# Type Grammars as Pregroups

JOACHIM LAMBEK
*McGill University, Department of Mathematics, 805 Sherbrooke Street W., Burnside Hall, Montréal QC, Canada H3A 2K6*

## 1. Introduction

At first sight, it seems quite unlikely that mathematics can be applied to the study of natural language. However, upon closer examination, it appears that language itself is a kind of mathematics: producing and recognizing speech involves calculations, albeit at a subconscious level, and the rules of grammar which the speaker has mastered, even if she cannot formulate them, resemble the axioms and rules of inference of mathematical logic.

In this article I will present an algebraic model of grammar in the form of a *pregroup*, which competes with an earlier model which was once proposed by me and is now being developed further by a small but dedicated group of researchers, and took the form of a *residuated monoid*. I am not fully convinced that either of these models really captures the cognitive processes involved, and I still suspect that rewrite systems, also known as production grammars, do a better job. Yet the algebraic models offer an alternative approach of interest to the more mathematically inclined students of language.

Although I have told this story before (Lambek, 1999), the present version is addressed to linguists, hence some mathematical definitions have been deferred to an appendix and proofs have been left out altogether. While some red herrings have been eliminated (protogroups, inflectors), the small fragment of English grammar treated here is essentially the same as in the earlier version.

## 2. A Hierarchy of Types

The idea of a type grammar is to assign to every English word one or more types and to check whether a given string of words is a well-formed sentence by making a calculation on the corresponding string of types. Hopefully, such a program would push the entire grammar into the dictionary.

For the language of classical mechanics such a type grammar exists, namely the free Abelian group generated by three symbols: $L$ (= length), $T$ (= time) and $M$ (= mass). An equation $a = b$ between physical quantities is considered to be well-formed if the types of $a$ and $b$ coincide.

For a natural language, Abelian groups or even groups will not do. Surprisingly, however, a slight generalization of the notion of a group will work quite well. This

*Grammars* **4:** 21–39, 2001.

is the notion of a *pregroup*, an ordered monoid in which each element $a$ has a *left adjoint* $a^\ell$ such that:

$$a^\ell a \leqslant 1 \leqslant aa^\ell$$

and a *right adjoint* $a^r$ such that:

$$aa^r \leqslant 1 \leqslant a^r a.$$

For many purposes, the free pregroup generated by an ordered set of basic types suffices. It is presented here as a *hierarchy* of types, which can be understood even by people who do not know what a pregroup is.

We begin with a (partially) ordered set $A$ of *basic* types, that is, a set with a binary relation $\leqslant$ which is reflexive, transitive and antisymmetric:

$a \leqslant a$,
if $a \leqslant b$ and $b \leqslant c$ then $a \leqslant c$,
if $a \leqslant b$ and $b \leqslant a$ then $a = b$.

From the basic types we construct *simple* types:

$$\cdots, a^{\ell\ell}, a^\ell, a, a^r, a^{rr}, \cdots.$$

A *compound* type is a string of basic types.

The collection $F(A)$ of all compound types is partially ordered by the ordering inherited from $A$ and by the following rules:

*contractions*: $a^\ell a \leqslant 1, a^{\ell\ell} a^\ell \leqslant 1, aa^r \leqslant 1$, etc,
*expansions*: $1 \leqslant aa^\ell, 1 \leqslant a^\ell a^{\ell\ell}, 1 \leqslant a^r a$, etc.

It is to be understood that the order relation of $F(A)$ satisfies the substitution rule:

if $\alpha \leqslant \beta$ then $\gamma\alpha\delta \leqslant \gamma\beta\delta$.

$F(A)$ is our hierarchy of types; technically speaking, it is the free pregroup generated by $A$, but the mathematically unsophisticated reader can ignore this. We shall assign compound types to English words and other grammatical entities. To check that a string of words, say of type $\alpha$, is a sentence, say a statement of type $\mathbf{s}$, we must verify that $\alpha \leqslant \mathbf{s}$. As will be seen in the Appendix, as long as we are interested only in showing that a string of English words has a simple type, the expansions are not needed.

## 3. Assigning Types to Verbs

To apply the above program to a small fragment of English we take as the set of basic types:

$$A = \{\pi_1, \pi_2, \pi_3, \mathbf{s}_1, \mathbf{s}_2, \mathbf{p}_1, \mathbf{p}_2, \mathbf{o}, \cdots\}.$$

Here:

$\pi_k = k$ - the person singular (all plurals in modern English being given the type $\pi_2$ of *you*),

$\mathbf{s}_i =$ statements, with $i = 1$ for the present tense and $i = 2$ for the past,

$\mathbf{p}_i =$ participles, again in the present or past tense,

$\mathbf{o} =$ object.

In the interest of a gentle introduction, we assume for the moment that $A$ is just a set, in other words, that the order is discrete. The following sample sentences will convey the main idea:

*she goes*

of type $\pi_3(\pi_3^r\mathbf{s}_1) = (\pi_3\pi_3^r)\mathbf{s}_1 \leqslant \mathbf{s}_1$,

*you had gone*

of type $\pi_2(\pi_2^r\mathbf{s}_2\mathbf{p}_2^\ell)\mathbf{p}_2 = (\pi_2\pi_2^r)\mathbf{s}_2(\mathbf{p}_2^\ell\mathbf{p}_2) \leqslant \mathbf{s}_2$,

*I saw her*

of type $\pi_1(\pi_1^r\mathbf{s}_2\mathbf{o}^\ell)\mathbf{o} = (\pi_1\pi_1^r)\mathbf{s}_2(\mathbf{o}^\ell\mathbf{o}) \leqslant \mathbf{s}_2$,

*he is seeing me*

of type $\pi_3(\pi_3^r\mathbf{s}_1\mathbf{p}_1^\ell)(\mathbf{p}_1\mathbf{o}^\ell)\mathbf{o} = (\pi_3\pi_3^r)\mathbf{s}_1(\mathbf{p}_1^\ell\mathbf{p}_1)(\mathbf{o}^\ell\mathbf{o}) \leqslant \mathbf{s}_1$,

*we were seen -*

of type $\pi_2(\pi_2^r\mathbf{s}_2\mathbf{o}^{\ell\ell}\mathbf{p}_2^\ell)(\mathbf{p}_2\mathbf{o}^\ell) = (\pi_2\pi_2^r)\mathbf{s}_2(\mathbf{o}^{\ell\ell}(\mathbf{p}_2^\ell\mathbf{p}_2)\mathbf{o}^\ell) \leqslant \mathbf{s}_2$,

*they have been seen -*

of type $\pi_2(\pi_2^r\mathbf{s}_1\mathbf{p}_2^\ell)(\mathbf{p}_2\mathbf{o}^{\ell\ell}\mathbf{p}_2^\ell)(\mathbf{p}_2\mathbf{o}^\ell) = (\pi_2\pi_2^r)\mathbf{s}_1(\mathbf{p}_2^\ell\mathbf{p}_2)(\mathbf{o}^{\ell\ell}(\mathbf{p}_2^\ell\mathbf{p}_2)\mathbf{o}^\ell) \leqslant \mathbf{s}_1$.

Here the dash denotes a Chomskyan trace. Although traces are not really needed for our description, we put them in for comparison with the literature. The double $\ell$ in $\mathbf{o}^{\ell\ell}$ is how type theory reflects traces.

We note that the finite forms of the verbs $go$, $see$, $have$ and *be* may be arranged in conjugation matrices (ignoring the almost obsolete subjunctive):

$$\begin{bmatrix} go & go & goes \\ went & went & went \end{bmatrix} \quad \text{of type} \quad \pi_k^r\mathbf{s}_i,$$

$$\begin{bmatrix} \textit{see} & \textit{see} & \textit{sees} \\ \textit{saw} & \textit{saw} & \textit{saw} \end{bmatrix} \quad \text{of type} \quad \pi_k^r \mathbf{s}_i \mathbf{o}^\ell,$$

$$\begin{bmatrix} \textit{have} & \textit{have} & \textit{has} \\ \textit{had} & \textit{had} & \textit{had} \end{bmatrix} \quad \text{of type} \quad \pi_k^r \mathbf{s}_i \mathbf{p}_2^\ell,$$

$$\begin{bmatrix} \textit{am} & \textit{are} & \textit{is} \\ \textit{was} & \textit{were} & \textit{was} \end{bmatrix} \quad \text{of type} \quad \pi_k^r \mathbf{s}_i \mathbf{p}_1^\ell \quad \text{or} \quad \pi_k^r \mathbf{s}_i \mathbf{o}^{\ell\ell} \mathbf{p}_2^\ell.$$

The choice in the last type assignment depends on whether the auxiliary verb *be* is used for the progressive aspect or the passive voice respectively.

In addition, we have the non-finite forms:

*going, gone* of types $\mathbf{p}_i$,

*seeing, seen* of types $\mathbf{p}_i \mathbf{o}^\ell$,

*being, been* of types $\mathbf{p}_i \mathbf{o}^{\ell\ell} \mathbf{p}_2^\ell$,

the last for the passive auxiliary *be*. The progressive auxiliary *be* does not admit a present participle, since one cannot say:

**I am being going,*

but it has the form:

*been* of type $\mathbf{p}_2 \mathbf{p}_1^\ell$,

as in:

*I have been going*.

While the participles *having* and *had* exist when *have* is used as a transitive verb, they are not admissible for the perfect auxiliary *have*:

* *I am having gone,*

* *I have had gone.*

Let us take another look at the example:

*they have been seen* – ,

as a hearer might analyze it step by step. The initial segments have types:

$\pi_2$,

$\pi_2(\pi_2^r \mathbf{s}_1 \mathbf{p}_2^\ell) \leqslant \mathbf{s}_1 \mathbf{p}_2^\ell$,

$(\mathbf{s}_1 \mathbf{p}_2^\ell)(\mathbf{p}_2 \mathbf{o}^{\ell\ell} \mathbf{p}_2^\ell) \leqslant \mathbf{s}_1 \mathbf{o}^{\ell\ell} \mathbf{p}_2^\ell$,

$(\mathbf{s}_1 \mathbf{o}^{\ell\ell} \mathbf{p}_2^\ell)(\mathbf{p}_2 \mathbf{o}^\ell) \leqslant \mathbf{s}_1$.

## 4. The Auxiliary *do* and the Modal Verbs

So far we have ignored the infinitive. We shall assign type $\mathbf{i}$ to the infinitive of an intransitive verb such as *go* or *snore*, hence type $\mathbf{io}^\ell$ to the infinitive of a transitive verb such as *see* or *like*. If we want to emphasize one of these verbs, we use the auxiliary verb *do* with conjugation matrix:

$$\begin{bmatrix} do & do & does \\ did & did & did \end{bmatrix} \quad \text{of type } \pi_k^r \mathbf{s}_i \mathbf{i}^\ell.$$

However, the auxiliary verbs *have* and *be* are emphasized by bearing stress directly:

$^*$ *he does have gone*,

$^*$ *she does be snoring*,

$^*$ *they do be seen* – .

To prevent the last three constructions, we assign a new basic type $\mathbf{j}$ to such compound infinitives as *have gone*, *be seeing* and *be seen* –. It follows that the infinitives of the perfect, progressive and passive auxiliaries have types $\mathbf{jp}_2^\ell$, $\mathbf{jp}_1^\ell$ and $\mathbf{jo}^{\ell\ell}\mathbf{p}_2^\ell$ respectively. The emphatic auxiliary *do* has no non-finite forms:

$^*$ *I am doing go,*

$^*$ *you have done go,*

$^*$ *he may do go.*

Let us now turn to the modal auxiliaries *may*, *can*, *shall*, *will* and *must*. Except for the last, they are accompanied by the forms *might*, *could*, *should* and *would*, which it is convenient to regard as their past tenses, if only for historical reasons. None of them possess non-finite forms; the finite forms are illustrated by:

$$\begin{bmatrix} may & may & may \\ might & might & might \end{bmatrix} \quad \text{of type } \pi_k^r \mathbf{s}_i \mathbf{j}^\ell,$$

with a noticeable absence of the morpheme +*s* in the third person of the present tense. The modal verbs act on all infinitives, both simple and compound, hence we postulate $\mathbf{i} \leqslant \mathbf{j}$. Thus we have:

*he might have gone*

of type $\pi_3(\pi_3^r \mathbf{s}_2 \mathbf{j}^\ell)(\mathbf{jp}_2^\ell)\mathbf{p}_2 \leqslant \mathbf{s}_1$ and:

*I may go*

of type $\pi_1(\pi_1^\ell \mathbf{s}_1 \mathbf{j}^\ell)\mathbf{i} \leqslant (\pi_1 \pi_1^\ell)(\mathbf{s}_1(\mathbf{j}^\ell \mathbf{j}) \leqslant \mathbf{s}_1$. On the other hand:

$^*$ *he does have gone*

has type $\pi_3(\pi_3^r \mathbf{s}_1 \mathbf{i}^\ell)(\mathbf{j}\mathbf{p}_2^\ell)\mathbf{p}_2 \leqslant \mathbf{s}_1\mathbf{i}^\ell\mathbf{j}$, which does not contract further.

Once we have decided to allow the basic types to be ordered, we may introduce some further basic types:

$\mathbf{s}$ for statements when the tense does not matter,

$\pi$ for nominative pronouns when the person does not matter,

and postulate:

$$\pi_k \leqslant \pi, \quad \mathbf{s}_i \leqslant \mathbf{s}.$$

This would allow us to retype *may* with $\pi^r \mathbf{s}_1 \mathbf{j}^\ell$ and *went* with $\pi^r \mathbf{s}_2$, et cetera, and to assign type $\mathbf{i}\mathbf{s}^\ell$ to such verbs as *say* and *know*, and even as a new type to *see*, to account for such sentences as:

*he said she snores*,
*she knows he snored*,
*I see you have gone*.

## 5. Negation

The grammatical negation of an English sentence is obtained by introducing the word *not* of type $xx^\ell$, where $x = \mathbf{i}, \mathbf{j}, \mathbf{p}_1$ or $\mathbf{p}_2$, at an appropriate place. This should account for:

*she does not go*,
*she is not going*,
*I was not seen*$-$,
*you have not gone*,
*we will not be going*;

but:

**she goes not*.

For example, the third sample sentence has type:

$$\pi_1(\pi_1^r \mathbf{s}_2 \mathbf{o}^{\ell\ell} \mathbf{p}_2^\ell)(\mathbf{p}_2\mathbf{p}_2^\ell)(\mathbf{p}_2\mathbf{o}^\ell) \leqslant \mathbf{s}_2\mathbf{o}^{\ell\ell}\mathbf{o}^\ell \leqslant \mathbf{s}_2,$$

and the fifth has type:

$$\pi_2(\pi^r \mathbf{s}_1 \mathbf{j}^\ell)(\mathbf{j}\mathbf{j}^\ell)(\mathbf{i}\mathbf{p}_1^\ell)\mathbf{p}_1 \leqslant \mathbf{s}_1,$$

since $\pi_2 \leqslant \pi$ and $\mathbf{i} \leqslant \mathbf{j}$.

We note that *does not*, *is not*, *was not*, *has not*, and *will not* are often abbreviated as *doesn't*, *isn't*, *wasn't*, *hasn't* and *won't* respectively. This is not too surprising, according to our analysis, since *does not* has type:

$$(\pi_3^r \mathbf{s}_1 \mathbf{i}^\ell)(\mathbf{i}\mathbf{i}^\ell) = \pi_3^r \mathbf{s}_1 \mathbf{i}^\ell,$$

in view of the equation $\mathbf{i}^\ell \mathbf{i}\mathbf{i}^\ell = \mathbf{i}^\ell$ (see Appendix). Hence *doesn't* has the same type as *does* and may be regarded as a single verb form.

It should be pointed out that the grammatical negation treated here is not necessarily the same as the logical negation. For example, the logical negation of *he must go* is *he need not go* and not *he must not go*, which is really the logical negation of *he may go*. On the other hand, the logical negation of *he can go* is indeed *he cannot go*, which perhaps explains why *cannot* is spelled as one word.

The modal verbs all lack the morpheme $+s$ in the third person and they act on the bare infinitive without *to*. This is not so for the semi-modal verbs *need* and *dare* when used affirmatively, but it is the case when they are negated:

*he needs to go*,
*he dares to go*,
*he need not go*,
*he dare not go*.

## 6. Questions

We distinguish between yes-or-no questions of type $\mathbf{q}_1$ in the present tense and $\mathbf{q}_2$ in the past tense. To handle wh-questions, we introduce the type $\mathbf{q}'$ for the class of all questions. We assume that the tense of a wh-question is irrelevant, it being a present request for information about the present or past. When the tense of a yes-or-no question is irrelevant, we assign it a new basic type $\mathbf{q}$ and postulate:

$$\mathbf{q}_1 \leqslant \mathbf{q}, \quad \mathbf{q}_2 \leqslant \mathbf{q}, \quad \mathbf{q} \leqslant \mathbf{q}'.$$

Sample yes-or-no questions are:

*does she go*?
*is she going*?
*was I seen* – ?
*have you gone*?
*will we be going*?

These may be accounted for by the following:

METARULE. If the finite form of an auxiliary verb has type $\pi_k^r \mathbf{s}_i x^\ell$ (or $\pi^r \mathbf{s}_i x^\ell$), then the same form, with rising intonation, can be assigned the type $\mathbf{q}_i x^\ell \pi_k^\ell$ (or $\mathbf{q}_i x^\ell \pi^\ell$).

In the above sample sentences, $x = \mathbf{i}$, $\mathbf{p}_1$, $\mathbf{p}_2\mathbf{o}^\ell$, $\mathbf{p}_2$ and $\mathbf{j}$ respectively, hence:

*does, is, was, have, will*,

as occurring there, have types:

$$\mathbf{q}_1\mathbf{i}^\ell\pi_3^\ell, \quad \mathbf{q}_1\mathbf{p}_1^\ell\pi_3^\ell, \quad \mathbf{q}_2\mathbf{o}^{\ell\ell}\mathbf{p}_2^\ell\pi_1^\ell, \quad \mathbf{q}_1\mathbf{p}_2^\ell\pi_2^\ell, \quad \mathbf{q}_1\mathbf{j}^\ell\pi^\ell$$

respectively. (Note that $(\mathbf{p}_2\mathbf{o}^\ell)^\ell = \mathbf{o}^{\ell\ell}\mathbf{p}_2^\ell$, see the Appendix.)

To simplify the discussion of wh-questions, we shall confine attention to the single question word *whom*. I realize that many people replace this by *who*, but I shall follow Inspector Morse in distinguishing between the two. Anyway, these two words have different types.

From the yes-or-no questions:

*does he see her*? *did he see her*?

of respective types:

$$(\mathbf{q}_1\mathbf{i}^\ell\pi_3^\ell)\pi_3(\mathbf{i}\mathbf{o}^\ell)\mathbf{o}, \quad (\mathbf{q}_2\mathbf{i}^\ell\pi^\ell)\pi_3(\mathbf{i}\mathbf{o}^\ell)\mathbf{o}$$

one may form:

*whom does he see* –? *whom did he see* –?

Having assigned type $\mathbf{q}'$ to these wh-questions we obtain the type $\mathbf{q}'\mathbf{o}^{\ell\ell}\mathbf{q}^\ell$ for *whom*.

Surprisingly, the same type will work to account for:

*whom (did you say) (he saw –)*?

As bracketed, this has type:

$$(\mathbf{q}'\mathbf{o}^{\ell\ell}\mathbf{q}^\ell)(\mathbf{q}_2\mathbf{s}^\ell)(\mathbf{s}_2\mathbf{o}^\ell) \leqslant \mathbf{q}'\mathbf{o}^{\ell\ell}\mathbf{o}^\ell \leqslant \mathbf{q}',$$

since $\mathbf{q}_2 \leqslant \mathbf{q}$ and $\mathbf{s}_2 \leqslant \mathbf{s}$.

## 7. The Noun Phrase

So far, the only noun phrases considered were pronouns of type $\pi_k$ $(k = 1, 2, 3)$ and accusative pronouns of type $\mathbf{o}$, such as *me* and *her*. To these we should add names such as *Jane*, *Napoleon Bonoparte*, *Inspector Morse*, et cetera, of type $\mathbf{n}$, and we must stipulate:

$$\mathbf{n} \leqslant \pi_3, \quad \mathbf{n} \leqslant \mathbf{o},$$

to allow e.g. for *Jane kissed Bill*.

Most noun phrases are constructed from nouns, where we distinguish:

mass nouns of type $\mathbf{m}$, e.g. *water, pork;*
count nouns of type $\mathbf{c}$, e.g. *apple, pig;*
plurals of type $\mathbf{p}$, e.g. *police, pigs.*

Mass nouns and plurals may appear without determiners, so we should ensure that:

$$\mathbf{m} \leqslant \pi_3, \quad \mathbf{m} \leqslant \mathbf{o}, \quad \mathbf{p} \leqslant \pi_2, \quad \mathbf{p} \leqslant \mathbf{o},$$

as we will do further down.

Some plurals, like *police* and *pants* have no corresponding singular form, but most are formed from singular count nouns. Bearing in mind a few exceptions, such as *mouse/mice,* such plurals are formed with the help of a morpheme $+s$ of type $\mathbf{c}^r\mathbf{p}$. This is the same morpheme, though not of the same type, as that used for forming the third person of verbs. It is subject to certain orthographic conventions, as exemplified by:

*knife* $+ s \geqslant$ *knives, kiss* $+ s \geqslant$ *kisses, go* $+ s \geqslant$ *goes.*

Noun phrases usually contain a determiner. While the determiner *the* is common to mass nouns, count nouns and plurals, some determiners distinguish between these types of nouns, e.g.:

*much pork, many a pig, many pigs*.

Let us assign to these noun phrases the types $\overline{\mathbf{m}}$, $\overline{\mathbf{c}}$ and $\overline{\mathbf{p}}$ respectively. Thus, the determiners:

*much, many a, many*

have the respective types:

$$\overline{\mathbf{m}}\mathbf{m}^\ell, \quad \overline{\mathbf{c}}\mathbf{c}^\ell, \quad \overline{\mathbf{p}}\mathbf{p}^\ell,$$

whereas *the* has all three of these types. We postulate:

$$\mathbf{m} \leqslant \overline{\mathbf{m}} \leqslant \pi_3, \mathbf{o}; \quad \overline{\mathbf{c}} \leqslant \pi_3, \mathbf{o}; \quad \mathbf{p} \leqslant \overline{\mathbf{p}} \leqslant \pi_2, \mathbf{o}.$$

There seems to be no harm in assuming that $\overline{\mathbf{m}} = \overline{\mathbf{c}}$.

There is some type shifting between the categories $\mathbf{n}$, $\mathbf{m}$ and $\mathbf{c}$. Names and mass nouns can be used as count nouns, with a change in meaning as in:

*I will have two beers*, *I met the real McCoy*,

and count nouns can occasionally be used as mass nouns, as in:

*cannibals prefer man to pork*.

## 8. Indirect Sentences

In addition to noun phrases constructed from nouns, there are those constructed from sentences, namely indirect statements and questions such as:

*that he snores*, *whether he went*,

to which we will assign types $\bar{\mathbf{s}}$ and $\bar{\mathbf{q}}$ respectively. I suppose these can, in principle, always appear in subject position, as in:

*that he snores annoys me*,

*whether he went does not concern me*.

I admit that one can then also produce some pretty odd sounding sentences, e.g.:

*that he snores eats snails*,

but I don't see how to easily rule these out on syntactic grounds, any more than Chomsky's:

*colorless green ideas sleep furiously*,

without complicating the grammar tremendously. One author asserted that the subject of *eat* must be animate, but he over-argued his case by saying that "stones do not eat", thus using the very construction he claimed to be ungrammatical.

On the other hand, indirect sentences can appear in object position only after certain verbs, whose types must be specified accordingly, so $\bar{\mathbf{s}} \not\leq \mathbf{o}$ and $\bar{\mathbf{q}} \not\leq \mathbf{o}$. For example, in:

*he says that he does not snore*,

*I don't know whether he snores*,

*say* has type $\mathbf{i}\bar{\mathbf{s}}^\ell$ and *know* has type $\mathbf{i}\bar{\mathbf{q}}^\ell$. It is tempting to identify $\bar{\mathbf{s}}$ with $\bar{\mathbf{q}}$, but the examples:

*I think that he snores*,

$^*$*I think whether he snores*,

show that this would be wrong.

It is more difficult to explain why usually the complementizer *that* of type $\bar{\mathbf{s}}\mathbf{s}^\ell$ can disappear in object position (except, for mysterious reasons, before a trace), but not in subject position:

*he says he does not snore*, $^*$*he snores annoys me*;

*he was told – he snores*, *he was told – that he snores*;

*who do you think – snores*, $^*$*who do you think that – snores*.

Among indirect sentences there are also indirect wh-questions such as:

*I don't know whom he saw –.*

In this context, *whom* should be assigned the type $\overline{\mathbf{q}}\mathbf{o}^{\ell\ell}\mathbf{s}^{\ell}$, which will also take care of:

*I don't know whom she says he saw –.*

In this connection, we should also mention such infinitival noun phrases as:

*whether to go, whom to see –.*

We introduce a new basic type $\bar{\mathbf{i}}$ for infinitives with *to* and assign type $\bar{\mathbf{i}}\mathbf{i}^{\ell}$ to *to*, $\overline{\mathbf{q}}\bar{\mathbf{i}}^{\ell}$ to *whether* and yet another type $\overline{\mathbf{q}}\mathbf{o}^{\ell\ell}\bar{\mathbf{i}}^{\ell}$ to *whom*.

## 9. Relative Clauses

We distinguish between restrictive relative clauses that modify a noun, as in:

*the man whom I saw –,*
*people whom I know –,*

and non-restrictive ones as in:

*John, whom she loved –*
*the chairman, whom she knows I fear –,*
*the Chinese, whom I like –.*

In these sample noun phrases, the relative clause has type $x^r x$, where $x = \mathbf{c}$, $\mathbf{p}$, $\mathbf{n}$, $\overline{\mathbf{c}}$ or $\overline{\mathbf{p}}$, hence the relative pronoun *whom* has type $x^r x\mathbf{o}^{\ell\ell}\mathbf{s}^{\ell}$.

Incidentally, the relative pronoun *whom* and the question word *whom* have distinct inanimate variants *which* and *what* respectively; but we shall ignore the latter in the small fragment of English grammar considered here.

The restrictive *whom* may be replaced by *that* or even by the zero morpheme, here denoted by the symbol Ø:

*the man that I saw –,*
*the man Ø I saw –.*

One way to account for this is to assign to *that* and Ø the same type as to the restrictive *whom*, namely $x^r x\mathbf{o}^{\ell\ell}\mathbf{s}^{\ell}$, where $x = \mathbf{c}$ or $\mathbf{p}$. Although we can easily imagine a speaker introducing a zero morpheme at an appropriate place in a sentence generated, say by dropping *that*, it is more difficult to see how a hearer would recognize a zero morpheme.

I think a simpler and more honest way to describe this situation is to admit a new basic type $\mathbf{r}$ for relative clauses, to postulate the *reduction rules:*

$$x\mathbf{r} \leqslant x, \quad \bar{\mathbf{s}}\mathbf{o}^{\ell} \leqslant \mathbf{r}, \quad \mathbf{s}\mathbf{o}^{\ell} \leqslant \mathbf{r} \tag{R1}$$

and to assign the type $\mathbf{r}\mathbf{o}^{\ell\ell}\mathbf{s}^{\ell}$ to the restrictive *whom*.

Unfortunately, the rules (R1) do not describe the order relation on the set of *basic* types, so we must abandon the attempt to formulate the grammar in terms of the *free* pregroup generated by an ordered set. In other words, the types no longer form an inductively defined *hierarchy* and we cannot claim that the whole grammar has been pushed into the dictionary: the reduction rules (R1) constitute additional grammatical rules.

## 10. Adjectives and Adverbs

Noun phrases may also include adjectives, as in:

*the sick old man*,
*stagnant water*,
*poor old John*.

We are tempted to assign the type $xx^{\ell}$, with $x = \mathbf{c}$, $\mathbf{m}$ or $\mathbf{n}$ to adjectives. But this will not explain why one can say:

*the sick man is old*,
*the old man is sick*,

but not:

**the man is sick old*.

It might be possible to get around this difficulty by special tricks, e.g. by assigning a new basic type $\mathbf{a}$ to all adjectives, in addition to the above mentioned $xx^{\ell}$, and to assign yet another type $\mathbf{j}\mathbf{a}^{\ell}$ to the copula *be*, with appropriate modification for its finite forms. However, it seems more natural to avoid this multiple type assignment to the large class of adjectives and to retain just the basic type $\mathbf{a}$ and to postulate the reduction rules:

$$\mathbf{ac} \leqslant \mathbf{c}, \quad \mathbf{am} \leqslant \mathbf{m}, \quad \mathbf{an} \leqslant \mathbf{n}. \tag{R2}$$

Once we abandon the idea that the set of types should be a free pregroup, we can also handle adverbs more easily than in my earlier paper (Lambek, 1999), by assigning to all adverbs the basic type $\alpha$ and to postulate the reduction rules:

$$\mathbf{i}\alpha \leqslant \mathbf{i}, \quad \mathbf{j}\alpha \leqslant \mathbf{j}, \quad \mathbf{p}_1\alpha \leqslant \mathbf{p}_1, \quad \mathbf{p}_2\alpha \leqslant \mathbf{p}_2. \tag{R3}$$

The second of these is probably not necessary and the last two could have been avoided by the strategy of Lambek (1999), where participles of intransitive verbs had been assigned the type $(\mathbf{p}_i\mathbf{i}^\ell)\mathbf{i}(i = 1, 2)$.

There remains, however, the problem of discontinuous dependencies, such as:

*whom did you see – yesterday*?

*I don't know whom you saw – yesterday*.

The easiest way to take care of these is to adopt as further reduction rules:

$$\mathbf{q}'\alpha \leqslant \mathbf{q}', \quad \overline{\mathbf{q}}\alpha \leqslant \overline{\mathbf{q}}. \tag{R4}$$

A more complicated solution had been proposed in Lambek (1999).

It seems reasonable to assign type $\alpha$ also to adverbial phrases such as *with Jane* and *when you left*. Then we should assign the type $\alpha\mathbf{o}^r$ to the preposition *with* and the type $\alpha\mathbf{s}^r$ to the conjunction *when*.

## 11. Constraints

Much attention has been paid to constraints on wh-transformations in the literature (McCawley, 1988). Similar constraints must of course affect our typing of *whom*. Typical examples of non-sentences are:

$^*$*whom did she see John and* –?

of type $(\mathbf{q}'\mathbf{o}^{\ell\ell}\mathbf{q}^\ell)(\mathbf{q}_2\mathbf{i}^\ell\pi^\ell)\pi_3(\mathbf{i}\mathbf{o}^\ell)\mathbf{n}(\mathbf{o}^r\mathbf{o}\mathbf{o}^\ell)$ and:

$^*$*whom do I know people who like* –?

of type $(\mathbf{q}'\mathbf{o}^{\ell\ell}\mathbf{q}^\ell)(\mathbf{q}_1\mathbf{i}^\ell\pi_1^\ell)\pi_1(\mathbf{i}\mathbf{o}^\ell)\mathbf{p}(\mathbf{r}\mathbf{s}^\ell\pi_2)(\pi_2^r\mathbf{s}_1\mathbf{o}^\ell)$. In both examples we can calculate the type $\mathbf{q}'$, but some constraint should prevent this.

In the first example we are first led to calculate:

$$\leqslant \mathbf{q}'\mathbf{o}^{\ell\ell}\mathbf{o}^\ell\mathbf{n}\mathbf{o}^r\mathbf{o}\mathbf{o}^\ell$$

and are now tempted to contract $\mathbf{o}^\ell\mathbf{n} \leqslant 1$ (since $\mathbf{n} \leqslant \mathbf{o}$) or $\mathbf{o}^{\ell\ell}\mathbf{o}^\ell \leqslant 1$, but must defer both of these contractions until after the contraction $\mathbf{n}\mathbf{o}^r \leqslant 1$ has been carried out to yield:

$$\leqslant \mathbf{q}'(\mathbf{o}^{\ell\ell}(\mathbf{o}^\ell\mathbf{o})\mathbf{o}^\ell) \leqslant \mathbf{q}'.$$

In the second example we calculate the following type for the initial fragment before *like*:

$$\mathbf{q}'\mathbf{o}^{\ell\ell}\mathbf{o}^\ell\mathbf{p}\mathbf{r}\mathbf{s}^\ell\pi_2.$$

On the way there we had to put aside the temptation to contract $\mathbf{o}^{\ell\ell}\mathbf{o} \leqslant 1$ and $\mathbf{o}^{\ell}\mathbf{p} \leqslant 1$ (since $\mathbf{p} \leqslant \mathbf{o}$). We may now apply the reduction rule $\mathbf{pr} \leqslant \mathbf{p}$ and obtain:

$$\leqslant \mathbf{q}'\mathbf{o}^{\ell\ell}(\mathbf{o}^{\ell}\mathbf{p})\mathbf{s}^{\ell}\pi_2 \leqslant \mathbf{q}'\mathbf{o}^{\ell\ell}\mathbf{s}^{\ell}\pi_2.$$

Multiplying this by the type $\pi_2^r\mathbf{s}_1\mathbf{o}^{\ell}$ of *like*, we finally obtain:

$$\mathbf{q}'(\mathbf{o}^{\ell\ell}(\mathbf{s}^{\ell}(\pi_2\pi_2^r)\mathbf{s}_1)\mathbf{o}^{\ell}) \leqslant \mathbf{q}'.$$

Although, on the surface, the two examples studied here look quite different, they both contain an expression of the form $x^{\ell\ell}x^{\ell}x$ (with $x = \mathbf{o}$, if we apply $\mathbf{p} \leqslant \mathbf{o}$), in which both tempting contractions $x^{\ell\ell}x^{\ell} \leqslant 1$ and $x^{\ell}x \leqslant 1$ have to be deferred. Perhaps this observation can be utilized in framing a constraint on wh-transformations.

## 12. The Invisible Dative Case

Morphologically, English does not distinguish between direct and indirect objects, though it does so syntactically and semantically. Consider the sentences:

*she gave him roses*,
*she taught him English*,
*she told him (that) I snore*.

Here *roses* and *English* are direct objects of types $\mathbf{p} \leqslant \mathbf{o}$ and $\mathbf{m} \leqslant \mathbf{o}$ respectively and *I snore* is a statement of type $\mathbf{s}_1 \leqslant \mathbf{s}$. We shall assign the type $\mathbf{o}'$ to the indirect object *him* in all three examples, the type $\mathbf{io}^{\ell}\mathbf{o}'^{\ell}$ to the bitransitive verbs *give* and *teach* and the type $\mathbf{is}^{\ell}\mathbf{o}'^{\ell}$ or $\mathbf{i}\bar{\mathbf{s}}^{\ell}\mathbf{o}'^{\ell}$ to *tell*.

German, a sister language of English, does distinguish between a dative *ihm* and an accusative *ihn*, both of which are rendered by *him* in English. To acknowledge the fact that, in English, the accusative case also serves as the dative, we postulate $\mathbf{o} \leqslant \mathbf{o}'$.

However, the dative case can be made visible with the help of the proposition *to*:

*she gave roses to him,*
*she taught English to him,*
*she told that I snore to him,*

although one might object to the last on stylistic grounds. If we assign type $\omega$ to the prepositional phrase *to him*, we are led to give alternative types to *give*, *teach* and *tell*, namely $\mathbf{i}\omega^{\ell}\mathbf{o}^{\ell}$ to the first two and $\mathbf{i}\omega^{\ell}\bar{\mathbf{s}}^{\ell}$ to the third.

It is a peculiarity of modern English that the indirect object can become the subject in the passive:

*he was given – roses*,

*he was taught – English*,

*he was told – (that) I snore*.

To account for these passives we need only to introduce a new type for *be*, namely $\mathbf{j}\omega^{\ell\ell}\mathbf{p}_2^\ell$, so that *was* will receive type $\pi_3^\ell \mathbf{s}_2 \omega^{\ell\ell}\mathbf{p}_2^\ell$.

As I have pointed out elsewhere, indirect objects, bearing an invisible dative case, semantically denote the object of causal actions. For instance, the three sample sentences at the beginning of this section may be paraphrased roughly thus:

*she let him have roses*,

*she made him learn English*,

*she let him know (that) I snore*.

The dative case may also apply to *whom*. Note that German distinguishes between the accusative *wen* and the dative *wem*. Consider the following questions:

*whom did she give – roses*?

*whom did she tell – (that) I snore*?

Here *whom* has type $\mathbf{q}' x^\ell \mathbf{o}'^{\ell\ell} x^{\ell\ell} \mathbf{q}^\ell$ with $x = \mathbf{o}, \mathbf{s}$ or $\bar{\mathbf{s}}$. Similarly, in the indirect questions:

*whom she gave – roses*,

*whom she told – (that) I snore*

*whom* has type $\overline{\mathbf{q}}\omega^{\ell\ell}\mathbf{s}^\ell$ with $x$ inserted as above. These same expressions may also be read as relative clauses, but then $\overline{\mathbf{q}}$ should be replaced by $\mathbf{r}$ in the type of *whom*.

## 13. Conclusions

We have attempted a formulation of English grammar based on an assignment of types to English words, where the types are elements of a *pregroup*, an ordered monoid in which each element has both a left and a right adjoint.

One had hoped that this pregroup was *freely* generated by a partially ordered set of basic types, so that all grammatical rules would be implicit in the dictionary. Reluctantly one realized that this position was difficult to maintain and that further grammatical rules had to be introduced, such as the *reduction* rules (R1) to (R4) in Sections 9 and 10. These may be written in the form $ab \leqslant c$, where $a, b$ and $c$ are simple types. They could be avoided if we allowed *zero morphemes* of type $a^r c b^\ell$, since:

$$a(a^r c b^\ell)b = (aa^r)c(b^\ell b) \leqslant c.$$

However, zero morphemes do not resolve the problem but merely rephrase it. While they cause no difficulty for the speaker, they present the hearer with the problem of where to insert them.

I see no way of eliminating (R1), except by the dubious device of introducing zero morphemes, but (R2) could have been avoided by assigning to each adjective the types $\mathbf{a}$, $\mathbf{c}\mathbf{c}^\ell$, $\mathbf{m}\mathbf{m}^\ell$ and $\mathbf{n}\mathbf{n}^\ell$. Similarly, (R3) and (R4) could have been avoided by assigning to each adverb (and to each adverbial phrase) the types $\mathbf{i}^r\mathbf{i}$, $\mathbf{j}^r\mathbf{j}$, $\mathbf{p}_1^r\mathbf{p}_1$, $\mathbf{p}_2^r\mathbf{p}_2$, $\mathbf{q}'^r\mathbf{q}'$ and $\overline{\mathbf{q}}^r\overline{\mathbf{q}}$, but somehow this looks like an artificial solution. It seems to me that big word classes, such as the class of adjectives and the class of adverbs, should receive single basic types.

On the other hand, frequently occurring words performing different syntactic functions may have many different compound types. For example, the accusative *whom* has been assigned the following types in this article:

$\mathbf{q}'\mathbf{o}^{\ell\ell}\mathbf{q}^\ell$ for direct questions in Section 6,

$\overline{\mathbf{q}}\mathbf{o}^{\ell\ell}\mathbf{s}^\ell$ and $\overline{\mathbf{q}}\mathbf{o}^{\ell\ell}\overline{\mathbf{i}}^\ell$ for indirect questions in Section 8,

$\mathbf{r}\mathbf{o}^{\ell\ell}\mathbf{s}^\ell$ for relative clauses in Section 9

Moreover, the dative *whom* has been assigned the types:

$\mathbf{q}'\omega^{\ell\ell}\mathbf{q}^\ell$ for direct questions,

$\overline{\mathbf{q}}\omega^{\ell\ell}\mathbf{s}^\ell$ for indirect questions,

$\mathbf{r}\omega^{\ell\ell}\mathbf{s}^\ell$ for relative clauses,

with judicious insertion of $x = \mathbf{o}$, $\mathbf{s}$ or $\overline{\mathbf{s}}$, in Section 12.

The form *is* of the auxiliary verb *be* also has many types:

$\pi_3^r\mathbf{s}_1\mathbf{p}_1^\ell$ for the progressive auxiliary in Section 3,

$\mathbf{q}_1\mathbf{p}_1^\ell\pi_3^\ell$ for questions in Section 6,

$\pi_3^r\mathbf{s}_1\mathbf{o}^{\ell\ell}\mathbf{p}_2^\ell$ for the passive auxiliary in Section 3,

$\mathbf{q}_1\mathbf{p}_2^\ell\mathbf{o}^{\ell\ell}\pi_3^\ell$ for questions in Section 6,

$\pi_3^r\mathbf{s}_1\mathbf{a}^\ell$ for the copula in Section 10,

$\pi_3^r\mathbf{s}_1\omega^{\ell\ell}\mathbf{p}_2^\ell$ for the passive auxiliary in Section 12.

I have not made a complete study of island constraints (see e.g. McCawley, 1988), but I suspect that their purpose is to prevent confusion and overburdening of the short term memory. For the hearer to analyze a sentence may require a number of parallel calculations, some of which lead into blind alleys. It seems that to ask her to resist both tempting contractions in $x^{\ell\ell}x^\ell x$ is asking too much.

## 14. Appendix

For the linguist not familiar with the mathematical terminology employed here, this appendix will present a number of basic definitions and it will summarize some algebraic results of Lambek (1999) without going into the proofs which may be found there.

A (partially) *ordered set* is a set equipped with a binary relation $\leqslant$ which is reflexive, transitive and antisymmetric.

A *monoid* is a set equipped with a unity element 1 and a binary operation, here denoted by juxtaposition such that:

$$a1 = a = 1a, \quad (ab)c = a(bc).$$

An *ordered monoid* is both an ordered set and a monoid and satisfies the condition:

if $a \leqslant b$ and $c \leqslant d$ then $ac \leqslant bd$,

or, equivalently:

if $a \leqslant b$ then $cad \leqslant cbd$.

A *pregroup* is an ordered monoid in which each element $a$ has a *left adjoint* $a^\ell$ such that:

$$a^\ell a \leqslant 1 \leqslant aa^\ell$$

and a *right adjoint* $a^r$ such that:

$$aa^r \leqslant 1 \leqslant a^r a.$$

Consequences of this definition are the following identities:

$$\begin{array}{l} 1^\ell = 1,\ a^{r\ell} = a,\ (ab)^\ell = b^\ell a^\ell,\ aa^\ell a = a,\ a^\ell aa^\ell = a^\ell; \\ 1^r = 1,\ a^{\ell r} = a,\ (ab)^r = b^r a^r,\ aa^r a = a,\ a^r aa^r = a^r \end{array}$$

and the following implication:

if $a \leqslant b$ then $b^\ell \leqslant a^\ell$ and $b^r \leqslant a^r$.

It should be noted, however, that in general:

$$a^r \neq a^\ell, \quad a^{\ell\ell} \neq a, \quad a^{rr} \neq a.$$

Given an ordered set $A$, we may construct the *free* pregroup $F(A)$ generated by $A$. If we write:

$$\cdots, a^{(-2)}, a^{(-1)}, a^{(0)}, a^{(1)}, a^{(2)}, \cdots$$

for:

$$\cdots, a^{\ell\ell}, a^\ell, a, a^r, a^{rr}, \cdots,$$

a typical element of $F(A)$ is of the form:

$$\alpha = a_1^{(n_1)} a_2^{(n_2)} \cdots a_k^{(n_k)},$$

(modulo the equality induced by the partial order), where the $a_i \in A$ and the $n_i \in \mathbb{Z}$. We put:

$$\alpha^\ell = a_k^{(n_k-1)} \cdots a_2^{(n_2-1)} a_1^{(n_1-1)},$$
$$\alpha^r = a_k^{(n_k+1)} \cdots a_2^{(n_2+1)} a_1^{(n_1+1)},$$

and write $\alpha \leqslant \beta$ if $\beta$ is obtained from $\alpha$ by a sequence of steps:

$$\begin{array}{lll} \text{contractions} & \gamma a^{(n)} a^{(n+1)} \delta \leqslant \gamma\delta, & \\ \text{expansions} & \gamma\delta \leqslant \gamma a^{(n+1)} a^{(n)} \delta, & \\ \text{induced steps} & \gamma a^{(n)} \delta \leqslant \gamma b^{(n)} \delta & \text{if } a \leqslant b \text{ and } n \text{ is even} \\ & & \text{or } b \leqslant a \text{ and } n \text{ is odd.} \end{array}$$

A *generalized contraction* combines a contraction with an induced step; it has the form:

$$\begin{array}{ll} \gamma a^{(n)} b^{(n+1)} \delta \leqslant \gamma\delta & \text{if } a \leqslant b \text{ and } n \text{ is even} \\ & \text{or } b \leqslant a \text{ and } n \text{ is odd.} \end{array}$$

*Generalized expansions* are defined similarly.

If $\alpha \leqslant \beta$ in the free pregroup $F(A)$, it can be shown (Lambek, 1999) that there exist strings $\alpha'$ and $\beta'$ such that:

$$\alpha \leqslant \alpha' \leqslant \beta' \leqslant \beta,$$

where $\alpha \leqslant \alpha'$ by generalized contractions only, $\alpha' \leqslant \beta'$ by induced steps only and $\beta' \leqslant \beta$ by generalized expansions only. In particular, if $\beta = b \in A$, no expansions need be involved.

An ordered monoid is said to be *residuated* if there are two binary operations / (*over*) and \ (*under*) such that:

$$ab \leqslant c \text{ iff } a \leqslant c/b \text{ iff } b \leqslant a\backslash c.$$

A pregroup is residuated, as is seen by taking:

$$a/b = ab^\ell, \quad b\backslash a = b^r a.$$

A residuated monoid is said to have a *dualizing* element 0 if:

$$(0/a)\backslash 0 = a = 0/(a\backslash 0).$$

Abbreviating $0/a = a^\ell$ and $a\backslash 0 = a^r$, one can easily show that:

$$1^\ell = 0 = 1^r, \quad (b^r a^r)^\ell = a + b = (b^\ell a^\ell)^r,$$

where $+$ is thus defined.

A pregroup may be described alternatively as a residuated ordered monoid with a dualizing object in which $a + b = ab$ for all $a, b \in A$. It follows that $0 = 1$ is the dualizing element.

I had used free residuated monoids (= syntactic calculus) for linguistic applications as early as 1958. Quite recently, Claudia Casadio similarly applied free residuated monoids with a dualizing element (= noncommutative linear logic). It was her example that led me to look at pregroups.

The syntactic types forming a residuated monoid are not far removed from the semantic types studied by Haskell Curry. The latter essentially form a semi-Heyting algebra, namely a residuated monoid subject to Gentzen's three structural rules: interchange, contraction and weakening. (Actually, Curry followed many early logicians in applying Occam's razor to eliminate conjunction, but that is a side issue here.) Pregroups unfortunately take us further away from Curry's semantics, but they allow us to handle Chomsky's traces by exploiting the fact that the double "negation" $a^{\ell\ell} \neq a$.

## 15. Postscript

Concerning the technique of postponing all expansions to the end, it has come to my attention that, in the context of the non-associative syntactic calculus, such a technique was first proposed by Buszkowski (1986) and elaborated by Kandulski (1988).

## Acknowledgements

This work was supported by the Natural Sciences and Engineering Research Council of Canada. I am indebted to the following persons for stimulating conversation, critical comments and encouragement: Michael Barr, Michael Brame, Wojciech Buszkowski, Claudia Casadio, François Lamarche and Michael Moortgat.

## References

I see no point in repeating the long list of references of my earlier version (Lambek, 1999), so I will only list those not already listed there.

Buszkowski, W. Generative capacity of nonassociative Lambek calculus, *Bulletin of the Polish Academy of Sciences (Mathematics)*, 34: 507–516, 1986.

Kandulski, M. The equivalence of nonassociative Lambek categorial grammars and context-free grammars, *Zeitschrift für mathematische Logik and Grundlagen der Mathematik*, 34: 41–52, 1988.

Lambek, J. Type grammar revisited. In A. Lecomte, F. Lamarche and G. Perrier, editors, *Logical Aspects of Computational Linguistics*, 1–27, Springer LNAI 1582, 1999.

McCawley, J.D. *The Syntactic Phenomena of English*, vol. 2, The University of Chicago Press, Chicago, 1988.