ON THE CALCULUS OF SYNTACTIC TYPES[1]

BY

JOACHIM LAMBEK

In classical physics it was possible to decide whether an equation was grammatically correct by comparing the "dimensions" of the two sides of the equation. These dimensions formed an abelian group with three generators L, M and T, admitting fractional exponents.[2]

One may ask whether it is similarly possible to assign "grammatical types" to the words of English in such a way that the grammatical correctness of a sentence can be determined by a computation on these types. As long as "John loves Jane" fails to imply "Jane loves John" one cannot expect these types to form an abelian group. Probably they should not form a group at all.

Some time ago [3] I suggested a group-like mathematical system, which I called the "syntactic calculus". For reasons that will appear later, it would have been better to call it the "associative syntactic calculus". My method was closely related to an earlier syntactic method by Bar-Hillel, which in turn goes back to the "semantic types" of Ajdukiewicz, Leśniewski and ultimately Husserl. Independent type theories were also developed by Church and Curry,[3] who calls his types "functional characters".

My proposal was briefly this: We start by assigning certain primitive types to words and strings of words. Thus all well-formed statements have type s, all names have type n, all intransitive infinitives have type i, etc. From these primitive types compound types are built up by three formal operations: multiplication, left division and right division. The following examples will illustrate the general method. We write $X \to x$ to indicate that the string X has type x.

$$\begin{aligned} \textit{John} &\to n \\ \textit{John works} &\to s \\ \textit{works} &\to n\backslash s \\ \textit{must work} &\to n\backslash s \\ \textit{work} &\to i \\ \textit{must} &\to (n\backslash s)/i \\ \textit{John work} &\to ni \end{aligned}$$

Given the method of assigning compound types implicit in the foregoing examples, one can prove that the following rules are universally valid [see 3, § 7]:

(1) $$x \to x$$

[1] This paper was written while the author held a grant from the Canada Council.

[2] In the language of modern algebra, this group is a three-dimensional vector-space over the field of rational numbers.

[3] See [3] for detailed references.

(2) $(xy)z \to x(yz)$ (2′) $x(yz) \to (xy)z$

(3) $\dfrac{xy \to z}{x \to z/y}$ (3′) $\dfrac{xy \to z}{y \to x\backslash z}$

(4) $\dfrac{x \to z/y}{xy \to z}$ (4′) $\dfrac{y \to x\backslash z}{xy \to z}$

(5) $\dfrac{x \to y \quad y \to z}{x \to z}$.

Here $x \to y$ means that all strings of type x also have type y. The horizontal line is meant to indicate that the "conclusion" below it follows from the "premise" or "premises" above it.

The free algebraic system or deductive system with axiom schemes (1), (2) and (2′) and rules of inference (3) to (5) may be called the associative calculus of types. Surprisingly, this system has no obvious decision procedure: While the provable formula $(x/y)y \to x$ suggests a method of simplifying expressions by decreasing their length, the provable formulas $x \to (xy)/y$ and $x \to y/(x\backslash y)$ contradict this. Nonetheless, it turned out [3] that a method first proposed by Gentzen for the intuitionistic propositional calculus also works here.[4]

Let us turn to the linguistic applications of the associative syntactic calculus.[5] It soon became apparent that, unless elaborate precautions were taken, the most natural assignments of types to English words tempted one to admit many pseudo-sentences as grammatical, e.g.

(*) *John is poor sad. John likes poor him. Who works and John rests?*

Now I have been unable to find any agreement among modern linguists as to what constitutes a grammatical sentence. At one extreme there are those who call every utterance a sentence, that is any string of words ever mouthed by poet or peasant. At the other extreme there are those who would declare cannibalism ungrammatical on the grounds that *man* does not belong to the class of food-nouns.

My own standard of grammaticalness will sound equally absurd. I have regarded as grammatical

John works today today, John works yesterday today,

simply because I was unable to state a simple procedure which would rule out these expressions without at the same time ruling out

John works today at lunch, John works at lunch today,

both of which seem acceptable to me.

However the pseudo-sentences (*) could be systematically ascribed to one cause, namely the fact that types had been assigned to unstructured strings of words. Suppose we assign types not to strings, but to *phrases*, that is bracketed strings of words or perhaps morphemes. Better still, let us avoid

[4] In fact, it works more neatly here, since Gentzen's "structural" rules completely disappear.

[5] See Appendix II for other applications in mathematics.

the difficult question of identifying the *atomic* phrases and define as follows:[6]

All atomic phrases are phrases.

If A and B are phrases so is (AB).

Types are introduced by a similar recursive definition:[7]

All primitive types are types.

If x and y are types so are (xy), (x/y) and $(x\backslash y)$.

It is understood that types are assigned to phrases in accordance with the following rules:

If A has type a and B has type b then (AB) has type (ab).

If (AB) has type c for all B of type b then A has type (c/b).

If (AB) has type c for all A of type a then B has type $(a\backslash c)$.

Again the types give rise to a formal calculus. Rules (1), (3), (3′), (4), (4′) and (5) are still valid, but the associativity rules (2) and (2′) fail to hold. In particular, the following consequences of the associative law must be dropped:

$$(x\backslash y)/z \rightleftarrows x\backslash(y/z)\,, \quad (x/y)/z \rightleftarrows x/(zy)\,, \quad (x/y)(y/z) \to x/z\,.$$

The last of these goes back to Ajdukiewicz.

A careful investigation shows that the decision procedure for the associative calculus can be adapted to the non-associative calculus. A sketch of the modified decision procedure will be found in Appendix I.

Previous papers have elaborated the claim that the syntactic calculus can be used to transfer many grammatical rules from the grammar to the dictionary. Let me illustrate this by the sentence "John must work". A phrase-structure analysis of this sentence looks like this:

$$\underbrace{\underbrace{\textit{John}}_{n}\ \underbrace{\underbrace{\textit{must}}_{m}\ \underbrace{\textit{work}}_{i}}_{v}}_{s}$$

This analysis presumes that we are given a dictionary with the type assignments

$$\textit{John} \to n\,, \quad \textit{must} \to m\,, \quad \textit{work} \to i$$

and a list of grammatical rules containing

$$mi \to v\,, \quad nv \to s\,.$$

Using the syntactic calculus, we can re-write the last two rules thus:

$$m \to v/i\,, \quad v \to n\backslash s\,.$$

These give rise to the single rule

$$m \to (n\backslash s)/i\,.$$

We may now revise the dictionary thus:

[6] Juxtaposition denotes concatenation and brackets denote themselves. As usual with recursive definitions it is understood that the class of all phrases is the smallest class satisfying the stated conditions. According to the usual convention, brackets on the outside of a complete phrase may be omitted.

[7] See the preceding footnote. Read "(x/y)" as "x over y" and "$(x\backslash y)$" as "x under y".

$$John \to n\ ,\quad must \to (n\backslash s)/i\ ,\quad work \to i\ .$$

Without any further grammatical rules we may then analyse the same sentence as follows:

$$\underbrace{\underset{n}{John}\ \overbrace{\underset{(n\backslash s)/i}{must}\ \underset{i}{work}}^{n\backslash s}}_{s}$$

Suppose we have replaced all grammatical rules of a language by suitable type assignments in the dictionary. It is then possible to parse a given string of words in a purely mechanical fashion: First we turn the string into a phrase X by bracketing and write under each word one of the types assigned to it by the dictionary. Let x be the compound type of the whole phrase. If $x \to s$ is a provable formula of the syntactic calculus, X is a statement. If $x \to i$ is provable, X may be one kind of request, and so on. This process is repeated for all other bracketings and type assignments. Of course a given string may possess syntactic ambiguity. For example, the string *time flies* may be parsed as a statement or as a request [3, § 5].

Since mechanical parsing of sentences is not only of conceptual but also of practical importance (in connection with mechanical translation of languages), it becomes of interest to know for which languages it is possible to replace all grammatical rules by type assignments in the dictionary. This is certainly the case for the formal languages of mathematical logic, at least for all those that I have looked at [3, § 4]. With a view of investigating this problem for natural languages such as English, we shall look at some phrase structure and transformational grammars. It has been pointed out by several people that these grammars possess a mechanical parsing procedure anyway, as long as one avoids elliptical transformation rules. Nonetheless, it is of some interest to determine the extent and limitations of the present methods.

For our purpose it will be convenient to think of a phrase structure grammar as follows: The dictionary assigns to each atomic phrase a finite number of primitive types. The grammar consists of a finite number of rules of the form $p_i p_j \to p_k$ where the p_i are primitive types.[8]

While it seems unlikely that the elimination of grammatical rules in favour of dictionary entries can be carried out for every phrase structure grammar in this sense (without making the dictionary infinite), this can be done in many examples (in fact all that I have tried).

Take for instance Chomsky's language L_2 [2, p. 151]. The sentences of this language are all strings of the form XX^*, where X is a string of A's and H's and X^* is the mirror image of X. One easily verifies that this language is completely described by the dictionary:

$$A \to a\ ,\quad s/a\ ,\quad (s/a)/s$$
$$H \to h\ ,\quad s/h\ ,\quad (s/h)/s\ .$$

[8] In Chomsky's hierarchy of grammars [2], this is essentially a type 2 grammar. See his discussion of his Theorem 5 on page 150. His arrow is the reverse of ours.

The informal comma separating the different types assigned to one word in the dictionary can be replaced by a formal connective $\cap$. Thus, in the last example, we could write

$$A \to (a \cap (s/a)) \cap ((s/a)/s) ,$$

where the right hand side represents a compound type in an enlarged syntactic calculus. We would have to adjoin two new axioms

$$x \cap y \to x , \quad x \cap y \to y$$

and one new rule of inference

$$\frac{t \to x \quad t \to y}{t \to x \cap y}$$

to our formal system. It would not be difficult to extend our decision procedure to this enlarged system, but we shall desist from doing so.[9]

It may happen that type assignments in a dictionary are in a sense stronger than the explicit rules of a phrase structure grammar. Let me illustrate this by an example.

I have previously [3, § 6] suggested that the pronoun *he* be given the type $s/(n\backslash s)$. This means that *he* when followed by any phrase of type $n\backslash s$ (e.g. *must work*) yields a phrase of type s, that is a statement. In other words, the type assignment

$$he \to s/(n\backslash s)$$

is equivalent to the transformation rule

$$\frac{nX \to s}{he\ X \to s} .$$

Such a rule may indeed be implied by the totality of rules and type assignments of a phrase structure grammar, but it is clearly not one of these rules.[10]

We have here a transformation rule which can be conveniently replaced by a type assignment in the dictionary. A similar example is the rule

$$\frac{nX \to s}{who\ X \to (?)}$$

where (?) is the type of well-formed questions. This is equivalent to the dictionary entry

$$who \to (?)/(n\backslash s) .$$

Again the transformation rule

$$\frac{n(must\ i) \to s}{(must\ n)i \to (?)}$$

[9] In the application to ideal theory mentioned in Appendix II, $A \cap B$ is of course the intersection of the ideals A and B. A dual connective to form the sum of two ideals also plays a role.

[10] It won't do to enter "$he \to n$" into the dictionary, for then the non-sentence "John must like he" would obtain type s.

could be handled by supplementing the original dictionary entry

(1) $$must \rightarrow (n\backslash s)/i$$

by the further assignment

(2) $$must \rightarrow ((?)/i)/n\ .$$

Unfortunately, this won't allow us to derive the sentence "Must he work?". Let $\hat{n} = s/(n\backslash s)$ be the type of *he* considered above; then we really want the type assignment

(3) $$must \rightarrow ((?)/i)/\hat{n}\ .$$

It is not necessary for the dictionary to carry both (2) and (3). (3) alone will do, since

$$((?)/i)/\hat{n} \rightarrow ((?)/i)/n$$

is a derivable formula in the syntactic calculus. Instead of proving this rule, let us parse the sentence "Must John work?" using (3) and the provable formula[11]

$$n \rightarrow \hat{n} = s/(n\backslash s)\ .$$

Thus

$$\begin{array}{lll} (Must & John) & work \\ ((?)/i)/\hat{n} & n & i \\ & \underbrace{\qquad\qquad}_{\hat{n}} & \\ & (?)/i & \\ & \underbrace{\qquad\qquad\qquad}_{(?)} & \end{array}\ .$$

Incidentally, we have here a simple example of a sentence analysis, in which the types need not get "shorter" as we go from top to bottom, i.e., in the direction of the arrow.[12]

Unfortunately there seems to be a limit to this sort of thing. I have not succeeded in handling the interrogative *whom*, without some rather artificial devices.

Consider for instance the sentence

Whom must John call today?

According to Chomsky [1], this is derived from the pseudo-sentence

**Him must John call today,*

which in turn is derived from

Must John call him today?

The easiest way to account for this sequence of transformations by type assignments is to assume that **him* and *whom* consist of two atomic phrases

[11] A proof of the dual formula $x \rightarrow (y/x)\backslash y$ appears in Appendix I.

[12] A similar example was given in [**3**, § 6]. Unfortunately this example was marred by a repeated misprint. In [**3**, p. 163, lines 1 and 4] all occurrences of the type symbol "s" should be replaced by occurrences of the type symbol "n^*".

each, say

$$\begin{array}{l} *him = * + h'm \\ \qquad\quad s/i \quad (i/(i/n))/((?)/i) \end{array}$$

$$\begin{array}{l} whom = w + h'm \\ \qquad\quad (?)/i \quad (i/(i/n))/((?)/i) \,. \end{array}$$

The reader will have no difficulty in verifying that the *whom*-sentence under consideration now receives type (?) as it should.

Some other type-assignments to *whom* will work equally well. Unfortunately they all have one defect in common: Certain non-sentences such as

Whom must John call and resign his job?

also receive type (?). If I am not mistaken, Chomsky can avoid the derivation of this non-sentence from

Must John call him and resign his job?

by stipulating that the transformation which introduces *and* must come after the transformation which introduces *whom*.

If we ignore the order of succession of transformation rules we would make this entry into our dictionary:

$$whom \to ((?)/i) + ((i/(i/n))/((?)/i)) \,.$$

The purpose of the plus sign is to indicate that the place occupied by it may be a phrase boundary.

But even if such clumsy type-assignments are admitted, there are still the "elliptical" transformation rules which cause difficulty, as for example the rule which allows us to drop the relative pronoun *whom* in *the man* (*whom*) *John must call today* $\cdots$. I don't see how such a rule could be incorporated into the dictionary, unless the first dictionary entry was a blank with the type of the relative pronoun *whom* assigned to it. The presence of such a blank in the dictionary would play havoc with our procedure for deciding whether a given string of English words can be bracketed to be a phrase of prescribed type.

I must confess that I am still in two minds about the usefulness of the syntactic calculus for the study of grammars of natural languages. I shall therefore let the above examples speak for themselves and refrain from drawing any conclusion.

Appendix I

We present the non-associative syntactic calculus as a deductive system Σ, beginning with the usual inventory.

Variables: $v_1, v_2, \cdots$.

Terms: All variables are terms.

If x and y are terms so are $(x \cdot y)$, $(x\backslash y)$, (x/y).

We shall make the usual convention for omitting brackets on the outside. Also we shall write xy for $x \cdot y$.

Formulas: $x \to y$, where x and y are terms.

Axiom scheme: $x \to x$.

Rules of inference:

$$\frac{xy \to z}{x \to z/y} \qquad \frac{xy \to z}{y \to x\backslash z}$$

$$\frac{x \to z/y}{xy \to z} \qquad \frac{y \to x\backslash z}{xy \to z}$$

$$\frac{x \to y \quad y \to x}{x \to z}.$$

For the purpose of describing a decision procedure for Σ we introduce a related system Σ_G which contains Σ. The terms of Σ_G will be called G-terms, to distinguish them from the terms of Σ.

G-terms: All terms are G-terms.

If X and Y are G-terms so is (X, Y).

G-formulas:[13] $X \to y$, X a G-term, y a term.

G-rules of inference:

$$\frac{T, y \to x}{T \to x/y} \qquad \frac{y, T \to x}{T \to y\backslash x}$$

$$\frac{T \to y \quad F(x) \to z}{F((x/y), T) \to z} \qquad \frac{T \to y \quad F(x) \to z}{F(T, (y\backslash x)) \to z}$$

$$\frac{P \to x \quad Q \to y}{P, Q \to xy}.$$

Cut-rule:

$$\frac{T \to x \quad F(x) \to y}{F(T) \to y}.$$

Here T, P, Q are arbitrary G-term. $F(X)$ represents a G-term in which X appears as a factor. By this we mean that $F(X) = X$ or $F(X) = G(X, T)$ or $F(X) = G(T, X)$ where $G(X)$ is a shorter G-term containing X as a factor.

With every G-term X there is associated a term x obtained from it by deleting all commas. Thus if X is $((v_1/v_2), v_2)$ then x is $((v_1/v_2)v_2)$. It is not difficult to prove the following:

METATHEOREM 1. *$X \to y$ is provable in Σ_G if and only $x \to y$ is provable in Σ.*

This is an fact an immediate consequence of the lemmas:

LEMMA 1. *If all commas are deleted in any G-rule, including the cut-rule, we obtain a derived rule of inference of Σ.*

LEMMA 2. *Every rule of inference in Σ is a derived rule of inference in Σ_G.*

The proofs of these two lemmas are much like the argument in [3, § 8]. The reader will easily supply the details.

COROLLARY. *$x \to y$ is provable in Σ if and only if it is provable in Σ_G.*

[13] The G-formulas are the non-associative analogues of the so-called "sequents" (*Folgen*) of Gentzen.

Here, for example are proofs of the formula $x \to (y/x)\backslash y$ in Σ and Σ_G:

$$\dfrac{\dfrac{y/x \to y/x}{(y/x)x \to y}}{x \to (y/x)\backslash y} \qquad \dfrac{\dfrac{x \to x \quad y \to y}{y/x,\ x \to y}}{x \to (y/x)\backslash y}\ .$$

By the *degree* of a G-term or G-formula we mean the total number of occurrences of the connectives $\cdot$, / and \ in it. Occurrences of the comma are not counted. Thus $((v_1/v_2)v_2)$ has degree 2, but $((v_1/v_2), v_2)$ has degree 1.

A glance at the G-rules other than the cut-rule shows that each such rule introduces an occurrence of one of the connectives $\cdot$, /, \ into the conclusion and that the degree of the conclusion is larger than the sum of the degrees of the premises. It is now easily seen that we have a decision procedure for Σ_G without cut-rule, namely the method of seeking proofs by working backward from the conclusion.

METATHEOREM 2. *Any G-formula provable with cut-rule is also provable without cut-rule.*

To show this, we define the degree of a cut (instance of the cut-rule) as the sum of the degrees of $F(T)$, x and y. The result then follows from the lemma:

LEMMA 3. *In any cut whose premises have been proved without cut, the conclusion is either identical with one of the premises, or else the cut can be replaced by one or two cuts of smaller degree.*

To prove this we must investigate a large number of cases and subcases as in [3, § 9]. In fact, the argument there goes over to the non-associative situation with minor changes. For example, let us consider Case 6 of [3, § 9]: The last steps in the proofs of both premises introduce the main connective of $x = x'/x''$. It follows that $F(x)$ has the form $G(x, U)$ and we have:

$$\dfrac{\dfrac{T, x'' \to x'}{T \to x'/x''} \qquad \dfrac{U \to x'' \quad G(x') \to y}{G(x'/x'', U) \to y}}{G(T, U) \to y}\ .$$

This derivation may be replaced by

$$\dfrac{U \to x'' \quad \dfrac{T, x'' \to x' \quad G(x') \to y}{G(T, x'') \to y}}{G(T, U) \to y}$$

where both new cuts have smaller degree than the old cut.

COROLLARY 1. *Σ_G has a decision procedure.*

COROLLARY 2. *Σ has a decision procedure.*

This follows from Corollary 1 and Metatheorem 1.

Appendix II

The associative syntactic calculus also has application in mathematics. In multiplicative ideal theory AB, A/B and $B\backslash A$ may be interpreted as the product, right residual quotient and left residual quotient of the ideals A and B respectively. More interesting perhaps is the application to bimodules worked out by G. D. Findlay and the present author in 1956. Here is a necessarily abbreviated report of our work, which has never been published.

The idea is briefly this: Each proof of a formula in the associative syntactic calculus may be used to construct a canonical mapping between functors of bimodules built up from $\otimes$ and Hom. For example the proof of the formula $C/(AB) \to (C/B)/A$ from postulates (1) to (5) gives an explicit construction for the canonical mapping of $\mathrm{Hom}_T(A \otimes_S B, C)$ into $\mathrm{Hom}_S(A, \mathrm{Hom}_T(B, C))$, where ${}_RA_S$, ${}_SB_T$ and ${}_RC_T$ are given bimodules. The decision procedure for the associative syntactic calculus can then be used to find all canonical mappings from one functor into another.

It is convenient to assign a number to each proof in the associative syntactic calculus as follows. We recall that the axioms and rules of inference were labelled 1, 2, 2′, 3, 3′, 4, 4′ and 5. For the present purpose we shall replace the labels 2′, 3′ and 4′ by 20, 30 and 40 respectively.

(i) The proof of an axiom (m), where $m = 1, 2$ or 20, has number m.

(ii) If a formula is derived from a formula whose proof has number n by rule (m), where $m = 3, 30, 4$ or 40, the resulting proof has number mn.

(iii) If a formula is derived by rule (5) from formulas whose proofs have numbers m and n, the resulting proof has number $5mn$.

For example, if t is short for $(x\backslash y)/z$, we have the following derivation

$$\dfrac{\dfrac{\dfrac{(xt)z \to x(tz)}{(xt)z \to y}(3)}{xt \to y/z}(30)}{t \to x\backslash(y/z)}(5) \qquad \dfrac{\dfrac{t \to (x\backslash y)/z}{tz \to x\backslash y}(4)}{x(tz) \to y}(40)$$

The number of this proof is 3035204041.

Given the final conclusion of a proof, the number of the proof almost completely describes the proof, but not quite: Every time rule (5) is used, the so-called middle term y must be specified. This shortcoming disappears when we work with the associative analogue of Σ_G and avoid the cut-rule,

If ${}_RA_S$, ${}_SB_T$ and ${}_RC_T$ are bimodules, R, S and T being associative rings, we write

$$AB = A \otimes_S B\,, \quad C/B = \mathrm{Hom}_T(B, C)\,, \quad A\backslash C = \mathrm{Hom}_R(A, C)\,.$$

AB is an R-T-bimodule, C/B is an R-S-bimodule and $A\backslash C$ is an S-T-bimodule, in a natural fashion.

For expository purposes we shall from now on only consider R-R-bimodules, where R is fixed once and for all. If A and B are such, then so are AB,

A/B and $B\backslash A$. Note that the traditional notation fails to distinguish between A/B and $B\backslash A$. The class of canonical mappings is defined recursively as follows. We assume that $a \in A$, $b \in B$, $c \in C$.

$$\phi_1 : A \to A$$
$$\phi_1 a = a$$

$\phi_2 : (AB)C \to A(BC)$	$\phi_{20} : A(BC) \to (AB)C$
$\phi_2 \Sigma(a \otimes b) \otimes c = \Sigma a \otimes (b \otimes c)$	$\phi_{20} \Sigma a \otimes (b \otimes c) = \Sigma(a \otimes b) \otimes c$
If $\phi_n : AB \to C$	If $\phi_n : AB \to C$
then $\phi_{3n} : A \to C/B$	then $\phi_{30n} : B \to A\backslash C$
$(\phi_{3n} a)b = \phi_n(a \otimes b)$	$a(\phi_{30n} b) = \phi_n(a \otimes b)$
If $\phi_n : A \to C/B$	If $\phi_n : B \to A\backslash C$
then $\phi_{4n} : AB \to C$	then $\phi_{40n} : AB \to C$
$\phi_{4n} \Sigma a \otimes b = \Sigma(\phi_n a)b$	$\phi_{40n} \Sigma a \otimes b = \Sigma a(\phi_{40n} b)$

$$\text{If } \phi_m : A \to B \text{ and } \phi_n : B \to C$$
$$\text{then } \phi_{5mn} : A \to C$$
$$\phi_{5mn} a = \phi_n(\phi_m a) .$$

This notation is not entirely unambiguous, e.g. the identity map of B into itself is also denoted by ϕ_1, but it will do.

It is now apparent that if n is the number of a proof of a formula $x \to y$ in the associative syntactic calculus, there is a corresponding canonical mapping $\phi_n : X \to Y$. One example should make this clear: We found that the formula $(x\backslash y)/z \to x\backslash(y/z)$ has a proof with number $n = 3035204041$. Consequently we should have a canonical mapping

$$\phi_n : (A\backslash B)/C \to A\backslash(B/C) .$$

Let $f \in (A\backslash B)/C$; then we compute

$$\begin{aligned}
(a(\phi_{3035204041} f))c &= (\phi_{35204041}(a \otimes f))c \\
&= \phi_{5204041}((a \otimes f) \otimes c) \\
&= \phi_{4041}(\phi_{20}((a \otimes f) \otimes c)) \\
&= \phi_{4041}(a \otimes (f \otimes c)) \\
&= a(\phi_{41}(f \otimes c)) \\
&= a((\phi_1 f)c) \\
&= a(fc) .
\end{aligned}$$

To see the meaning of this, let us return to the classical notation. Assume we have bimodules ${}_RA_S$, ${}_RB_T$ and ${}_UC_T$. (It is important that the subscripts match as indicated.) We have shown that there is a canonical mapping

$$\phi_n : \mathrm{Hom}_T(C, \mathrm{Hom}_R(A, B)) \to \mathrm{Hom}_R(A, \mathrm{Hom}_T(C, B))$$

given by

$$(a(\phi_n f))c = a(fc) .$$

This is of course what one would have expected.

The decision procedure for the associative syntactic calculus given in [3] can be used to find all canonical mappings (according to our recursive definition of "canonical") from one functor into another. However, this decision procedure operates not on formulas $x \to y$ but on "sequents" $x_1, x_2, \cdots, x_n \to y$. These are the associative analogues of the G-formulas of Appendix I. The suggested method for finding canonical mappings therefore does not deal with mappings $\phi : A \to B$ directly but with multilinear mappings $\Phi : A_1 \times A_2 \times \cdots \times A_n \to B$. It has already been observed by Bourbaki[14] that linear mappings of the kind we are interested in are best defined with the help of multilinear mappings. The details are too technical to be given here, but an example may help the interested reader to reconstruct the general method.

A proof of the formula $(a\backslash b)/c \to a\backslash(b/c)$ using the decision procedure of [3] looks like this:

$$\dfrac{\dfrac{\dfrac{\dfrac{a \to a \quad b \to b}{a, a\backslash b \to b} \qquad c \to c}{a, (a\backslash b)/c, c \to b}}{a, (a\backslash b)/c \to b/c}}{(a\backslash b)/c \to a\backslash(b/c)}$$

We interpret this proof in terms of mappings:

1. From the identity mappings $A \to A$ and $B \to B$ one derives the multilinear mapping

$$\Phi : A \times \mathrm{Hom}_R(A, B) \to B$$

defined by

$$\Phi(a, g) = ag \ ,$$

where $a \in A$ and $g \in \mathrm{Hom}_R(A, B)$.

2. From this and the identity mapping $C \to C$ one derives the multilinear mapping

$$\Phi' : A \times \mathrm{Hom}_T(C, \mathrm{Hom}_R(A, B)) \times C \to B \ ,$$

where

$$\Phi'(a, f, c) = \Phi(a, fc) = a(fc) \ ,$$

for $c \in C$, $f \in \mathrm{Hom}_T(C, \mathrm{Hom}_R(A, B))$.

3. From this one derives

$$\Phi'' : A \times \mathrm{Hom}_T(C, \mathrm{Hom}_R(A, B)) \to \mathrm{Hom}_T(C, B) \ ,$$

defined by

$$(\Phi''(a, f))c = \Phi'(a, f, c) = a(fc) \ .$$

4. From this one derives the mapping

$$\phi : \mathrm{Hom}_T(C, \mathrm{Hom}_R(A, B)) \to \mathrm{Hom}_R(A, \mathrm{Hom}_T(C, B))$$

defined by

$$a((\phi f)c) = (\Phi''(a, f))c = a(fc) \ .$$

ϕ is the canonical mapping we were expecting.

[14] See N. Bourbaki, *Algèbre multilinéaire*, Paris, Hermann, 1948.

REFERENCES[15]

1. N. Chomsky, *Syntactic structures*, The Hague, Mouton and Company, 1957.
2. ———, *On certain formal properties of grammars*, Information and Control vol. 2 (1959) pp. 137-167.
3. J. Lambek, *The mathematics of sentence structure*, Amer. Math. Monthly vol. 65 (1958) pp. 154-169.
4. ———, *Contributions to a mathematical analysis of the English verb-phrase*, J. Canad. Ling. Assoc. vol. 5 (1959) pp. 83-89.

INSTITUTE FOR ADVANCED STUDY,
PRINCETON, NEW JERSEY

[15] Further references may be found in [3].