Type theory and the informal language of mathematics

Aarne Ranta
Department of Philosophy, P.O.Box 24,
00014 University of Helsinki, Finland.

September 1993.

In the first comprehensive formalization of mathematics, the *Begriffsschrift* (1879), Frege gave up the structure of informal language, in order to reveal the structure of mathematical thought itself. Attempts to apply Frege's formalism to informal discourse outside mathematics followed in this century, e.g. by Russell, Carnap, Quine, and Davidson. In this tradition, the application of logical formalism to informal language is an exercise of skill, rather than an algorithmic procedure, precisely because the linguistic structure is different from the logical structure.

It was Chomsky (1957) who started the study of natural language itself as a formal system, inductively defined by the clauses of a generative grammar. But the structure he gave to his fragment of English was quite different from the structure of a logical formalism.

Finally, Montague (1970) unified the enterprises of Frege and Chomsky in an attempt to give a systematic logical formalization to a fragment of English. His grammar applies to a piece of informal discourse outside mathematics. But as modern logic, even in the form employed by Montague, stems from Frege, who designed it for mathematics, a grammar like Montague's should be applicable, if at all, to the language of mathematics.

Following roughly the format of Montague grammar, I have been working within the constructive type theory of Martin-Löf. (See Martin-Löf 1984 and Nordström & al. 1990 for the type theory, and Ranta 1991 and 1993 for the grammar.) Type theory has proved to be structurally closer to natural language than predicate calculus at least at the following points.

First, type theory makes a distinction between substantival and adjectival terms, e.g. between *number* and *prime*. These are formalized as N : set $P : (N)\mathrm{prop}$, respectively, whereas in predicate calculus they are both formalized as one-place propositional functions.

Second, type theory has quantifier phrases, like *every number*— in type theory,

$$\Pi(N) : ((N)\mathrm{prop})\mathrm{prop}$$

and *every prime number*— in type theory,

$$\Pi(\Sigma(N,P)) : ((\Sigma(N,P))\mathrm{prop})\mathrm{prop}.$$

Predicate calculus dissolves these quantifier phrases, because it does not have expressions corresponding to them.

Third, type theory has progressive connectives, i.e. a conjunction and an implication of the type

$$(X : \mathrm{prop})((X)\mathrm{prop})\mathrm{prop}.$$

Such connectives are abundant in informal language, in sentences like *if this equation has a root it is negative.*

To express this in type theory, first look at the implicans *this equation has a root*. It is an existential proposition, of the form

$$\Sigma(R, E).$$

To form the implicandum, use the propositional function x *is negative* defined for $x : R$, i.e.

$$N : (R)\text{prop}$$

in the context of a proof of the implicans,

$$z : \Sigma(R, E).$$

Left projection gives $p(z) : R$, whence

$$N(p(z)) : \text{prop}$$

by application,

$$(z)N(p(z)) : (\Sigma(R, E))\text{prop}$$

by abstraction, and finally

$$\Pi(\Sigma(R, E), (z)N(p(z))) : \text{prop}$$

to express the implication. Predicate calculus, which only has connectives of type

$$(\text{prop})(\text{prop})\text{prop},$$

cannot compose the sentence from the implicans and the implicandum, but must use something like

$$(\forall x)(R(x) \& E(x) \supset N(x)),$$

which does not have constituents formalizing the two subclauses of the sentence in question. This lack of compositionality has been first noted in the discussion of so-called donkey sentences, e.g.

if John owns a donkey he beats it

which has the same form as our mathematical example. Some linguists think such sentences are artificially complicated, but they are certainly abundant in the informal language of mathematics.

1 Formalization and sugaring

There are two directions of grammatical investigation. One can ask:

> How is this sentence / mode of expression / fragment of discourse represented in the formalism?

Questions put in this way, starting with what is given in the informal language, are questions of *formalization*. But one can also start with what is given in the formalism and ask:

> How is this proposition / logical constant / theory expressed in natural language?

These questions will be called questions of *sugaring*.

A special case of formalization is *parsing*: given a string of words belonging to an inductively defined set of such strings, find the grammatical structure. This notion of parsing is of course secondary to the notion of *generation*, the inductive definition of the set of strings. Furthermore, as we can think of generation as the composition of (1) the definition of the formalism and (2) the sugaring of the formalism, we see that parsing is secondary to sugaring in the conceptual order.

2 Basic expressions of geometry

In what follows we shall, even if not define a complete sugaring algorithm, look at mathematics expressed in type theory from the sugaring point of view. We shall apply the sugaring principles of Ranta 1991, 1993, originally presented for everyday discourse (like the donkey sentences), to the language of axiomatic geometry such as in Hilbert 1899 and, within type theory, von Plato 1993.

Start with simple set terms,

$$\mathrm{point} : \mathrm{set},$$

$$\mathrm{line} : \mathrm{set},$$

$$\mathrm{plane} : \mathrm{set}.$$

The sugaring of simple set expressions into common nouns is simple (in the absence of the singular and plural number of nouns),

$$\mathrm{point} \quad \rhd \; \mathit{point},$$

$$\mathrm{line} \quad \rhd \; \mathit{line},$$

$$\mathrm{plane} \quad \rhd \; \mathit{plane}.$$

We use the form $F \quad \rhd \; E$ to express the relation

$$F \text{ can be sugared into } E.$$

Thus it is not an expression for a clause in a deterministic sugaring algorithm.

Then some propositional functions sugared into verbs and adjectives.

$$\text{lie\_PL} : (\mathrm{point})(\mathrm{line})\mathrm{prop},$$
$$\text{lie\_PL}(a, b) \quad \rhd \quad a \textit{ lies on } b,$$

$$\text{lie\_PPl} : (\mathrm{point})(\mathrm{plane})\mathrm{prop},$$
$$\text{lie\_PPl}(a, b) \quad \rhd \quad a \textit{ lies in } b,$$

$$\text{lie\_LPl} : (\mathrm{line})(\mathrm{plane})\mathrm{prop},$$
$$\text{lie\_LPl}(a, b) \quad \rhd \quad a \textit{ lies in } b,$$

$$\mathrm{parallel} : (\mathrm{line})(\mathrm{line})\mathrm{prop},$$
$$\mathrm{parallel}(a, b) \quad \rhd \quad a \textit{ is parallel to } b,$$

$$\mathrm{equal} : (A : \mathrm{set})(A)(A)\mathrm{prop},$$
$$\mathrm{equal}(A, a, b) \quad \rhd \quad a \textit{ is equal to } b.$$

Observe how sugaring overloads the English expressions *lies in* and *equal*. The adjective *equal* is fully polymorphic, the verb *lies in* has two uses. The adjective *parallel* and the verb *lies on* are, in this fragment at least, uniquely typed.

3 Logical constants

There are quantifier words like

$$\text{every} : (X : \text{set})((X)\text{prop})\text{prop},$$

$$\text{Indef} : (X : \text{set})((X)\text{prop})\text{prop},$$

$$\text{some} : (X : \text{set})((X)\text{prop})\text{prop}.$$

A quantified proposition is sugared by replacing the bound variable by a quantifier phrase,

$$\text{every}(A,(x)B) \;\triangleright\; B[\mathit{every}\ A/x],$$

$$\text{Indef}(A,(x)B) \;\triangleright\; B[\text{INDART}(A)\ A/x],$$

$$\text{some}(A,(x)B) \;\triangleright\; B[\mathit{some}\ A/x].$$

INDART(A) is the indefinite article corresponding to the sugaring of A, either *a* or *an*. Observe that if the number of occurrences of x in B is other than one, we may get odd results like

$$\text{every}(\text{line},(x)\text{parallel}(x,x))$$
$$\triangleright\ \textit{every line is parallel to every line.}$$

The uniqueness of replacements can be attained e.g. by using pronouns (see Section 6). It is one of the central problems of the logical formalization of natural language, stemming from the apparently quite different modes of expression of quantification in them. Following Frege (1879, § 9), we shall use the word *main argument* for the occurrence of x to be replaced by the quantifier phrase. (See Ranta 1991, Section 5, for a definition of the main argument.)

Another difficulty with the replacement procedure in the sugaring of quantifiers is that the relative scopes of the quantifiers get lost. The rules give e.g.

$$\left.\begin{array}{l}\text{some}(\text{point},(x)\text{every}(\text{line},(y)\text{lie\_PL}(x,y)))\\ \text{every}(\text{line},(y)\text{some}(\text{point},(x)\text{lie\_PL}(x,y)))\end{array}\right\} \;\triangleright\; \textit{some point lies on every line},$$

and such sentences are indeed considered ambiguous in Montague grammar. But it seems that the mathematician would without hesitation interpret the sentence as the first proposition, although it is a plainly false proposition. He would follow the principle according to which the scopes of the quantifiers get narrower from left to right. (On this rule of precedence, as well as some other ones, cf. Ranta 1993, Chapters 3 and 9.)

As for connectives, we introduce two progressive ones and one that is not progressive.

$$\text{if} : (X : \text{prop})((X)\text{prop})\text{prop},$$
$$\text{if}(A,(x)B) \;\triangleright\; \mathit{if}\,A, B[\emptyset/x],$$

$$\text{and} : (X : \text{prop})((X)\text{prop})\text{prop},$$
$$\text{and}(A,(x)B) \;\triangleright\; A\ \mathit{and}\ B[\emptyset/x],$$

$$\text{or} : (\text{prop})(\text{prop})\text{prop},$$
$$\text{or}(A,B) \;\triangleright\; A\ \mathit{or}\ B,$$

where $\emptyset$ is the ellipsis, the empty morph.

Connective and quantifier words are not type-theoretical primitives, but have the definitions

$$\text{every} = \Pi : (X : \text{set})((X)\text{prop})\text{prop},$$

$$\text{Indef} = \Sigma : (X : \text{set})((X)\text{prop})\text{prop},$$

$$\text{some} = \Sigma : (X : \text{set})((X)\text{prop})\text{prop},$$

$$\text{if} = \Pi : (X : \text{prop})((X)\text{prop})\text{prop},$$

$$\text{and} = \Sigma : (X : \text{prop})((X)\text{prop})\text{prop},$$

$$\text{or} = + : (\text{prop})(\text{prop})\text{prop}.$$

The main difference between quantifiers and progressive connectives is in the sugaring of the first argument: for quantifiers, it is a common noun, and for connectives, a sentence. But there is another difference, which has to do with the expressive capacities of the two modes of expression in English. We noted before that the sugaring of a quantified proposition $Q(A, (x)B)$ requires there to be precisely one main argument occurrence of x in B. For connectives, there is no such restriction. Thus for instance the vacuous quantification

$$\Pi(\text{equal}(N, 0, 1), (x)\text{equal}(N, 1, 10000))$$

gives, by the sugaring rule for every, the falsity *one is equal to ten thousand*, and it is the rule for if that gives the right true proposition,

if zero is equal to one, one is equal to ten thousand.

Thus connectives provide a more widely applicable means of expressing propositions than quantifiers.

4 Objects and expressions

Sugaring is not a function on type-theoretical objects, such as propositions, but on expressions for those objects. For by the extensionality of functions, a proposition would be then sugared in the same way, in whatever way expressed. But we certainly want to sugar every(A, B) differently from if(A, B), although they are both equal to $\Pi(A, B)$. Even more clearly, if we introduce an abbreviatory expression by explicit definition, we want to sugar the definiendum differently from the much longer definiens. Consider, for instance,

$$\text{triangle} = \Sigma(\text{line}, (x)\Sigma(\text{point}, (y)\text{outside\_PEl}(y, \text{extended}(x)))) : \text{set},$$

where outside_PEl(a, b) says that the point a lies outside the extended line b, and extended(a) is the infinite extension of the finite line a.

In general, we want to introduce so many definitional variants of type-theoretical expressions that there is a one-to-one correspondence between English and type-theoretical expressions.

The propositions as types principle is, analogously, assumed for the objects of type theory only. We want the type prop to correspond to sentences, and the type set to common nouns. For type-theoretical objects. we have

$$\text{prop} = \text{set} : \text{type},$$

but for expressions, this equation is not effective, whereas we assume the transformation

$$\text{there} = (X)X : (\text{set})\text{prop}$$

sugared

$$\text{there}(A) \;\; \triangleright \;\; \textit{there is}\ \text{INDART}(A)\ A.$$

It may happen that some expression cannot be sugared, e.g. if it contains a quantifier with no or multiple main arguments. In such a case, the sugaring of the proposition expressed must proceed by finding a definitional variant that can be sugared.

5 Relative pronouns

To form complex set terms, we can use relative pronouns, e.g.

$$\text{that} = \Sigma : (X : \text{set})((X)\text{prop})\text{set},$$
$$\text{that}(A, (x)B) \;\; \triangleright \;\; A\ \textit{that}\ B[\emptyset/x],$$

$$\text{such\_that} = \Sigma : (X : \text{set})((X)\text{prop})\text{set},$$
$$\text{such\_that}(A, (x)B) \;\; \triangleright \;\; A\ \textit{such that}\ B[\emptyset/x].$$

These definitions accord with Martin-Löf's (1984) explanation of *such that* as forming a set of elements of the basic set paired with witnessing information. This treatment is necessary for a compositional formalization of quantifier phrases whose domains are given by using relative clauses, and reference is also made to the witnessing information; cf. Section 7 below. But at the same time, we will have to sugar e.g.

$$\text{every}(\text{that}(A, B), (x)C) \;\; \triangleright \;\; C[\text{that}(A, B)/p(x)],$$

i.e. not replace x but $p(x)$. This can be accomplished by the general rule

$$p(x) \;\; \triangleright \;\; x.$$

The slight unnaturalness of the solution is, so it seems to me, one instance of the problems we have in formalizing separated subsets by Σ and trying to get rid of the extra information in some cases, while having to keep it in some other cases.

The difference between that and such_that is analogous to the difference between quantifiers and connectives: that requires there to be exactly one main argument in the relative clause, but such_that does not. such_that is thus more widely applicable than that.

6 Anaphoric expressions

Pronouns are introduced to our fragment of English by the rules

$$\text{Pron} = (X)(x)x : (X : \text{set})(X)X,$$
$$\text{Pron}(A, a) \;\; \triangleright \;\; \text{PRO}(A).$$

In mathematical language, we do not need *he* or *she*, so $\text{PRO}(A)$ is always *it*, and we could as well have

$$\text{it} = (X)(x)x : (X : \text{set})(X)X,$$
$$\text{it}(A, a) \;\; \triangleright \;\; \textit{it}.$$

Observe that our definition of *it* as the polymorphic identity mapping whose argument is sugared away is very similar to the *it* of ML. The main difference is the *interpretation rule* stating in what situations *it* may replace a singular term. In ML, *it* always refers to the value of the latest value declaration. But this rule is too simple for the informal language of mathematics, where *it* can have different—yet definite—interpretations in one and the same clause, e.g.

if the function f has a maximum, it reaches it at least twice.

Our main rule regulating the use of pronouns (and other anaphoric expressions; see below) is that

> the interpretation of an anaphoric expression is an object of appropriate type given in the context in which the expression is used.

Context here is, in the technical sense of type theory, a list of declarations of variables assumed when the expression is formed. For instance, the proposition B in $\Pi(A,(x)B)$ is formed in the context $x : A$. To these variables we add the constant singular terms used in the same sentence. Moreover, we close the "universe of discourse" based on the context under selector operations (cf. Ranta 1993, Chapter 4, for more details).

An interpretation $a : A$ of a pronoun E in the English expression $\perp\perp\perp E \perp\perp\perp$ must thus fulfil the following two conditions.

$\mathrm{Pron}(A,a) \;\triangleright\; E,$

there is a propositional function $B(x) : \mathrm{prop}\ (x : A)$
such that $B(a) \;\triangleright\; \perp\perp\perp a \perp\perp\perp.$

As the only pronoun in the mathematical fragment is *it*, the first condition is always satisfied. If there are many objects given in context, it is the second condition that saves the *uniqueness* of reference, expressed by the principle that

> the interpretation of an anaphoric expression must be unique in the context in which it is used.

There are other anaphoric expressions besides pronouns, more specific in the sense that they do not suppress all information about the object referred to. A definite noun phrase formed by the definite article the preserves the type of the object. A modified definite phrase formed by Mod makes explicit some more information given about the object in the context.

$\mathrm{the} = (X)(x)x : (X : \mathrm{set})(X)X,$
$\mathrm{the}(A,a) \;\triangleright\; \mathit{the} A.$

$\mathrm{Mod} = (X)(Y)(x)(y)x : (X : \mathrm{set})(Y : (X)\mathrm{prop})(x : X)(Y(x))X,$
$\mathrm{Mod}(A,B,a,b) \;\triangleright\; \mathit{the}\ \ A\ \ \mathit{that}\ \ B[\emptyset/x].$

7 Example: the axiom of parallels

To see how the sugaring principles work, take as an example the axiom of parallels in the formulation (written in lower level notation for readability)

$$(\Pi z : (\Sigma x : \mathrm{point})(\Sigma y : \mathrm{line})\mathrm{outside\_PL}(x,y))\mathrm{DAP}(p(z),p(q(z))),$$

where

$$\text{outside\_PL}(a, b) \;\; \triangleright \;\; \textit{a lies outside b},$$

$$\text{DAP}(a, b) = (\exists! x : \text{line})(\text{lie\_PL}(a, x) \& \text{parallel}(x, b)) : \text{prop for } a : \text{point}, b : \text{line},$$
$$\text{DAP}(a, b) \;\; \triangleright \;\; \textit{a determines a parallel to b.}$$

To find the different possibilities to express the axiom of parallels in English provided by our grammar, recall the definitional variants

every and if for Π,

Indef, some, and, that, and such_that for Σ,

A and there(A) for A : set,

Pron(A,a) and the(A,a) for a : A.

Start sugaring from the outermost form of the proposition. First choose the definitional variant every for Π. Then you must sugar the domain of quantification

$$(\Sigma x : \text{point})(\Sigma y : \text{line})\text{outside\_PL}(x, y)$$

into a set expression. The only choice for the first Σ is a relative pronoun, that or such_that. The domain of this Σ must be sugared into the common noun *point*. The remaining part must be found a sentence-like expression. All ways of sugaring Σ are usable: if you choose the relative pronoun, just apply there. The domain of quantification of the axiom of parallels thus has the following sugarings, among others.

point that lies outside a line,

point that lies outside some line,

point such that there is a line and it lies outside it.

The third sugaring is a little strange, because the interpretation of the two occurrences of *it* seems not to be unique. The language of geometry overloads the verb *lie outside*, so that

$$\left.\begin{array}{l}\text{outside\_PL}(x, y) \\ \text{outside\_LP}(x, y)\end{array}\right\} \;\; \triangleright \;\; \textit{x lies outside y},$$

whence *it lies outside it* has two interpretations that, although equivalent, are distinct propositions. We cannot tell whether the sentence says that the point lies outside the line or that the line lies outside the point. But this explanation of the strangeness already contains the solution, which is to use definite noun phrases instead of pronouns,

point such that there is a line and the point lies outside the line.

To finish the first way of sugaring the axiom of parallels, we replace the first argument in

$$\text{DAP}(p(z), p(q(z)))$$

by the quantifier phrase, as explained in Section 5, and the second argument by a pronoun or a definite phrase of the type line. Applying the sugaring rules for DAP and Pron and choosing the expression for the domain to be the first one cited above, gives

every point that lies outside a line determines a parallel to it as an unambiguous statement of the axiom of parallels. The word-to-word formalization of this sentence is the definitional variant

$$\text{every}(\text{that}(\text{point}, (x)\text{Indef}(\text{line}, (y)\text{outside\_PL}(x, y))), (z)\text{DAP}(p(z), \text{Pron}(\text{line}, p(q(z)))))$$

of the original proposition.

The reader can check that the proposition also has the following variants, and find some more of them.

if a point lies outside a line, it determines a parallel to it,

if there is a point such that there is a line such that the point lies outside the line the point determines a parallel to the line.

Observe that the two occurrences of *it* in the first variant are uniquely interpretable, because *determines a parallel to* is not overloaded. An early implementation of sugaring, written in Prolog by Petri Mäenpää, found 1128 variants of a donkey sentence with the same structure as the axiom of parallels.

8 Some uses of the plural

In the informal language of mathematics, it is often possible to find clear and unambiguous usages of linguistic structures that appear as hopelessly complex, if an unlimited fragment of natural language is taken under consideration. One such structure is the plural, which has been a persistent problem in logical semantics of Montague style. It has several uses that, when cooccurring, lead to multiple ambiguities. Mathematical texts still make unambiguous use of the plural, e.g. in the sentences

points A and B lie on the line a,

A and B are equal points,

all lines that pass through the center of a circle intersect its circumference.

The first of these sentences shows what von Plato (1993) defines as the term conjunction,

$$C(a.b) = C(a)\&C(b) : \text{prop for } A : \text{set},\ A : (A)\text{prop},\ a : A,\ b : A.$$

It is thus propositionally equal to the sentence

the point A lies on the line a and the point B lies on the line a,

in which no plural form occurs. In this case, the plural is just used for finding a more concise expression.

The second sentence does not employ the term conjunction, but it is propositionally equal to the singular sentence

A is equal to B

The difference between the first sentence and this one is an instance of the distinction between what is called distributive and nondistributive plural in linguistics. The distributive plural can be analyzed as a conjunction of singular instances, but the nondistributive plural cannot. For this particular sentence, we do have a nonplural equivalent, but I am not sure whether we always do.

The third sentence is propositionally equal to *every line that passes through the center of a circle intersects its circumference.*

Here there is no difference between *all lines* and *every line*, except the number agreement of the verb.

We have formulated a sugaring algorithm producing these uses of the plural (Ranta 1993, Chapter 9). In each of these cases, the plural forms of nouns and verbs are only produced in the sugaring process, and there is no type-theoretical operator corresponding to the plural. The rules we have discussed do not yet cover all uses of the plural in the informal language of mathematics. (But as long as we work in the direction of sugaring only, it makes no harm that all uses of an English mode of expression are not produced.) For instance, we do not yet quite understand the nondistributive use of the quantifier word *all* as in

all lines that pass through the center of a circle converge.

Nor do we quite understand the use of the plural pronoun *they*, which is sometimes distributive, paraphrasable by the term conjunction, e.g.

if A and B do not lie outside the line a, they are incident on it,

but sometimes used on the place of the "surface term conjunction", so that it fuses together the arguments of a predicate, e.g.

if a and b do not converge, they are parallel.

9 Problems and prospects

As indicated in the beginning of this paper, very little linguistic work has been done concerning the informal language of mathematics. To capture the essential structure of mathematical text, a grammatical representation of it should, at least, be able to express the mathematical propositions precisely. This can hardly be expected from all grammars in standard linguistics, but requires a grammatical formalism that comprises logic. Moreover, the formal and the informal language should be tied together by sugaring and parsing algorithms that satisfy the following condition.

> A correct informal proof results, when parsed, in a correct formal derivation, and vice versa.

There are two properties concerning ambiguity that can be stated. First,

> all expressions of the informal fragment are unambiguous.

But this is maybe too severe a condition. It makes little harm if the English fragment recognized contains ambiguities, if only the parser can detect them and ask the user to disambiguate. Instead, one can pose the weaker condition that

> every proposition of the formal theory can be expressed by an unambiguous English sentence.

A sugaring program satisfying this condition can provide a natural language interface to a formal proof system, stating theorems and their proofs in an easily readable form.

When considering mathematical language, instead of the fragment of everyday language familiar to the linguist, one soon realizes both a higher demand of unambiguity and a higher complexity of the propositions. There is still work to be done to find a sugaring algorithm that gives unambiguous expressions for all propositions of a formal theory. One particular problem is that the context in which a proposition is formed can be arbitrarily large, so that there are not enough anaphoric expressions to refer to each object uniquely. A very simple such context is created by the opening

given two lines, ...

formalizable by the quantifier

$$(\Pi z : (\Sigma x : line)line)$$

The anaphoric expressions that can be used for an arbitrary line are *it* and *the line*, but neither of these refers uniquely in this context. One way to solve this problem is to use the expressions *the first line*, *the second line*. Another one, much more idiomatic in mathematical language, is to introduce variables,

given two lines a and b, ...

whereafter reference can be made to *the line a* and to *the line b*. But this opening cannot be formalized as a quantifier, because the variable names are not usable outside the scope of the quantifier. The axiom of parallels in the formulation

if a point A lies outside a line a, A determines a parallel to a.

cannot thus be given the logical form we gave it in Section 7.

A more general defect of our Montague style grammar is that it only concerns propositions and not judgements, of which type theory has several forms that are all needed in precise formalization of mathematics. What we have done here only suffices for expressing axiomatic theories, in the format familiar from the metamathematical thinking of this century. Going beyond this format in mathematics, type theory also shows a model for grammar in general, to extend its views from propositions to judgements and other linguistic acts.

References

Noam Chomsky, *Syntactic Structures*, Mouton, The Hague, 1957.

Gottlob Frege, *Begriffsschrift*, Louis Nebert, Halle, 1879.

David Hilbert, *Grundlagen der Geometrie*, 1899; 2. Aufl. Teubner, Leipzig, 1903.

Per Martin-Löf, *Intuitionistic Type Theory*, Bibliopolis, Naples, 1984.

Richard Montague, "English as a formal language", B. Visentini & al. (eds), *Linguaggi nella società e nella tecnica*, Edizioni di comunità, Milan, 1970. Reprinted in R. Montague, *Formal Philosophy*, Yale University Press, New Haven, 1974., .

Bengt Nordström, Kent Petersson and Jan Smith, *Programming in Martin-Löf's Type Theory. An Introduction*, Clarendon Press, Oxford, 1990.

Jan von Plato, "Axiomatization of geometry based on separation principles: point and line apartness and line convergence", paper manuscript, University of Helsinki, 1993.

Aarne Ranta, "Intuitionistic categorial grammar", *Linguistics and Philosophy* 14, pp. 203–239, 1991.

Aarne Ranta, “Type-Theoretical Grammar”, book manuscript, University of Helsinki, 1993. To appear in the Indices series at Oxford University Press.