{otherwise.} \end{cases}$$

We shall skip the rather tedious spelling rules for *+ing* and *+ed*.

Here is a complete derivation of our first sample sentence:

$$\begin{aligned}
\mathrm{S} &\longrightarrow \mathrm{Subj\ Pred}\\
&\longrightarrow \mathrm{NP_3\ P_3\ Tens\ Inf}\\
&\longrightarrow \mathrm{he\ P_3\ T_1\ VP}\\
&\longrightarrow \textit{he}\ \mathrm{C_{13}}\ \textit{call}\ \mathrm{Obj}\\
&\longrightarrow \textit{he call} + \textit{s}\ \mathrm{Acc\ NP_3}\\
&\longrightarrow \textit{he calls}\ \mathrm{Acc}\ \textit{she}\\
&\longrightarrow \textit{he calls her}
\end{aligned}$$

Alternatively, we may respresent this derivation by a parsing diagram:



Such diagrams serve the convenience of the grammarian; I do not believe that they have any psychological reality. Anyway, it should be clear that the diagram is not a tree. To obtain a tree one would have to delete everything below the words *he* $\mathrm{P_3}$ $\mathrm{T_1}$ *call* Acc *she*.

4. **Further discussion.** It is well-known to mathematicians that all and only recursively enumerable sets of symbols can be generated by production grammars from a single initial symbol. If we believe with Chomsky that the set of grammatical sentences of a natural language such as English is recursively enumerable, it becomes evident that the grammar of such a language can be presented in the form of productions.

What is not quite so obvious is how to write down the precise productions that are needed. This is like finding the postulates from which known theorems, as in geometry, can be deduced, an activity no mathematician need be ashamed of. In a sense, the speaker of a language "knows" such rules, though not consciously, and it may require a persistent Socratic process to uncover them.

Many authors feel uneasy about accepting unrestricted production grammars. For example, Cohn (1975) says: "... further restrictions are needed to impart a typical linguistic flavour. This is achieved by imposing restrictions on the rewriting rules." He then goes on to discuss context-sensitive, contextfree and finite-state productions. However, I hope to have convinced the reader that such restrictions, though interesting mathematically, are not at all natural in linguistics if one wants to treat morphology with the same ease as syntax. For example, one quarter of the productions used in deriving our sample sentence *he calls her* were not even context-sensitive, inasmuch as the strings on the right side of the arrows were shorter than those on the left. This is surprising, as English comes close to being a so-called "analytic" language in having only very few surviving inflections.

As we have seen, to check the grammaticality of an English sentence is like finding the proof of a theorem. On the other hand, to produce a sentence involves at least two steps: creating its underlying structure and then converting this into a string of English words. While the first step is clearly non-deterministic, the second step is very much like computing a given numerical function for given arguments. Actually, as I have shown elsewhere (1987), there is a simple three-tape machine which carries out both of these steps in producing English sentences and which is also capable of analyzing given strings of English words. Of course, the machine is equivalent to a Turing machine, but it resembles a combination of two pushdown automata.

A few words should be said about ambiguity. For example, in our kinship grammar of Section 2 we could derive $W \longrightarrow CSP$ in two different ways:

$$W \longrightarrow CW \longrightarrow CWP \longrightarrow CSP,$$
$$W \longrightarrow WP \longrightarrow CWP \longrightarrow CSP.$$

It seems reasonable to identify these two derivations. To give a convincing example of when two derivations should not be identified, let us consider a somewhat expanded form of our kinship grammar, which also includes the symbol

$$\Sigma = \text{spouse}.$$

In this grammar there would be two essentially inequivalent proofs that

$$R \longrightarrow \textit{sister-in-law} \text{ (in the context } \# - \#);$$

one derivation goes via the kinship description $F\Sigma S$, the other via $FS\Sigma$. Or again, consider the English sentence "we shall pay $1000 for your expenses and your airline ticket". There are two distinct ways of deriving this sentence, as the author has discovered to his cost.

To handle the questions raised by ambiguity or the lack of it one should introduce an equality relation between derivations in a production grammar. One then obtains a kind of two-category, to be precise, a strictly monoidal category (Hotz 1966, Benson 1975).

I have refrained from saying anything about the semantics of natural languages. This is a nontrivial problem, as the reader may gather from the brief discussion of grammatical versus logical negation in Section 3. Nonetheless, it has attracted the attention of a number of mathematicians and logicians. For example, a translation of

English into some form of type theory is implicit in the work of Curry (see e.g. Curry and Feys 1958), an idea which was considerably expanded by Montague (see e.g. Montague 1974). From a categorical point of view, this means that we interpret the given natural language in a Cartesian closed category or even a topos.

Benson (1970) had suggested that the interpretation should be viewed as a strictly monoidal functor from the production grammar, as categorized above, into the category of sets with canonical cartesian products. In place of the category of sets one might take any kind of category with a monoidal structure given by a Cartesian product, e.g. a topos, a Cartesian closed category or just a Cartesian category. One might in fact construct such a category freely from a production grammar by forcing the implicit tensor product given by concatenation to become a Cartesian product. Thus one might make mathematical sense out of the Sapir-Whorf thesis that the world as seen by a linguistic community is constructible from their language.

References

1. W. S. Anglin, *A mathematical analysis of a production grammar*, Theoretical Linguistics **13** (1986), 125–138.

2. Y. Bar-Hillel, M. Perlis and E. Shamir, *On formal properties of simple phrase structure grammars*, Z. Phonetik, Sprachwiss. Kommunikat. **14** (1961), 143–172.

3. D. B. Benson, *Syntax and semantics: a categorical view*, Information and Control **17** (1970), 145–160.

4. ——— *The basic algebraic structures in categories of derivations*, Information and Control **28** (1975), 1–29.

5. M. Bhargava and J. Lambek, *A production grammar for Hindi kinship terminology*, Theoretical Linguistics **10** (1983), 227–245.

6. ———, *A production grammar for Sanskrit kinship terminology*, McGill University Working papers in Cognitive Science **6** (1985).

7. B. Brainerd, *Introduction to the mathematics of language study*, Elsevier, New York 1971.

8. J. R. Buchler and H. A. Selby, *Kinship and social organization*, MacMillan, New York 1968.

9. N. Chomsky, *Syntactic structures*, Mouton, s'Gravenhage 1957.

10. ———, *Lectures on government and binding*, Foris Publications, Dordrecht 1982.

11. P. M. Cohn, *Algebra and language theory*, Bull. London Math. Soc. **7** (1975), 1–29.

12. H. B. Curry and R. Feys, *Combinatory Logic I*, North Holland, Amsterdam 1958.

13. W. H. Goodenough, *Yankee kinship terminology: a problem in componential analysis*. In: E. A. Hammel (ed.), *Formal semantics*, American Anthropologist **67** (5) Part 2 (1965), 259–287.

14. M. Gross, *Mathematical models in Linguistics*, Prentice Hall, Englewood Cliffs N.J. 1972.

15. G. Hotz, *Eindeutigkeit und Mehrdeutigkeit formaler Sprachen*, Elektronische Informationsverarbeitung und Kybernetik **2** (1966), 235–247.

16. S. C. Kleene, *Introduction to metamathematics*, Van Nostrand, New York 1952.

17. J. Lambek, *A mathematician looks at French conjugation*, Theoretical Linguistics **2** (1975), 203–214.

18. ——— *Computability and mathematical linguistics*, Notes for Mathematics 180–328, fourth edition, McGill University, Montreal 1976.

19. ——— *A mathematician looks at Latin conjugation*, Theoretical Linguistics **6** (1979), 221–224.

20. ——— *A production grammar for English kinship terminology*, Theoretical Linguistics, **13** (1986), 19–36.

21. ——— *Production grammars revisited*, Proceedings Sixth Amsterdam Colloquium, Amsterdam 1987, 175–195.

22. ——— and M. Lambek, *The kinship terminology of Malagasy speakers in Mayotte*, Anthropological Linguistics **22** (1981), 154–182.

23. F. G. Lounsbury, *Another view of Trobriand kinship categories.* In: E. H. Hammel (ed.) *Formal semantics*, American Anthropologist **67** (5) Part 2 (1965), 142–185.

24. R. Montague, *Formal philosphy, selected papers*, edited by R. H. Thomason, Yale University Press, New Haven 1974.

25. Webster's New Ideal Dictionary, Merrian Company, Springfield 1973.

McGill University