Syntactic Categories in the Language of Mathematics

Aarne Ranta

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

In this paper, we shall present a system of syntactic categories that unifies the expressive capacities of constructive type theory and phrase structure grammar. It makes both the type distinctions of the former and the distinctions between different linguistic categories (such as adjectives and verbs) of the latter. Both kinds of distinctions are needed in an accurate description of the informal language of mathematics.

The system of categories, a generative grammar of a fragment of English, and a semantic interpretation of the fragment in standard type theory are all presented formally in the higher-level type theory of Martin-Löf and implemented in the proof editor ALF ("Another Logical Framework", see Magnusson and Nordström 1994). The implementation gives a method of sugaring syntax trees into strings of English words, as well as a method of interpreting syntax trees in type theory. Parsing strings of words to get syntax trees can be performed as interactive search of construction, but how far parsing can be made algorithmic is still an open question.

The work presented here can be seen as a formal linguistic elaboration of the type-theoretical analysis of natural language discussed in Ranta 1994a and 1994b. But a similar system of syntactic categories is also useful in the definition of formal languages, if one wants to have variation in the concrete syntax (such as prefix and posfix operators). For this purpose, Petri Mäenpää (1994) has investigated the generalization of the Backus-Naur formalism in a way analogous to our generalization of phrase structure grammar.

1 Logical and Grammatical Structure

In his *Begriffsschrift* (1879), Frege rejected the traditional analysis of language in terms of subjects and predicates, and replaced it by an analysis into functions and arguments. Frege's purpose was to reveal the logical structure of language, the structure that is relevant for the correctness of inference, and abstract from all other structure. The logically most important advantage of Frege's move were the distinctions he was able to make between different logical structures disguised in one and the same linguistic structure. Thus the subject *seven* of the sentence

seven is prime

comes out as the argument 7 of the function Pr in the proposition

$$\mathrm{Pr}(7),$$

but the subject *some number* of

some number is prime

comes out as the function $\exists(N)$ that operates on the argument Pr in

$$\exists(N)(\mathrm{Pr}).$$

But the function argument structure revealed in logical analysis does not only make explicit what the traditional grammatical structure hides. It also suppresses distinctions in the linguistic structure. Thus a two-place predicate, such as the geometric incidence predicate

$$\mathrm{In}(a, l)$$

is linguistically manifested now as a transitive verb,

l contains a,

now as a verb with a preposition,

a lies on l,

now as an adjective with a preposition,

a is incident with l,

now as an adjective with a collective subject,

a and l are incident.

We have here four linguistic categories corresponding to one and the same logical category of two-place predicates. When complex sentences are formed, expressions of different linguistic categories are combined in different ways, and these differences cannot be spelt out in terms of the logical categories.

2 Phrase Structure Grammar

In linguistics, the traditional system of categories based on the subject predicate relation has been refined to analysing the structure of all kinds of phrases. Through the work of Chomsky (1957), a notation has been established for this kind of analysis, called phrase structure grammar. The first rule, so to say, corresponding to the subject predicate analysis, says that a sentence (S) may consist of a noun phrase (NP) combined with a verb phrase (V1),

$$\mathrm{S} \rightarrow \mathrm{NP\ V1}.$$

(In the Backus-Naur notation used in the definition of programming languages, the same rule could also be written S ::= NP V1.) The verb phrase can now have the structure of a transitive verb (V2) combined with a noun phrase,

$$\mathrm{V1} \rightarrow \mathrm{V2\ NP}.$$

The transitive verb can be composed of a two-place verb taking a preposition r (V4(r)), by adding the preposition,

$$\text{V2} \rightarrow \text{V4}(r)\ r.$$

Prepositions, expressions of the category Prep, are single words, such as

Prep $\rightarrow$ *on*,
Prep $\rightarrow$ *with*.

The V4(r) verb can consist of the copula *is* with a two-place adjective (A4(r)) taking the same preposition r,

$$\text{V4}(r) \rightarrow \textit{is}\ \text{A4}(r).$$

But a sentence need not have the subject predicate structure NP V1. It may also be formed from two noun phrases combined with a collective verb (V5),

$$\text{S} \rightarrow \text{NP}\ \textit{and}\ \text{NP V5}.$$

The collective verb may, in turn, be formed by adding the plural copula *are* to a collective adjective (A5),

$$\text{V5} \rightarrow \textit{are}\ \text{A5}.$$

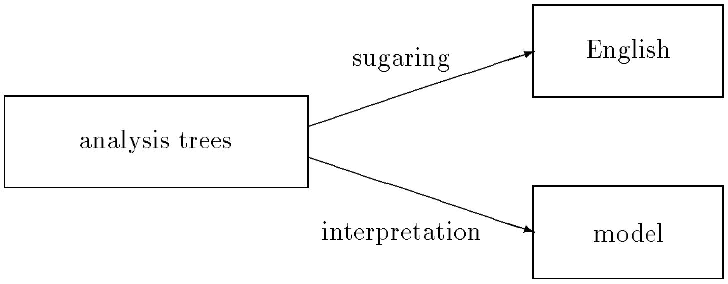
Now the linguistic variants of the incidence predicate are each assigned a different category,

V2 $\rightarrow$ *contains*,
V4(on) $\rightarrow$ *lies*,
A4(with) $\rightarrow$ *incident*,
A5 $\rightarrow$ *incident*.

(The system of phrase structure rules just presented is not from any particular source, but has been designed with the purpose of giving a logical interpretation to these grammatical structures. In particular, the treatment of collective predication separately from the NP V1 structure might be rejected for simplicity reasons in another kind of analysis.)

3 Montague Grammar

In the work of Montague (1974), the logical and linguistic analyses were unfied in the formalism of what he called analysis trees, which simultaneously admit of an unambiguous logical interpretation and preserve the syntactic distinctions of natural language. A Montague style formulation of the above phrase structure grammar has five syntactic categories, V2, V4(on), V5, A4(with), and A5, corresponding to the one and the same semantic category $(D)(D)$prop of two-place propositional functions on the domain D. (In Montague's notation, the semantic category is $(e,(e,t))$.) The expressions of the syntactic categories—the analysis trees—can be interpreted in a model and, at the same time, sugared into English, that is, into strings of words without tree structure.



(Montague himself did not have any special word for the operation that we call sugaring.)

The type theory that Montague employed was simple type theory with one domain of individuals only and no dependent types. Simple type theory suffices well for the interpretation of phrase structure grammar, which does not make domain distinctions either. But for an accurate analysis of the language of mathematics, it is not enough, since the combinations of phrases have domain restrictions. For instance, the verbs *contain* and *intersect* are both transitive verbs, of the category V2, but they apply to different domains. The subject can be, in both cases, a line, but the object of *contain* is a point and the object of *intersect* is a line. Logically, we have the propositional functions

$\mathrm{Cont} : (\mathrm{Ln})(\mathrm{Pt})\mathrm{prop},$
$\mathrm{Int} : (\mathrm{Ln})(\mathrm{Ln})\mathrm{prop},$

to interpret the verbs *contain* and *intersect*, respectively. If a verb is combined with a noun phrase of wrong type, as in the sentence

every line intersects the point a,

the interpretation does not work.

In linguistics, domain restrictions have been studied under the title of selectional restrictions, since Chomsky 1965. They have, however, remained outside the hard core of formal grammar and logical semantics, and one reason for this is that categorial grammar based on simple type theory is unable to formalize them.

4 Domain-Relative Syntactic Categories

To have a grammar that combines the expressive capacities of phrase structure grammar and type theory with different domains, we relativize the standard grammatical categories to domains. The category V1 of verb phrases is replaced by the family of categories

$$\mathrm{V1}(A),$$

where $A : \mathrm{dom}$, of verb phrases of subject type A. The expressions of $\mathrm{V1}(A)$ are interpreted as objects of type

$$(A)\mathrm{prop}.$$

The categories V2, V4(r), V5, A4(r), and A5, where r : Prep, are replaced by the families

V2(A, B),
V4(A, r, B),
V5(A, B),
A4(A, r, B),
A5(A, B),

respectively, where A : dom, B : dom. The expressions of each of these categories are interpreted as objects of type

$$(A)(B)\mathrm{prop}.$$

The type dom is the type of the domains of individuals that the language to be defined speaks about. It cannot be identified with the type set of all sets, as this would make the definitions of the syntactic categories, which are sets themselves, impredicative. (This was pointed out to me by Thorsten Altenkirch.) In this paper, dom is presented as a universe à la Russell, that is, the same notation is used for members of dom and the corresponding sets. In an implementation in ALF, however, such overloading of expressions is not possible, and a formulation à la Tarski must be followed, in which $T(A)$ is the set corresponding to A : dom. (See Martin-Löf 1984 for universes à la Russell and à la Tarski.)

To control the compatibility of the subject and object with the verb, we also replace the category NP of noun phrases by the family of categories

$$\mathrm{NP}(A),$$

where A : dom, of noun phrases of type A. The expressions of NP(A) are interpreted as objects of type

$$((A)\mathrm{prop})\mathrm{prop},$$

that is, as quantifiers over the domain A. This interpretation of noun phrases is due to Montague.

In type theory, the application of a quantifier, like

$$\Pi : (X : \mathrm{set})((X)\mathrm{prop})\mathrm{prop},$$

to a domain A indeed yields an object of the right type to interpret a noun phrase of type A,

$$\Pi(A) : ((A)\mathrm{prop})\mathrm{prop}.$$

In English, a noun phrase can be built by combining a quantifier word (Q1) with a common noun (CN),

$$\mathrm{NP} \rightarrow \mathrm{Q1\ CN}.$$

To give a type-theoretical version of this rule, we notice that the expressions of the category

$$\mathrm{CN}$$

can be interpreted as objects of type

$$\mathrm{dom}.$$

There is no dependence on domain in the category CN. Expressions of

$$\mathrm{Q1}$$

can be interpreted as objects of type

$$(X : \mathrm{dom})((X)\mathrm{prop})\mathrm{prop},$$

which is also without dependencies. The combination of an expression $R : \mathrm{Q1}$ with an expression $A : \mathrm{CN}$ can now be interpreted as an application of the interpretation of R to the interpretation of A. This application is of type

$$((A^\star)\mathrm{prop})\mathrm{prop},$$

where $A^\star$ is the interpretation of A. (We shall later show how the other grammatical combinations are interpreted.)

To finish the domain relativization of the system of syntactic categories, we notice that the category

$$\mathrm{S}$$

of sentences has no dependencies, and its expressions are interpreted as objects of the type

$$\mathrm{prop}$$

of propositions, which is classically the set Bool of truth values, but intuitionistically the type set of sets (in the present treatment, the universe dom of domains). The general principle in type relativization is that if a category C is interpreted in a type that is formed by using the domain D, it is replaced by the family of categories $C(A)$, where $A : \mathrm{dom}$ is arbitrary, and interpreted in the corresponding type where the constant D is replaced by the variable A. If D occurs several times in the type, all occurrences are replaced by distinct domain variables, A, B, C, etc.

5 Proper Names and Noun Phrases

To Frege's distinction between quantifier phrases and proper names corresponds the grammatical distinction between noun phrases and proper names. In phrase structure grammar, proper names (PN) are often treated as a special class of noun phrases,

$$\mathrm{NP} \rightarrow \mathrm{PN}.$$

In type theory, we replace the category PN by the family of categories

$$\mathrm{PN}(A),$$

where A : dom, of proper names of type A. Expressions of $\mathrm{PN}(A)$ are interpreted as elements of the set

$$A.$$

Montague wanted to analyse quantifier phrases like *every line* in the same way as proper names. He found a way of interpreting the subsumption of the category PN under the vategory NP, by introducing the raising operator

$$\mathrm{RAISE} : (\mathrm{PN})\mathrm{NP},$$

whose application to a proper name a is interpreted as the function

$$(Y)Y(a^{\star}) : ((D)\mathrm{prop})\mathrm{prop},$$

where $a^{\star} : D$ is the interpretation of a. With arbitrary domains, the corresponding operator is

$$\mathrm{RAISE} : (A : \mathrm{dom})(\mathrm{PN}(A))\mathrm{NP}(A),$$

and we can interpret it in the same way as Montague.

The language of mathematics has a rich system of complex proper names, built by using functional expressions. There are one-place functional expressions ($\mathrm{F1}(r)$) taking their arguments with a preposition r,

$$\mathrm{PN} \rightarrow \textit{the}\ \mathrm{F1}(r)\ r\ \mathrm{PN}.$$

The type-theoretical generalization of $\mathrm{F1}(r)$, where r : Prep, is the family of categories

$$\mathrm{F1}(r, A, B),$$

where A : dom, B : dom. Expressions of the category $\mathrm{F1}(r, A, B)$ are interpreted as objects of type

$$(A)B.$$

An example of such an expression is

$$\mathrm{centre} : \mathrm{F1}(\mathrm{of}, \mathrm{Cl}, \mathrm{Pt}),$$

which is interpreted as the function that assigns to each circle the centre of it.

There are also two-place functional expressions ($\mathrm{F2}(r, s)$) taking their arguments with possibly different prepositions. Their application can be analysed as a two-step process, first taking one argument and producing an expression of type $\mathrm{F1}(s)$,

$$\mathrm{F1}(s) \rightarrow \mathrm{F2}(r, s)\ r\ \mathrm{PN}.$$

The type-theoretical generalization of $\mathrm{F2}(r, s)$, where r, s : Prep, is the family of categories

$$\mathrm{F2}(r, A, s, B, C),$$

where A, B, C : dom. Expressions of the category $\mathrm{F2}(r, A, s, B, C)$ are interpreted as objects of type

$$(A)(B)C.$$

An example is

$$\mathrm{parallel} : \mathrm{F2}(\mathrm{to}, \mathrm{Ln}, \mathrm{through}, \mathrm{Pt}, \mathrm{Ln}),$$

which is interpreted as the function that to a given line and point assigns the parallel to the line through the point.

6 Principles of the Formalization of the Grammar

Now we have described, informally, a system of syntactic categories and rules for forming expressions of these categories, as well as the semantic interpretation of this system in type theory. To formalize this system, we shall define the syntactic categories as sets and families of sets in type theory. In this section, we shall illustrate the principles of this formalization with a very small fragment of the grammar. More rules will be shown in the sections that follow.

The categories of sentences, noun phrases, and verb phrases have the following formation rules.

$$\mathrm{S} : \mathrm{set}, \qquad \frac{A : \mathrm{dom}}{\mathrm{NP}(A) : \mathrm{set}}, \qquad \frac{A : \mathrm{dom}}{\mathrm{V1}(A) : \mathrm{set}}.$$

The phrase structure rules are formalized as introduction rules of these sets. The introduction rule corresponding to the phrase structure rule

$$\mathrm{S} \rightarrow \mathrm{NP}\ \mathrm{V1}$$

says that an element of S may be formed from an element of $\mathrm{NP}(A)$ and an element of $\mathrm{V1}(A)$, where $A : \mathrm{dom}$. We give the corresponding constructor the suggestive name SUBJ.

$$\frac{A : \mathrm{dom} \quad Q : \mathrm{NP}(A) \quad F : \mathrm{V1}(A)}{\mathrm{SUBJ}(A, Q, F) : \mathrm{S}}.$$

The introduction rule alone does not generate an English sentence, in the sense of a sequence of words, but a functional term, or, as a linguist would say, a syntax tree. To get strings from trees, we define sugaring functions for each category,

$$\begin{aligned}
&\mathrm{SUGS} : (\mathrm{S})E,\\
&\mathrm{SUGNP} : (A : \mathrm{dom})(\mathrm{NP}(A))E,\\
&\mathrm{SUGV1} : (A : \mathrm{dom})(\mathrm{Number})(\mathrm{V1}(A))E,
\end{aligned}$$

where E is the set of strings of words and Number is the set of the two grammatical numbers, the singular and the plural. (We need to be able to produce both singular and plural forms of verbs.) Each introduction rule has a corresponding sugaring rule that tells how the sugaring of the complex is composed from the sugarings of its constituents. For SUBJ, the sugaring rule is

$$\mathrm{SUGS}(\mathrm{SUBJ}(A, Q, F)) = \mathrm{SUGNP}(A, Q)\_\mathrm{SUGV1}(A, \mathrm{NUMNP}(A, Q), F),$$

where _ appends two strings and

$$\mathrm{NUMNP} : (A : \mathrm{dom})(\mathrm{NP}(A))\mathrm{NUMBER}$$

is a function that assigns the grammatical number to noun phrases. (Thus the sugaring rule expresses the grammatical principle that the verb of a sentence receives the number of the subject.)

Formally, the sugaring rule is the pattern equation that defines the function SUGS for those elements of S that have the form SUBJ(A, Q, F) (cf. Coquand 1992). We shall use the judgement form

$$a \; = \; b$$

for pattern equations, following the notation of ALF. In the presentation of the grammar and the lexicon, it is not so natural to list the pattern equations after the functions that they define, but to scatter them around, after each introduction rule of a syntactic category.

In addition to the sugaring function, every syntactic category has an interpretation function, for instance,

IS : (S)prop,
INP : $(A$: dom)(NP(A))((A)prop)prop,
IV1 : $(A$: dom)(V1(A))(A)prop.

Every introduction rule of a syntactic category has a corresponding interpretation rule that tells how the interpretation of the complex is composed from the interpretations of its constituents. For SUBJ, the interpretation rule is the pattern equation

$$\mathrm{IS}(\mathrm{SUBJ}(A, Q, F)) \; = \; \mathrm{INP}(A, Q, \mathrm{IV1}(A, F)).$$

As we have relativized syntactic categories to domains, the introduction rules may have to contain, as premises, judgements of the form A : dom. The premises that follow may then depend on A. This means that the type of the constructor in question cannot be formed without using dependent types: the higher-level judgement introducing SUBJ is

$$\mathrm{SUBJ} : (A : \mathrm{dom})(Q : \mathrm{NP}(A))(F : \mathrm{V1}(A))\mathrm{S}.$$

A type-theoretical formalization of ordinary phrase structure grammar, with just one domain D, can be given without dependent types, as was done by Montague. The judgement corresponding to the rule S $\rightarrow$ NP V1 is

$$\mathrm{SUBJ} : (\mathrm{NP})(\mathrm{V1})\mathrm{S}.$$

(Bar-Hillel 1960 presents a proof, for which he gives credit to Gaifman and Shamir, of the equivalence between context-free phrase structure grammars and categorial grammars without dependent types.)

In addition to requiring dependent types, the relativization of syntactic categories to domains requires a radical adjustment of the architecture of formal grammar. For Montague, it was possible to define the language of analysis trees first and only then to build a model and give an interpretation of the language. But in our grammar, the definition of the language already presupposes the definition of the model, since the syntactic rules—the introduction rules of the syntactic categories—have semantic premises, that is, premises introducing arguments that do not belong to the syntactic categories but to semantic types. Such is the premise A : dom of the SUBJ rule.

Now that the definition of the language depends on the model, we could not be satisfied with an informal model theory, like Montague. We must define the model formally, in type theory, in the same way as the language. Otherwise the definition of the language would not be completely formal. (This remark is due to Petri Mäenpää.)

The syntax trees are not composed from other syntax trees only, but they have semantic constituents as well. In the tree $\mathrm{SUBJ}(A, Q, F)$, the constituents Q and F are syntactic, but A is semantic. The fundamental principle that has governed our design of the grammar is that

all and only semantic arguments are deleted in sugaring.

That all semantic arguments must be deleted is clear from the fact that sugaring is defined for the syntactic categories only. That no syntactic arguments are deleted conforms to the practices of phrase structure grammar: complex expressions are built by combining given expressions together, and no constituent is ever deleted.

Our syntax trees are thus, so to say, like the trees of ordinary phrase structure grammar completed with semantic arguments. What we have considered so far are type information arguments. But the fundamental principle just stated extends to other cases of "hidden information", such as the anaphoric pronoun

$$\mathrm{Pron} : (A : \mathrm{dom})(a : A)\mathrm{PN}(A),$$

with the rule that $\mathrm{Pron}(A, a)$ always sugars to the word *it*, so that both arguments are deleted (cf. Ranta 1994a, chapter 4, and 1994c, section 9).

In the language of geometry, we have expressions like

the intersection point of l and m.

In phrase structure grammar, this is just an expression of the category PN built from the given proper names l and m. If we introduce a new category F4 for expressions like *intersection point*, we have the phrase structure rule

$$\mathrm{PN} \rightarrow \textit{the}\ \mathrm{F4}\ \textit{of}\ \mathrm{PN}\textit{and}\ \mathrm{PN}.$$

But in a closer analysis, we notice that the expression *the intersection point of l and m* is well formed only if l and m are proper names of lines that converge. The phrase structure rule does not show any conditions concerning l and m. To express such conditions, we replace the category F4 by the family of categories

$$\mathrm{F4}(A, B, C, D),$$

where A, B, D : dom and $C : (A)(B)\mathrm{prop}$. The expressions of $\mathrm{F4}(A, B, C, D)$ are interpreted as objects of type

$$(x : A)(y : B)(C(x, y))D.$$

In sugaring an application instance of F4, the third argument, the proof of C for the first two arguments, is deleted.

Parsing into ordinary phrase structure trees, that is, into our syntax trees with semantic arguments left blank, can be performed by standard context-free techniques. In some cases, semantic arguments can then be filled in by type checking. (This seems to be the case when they are non-complex type information arguments, as in almost all examples of this paper.) But there is no reason why type checking should always be decidable. For instance, to see whether the expression *the intersection point of l and m* is well formed, one has to find out whether a proof is available that the lines l and m converge, and this is not generally decidable.

7 Combinators for Predication

Now we shall formulate the rest of the phrase structure rules presented above as introduction rules of the syntactic categories. For each constructor of a syntactic category, we shall tell how the expressions formed by it are interpreted and sugared.

We start by introducing constructors for the formation of complex expressions. These constructors will be called combinators, since they are used for combining expressions into more complex expressions. (Moreover, their interpretations correspond to certain combinators of type theory, in the standard sense of the word combinator.)

For brevity, as well as for readability, we shall use the notation

$$c^{\star}$$

for the interpretation of the expression c, and the notation

$$c^{\circ}$$

for the sugaring of c, irrespective of the category of c. These notations should be easy to replace by the type-theoretical operators IS, INP, SUGS, SUGNP, etc. In those cases in which sugaring depends on the grammatical number, we write

$$c^{\circ}[n]$$

for the sugaring of c in the number n.

Thus the first syntactic rule is formulated as follows.

S $\rightarrow$ NP V1.

$$\frac{A : \mathrm{dom} \quad Q : \mathrm{NP}(A) \quad F : \mathrm{V1}(A)}{\begin{array}{l}\mathrm{SUBJ}(A,Q,F) : \mathrm{S}, \\ \mathrm{SUBJ}(A,Q,F)^{\star} = Q^{\star}(F^{\star}), \\ \mathrm{SUBJ}(A,Q,F)^{\circ} = Q^{\circ}\_F^{\circ}[\mathrm{NUMNP}(A,Q)].\end{array}}$$

The combinator OBJ combines a transitive verb with an object noun phrase to form an intransitive verb. The inflectional operations on this complex verb (the formation of the singular and the plural) are performed on the transitive verb.

V1 $\rightarrow$ V2 NP.

$$\frac{A : \mathrm{dom} \quad B : \mathrm{dom} \quad F : \mathrm{V2}(A,B) \quad Q : \mathrm{NP}(B)}{\begin{array}{l} \mathrm{OBJ}(A,B,F,Q) : \mathrm{V1}(A), \\ \mathrm{OBJ}(A,B,F,Q)^{\star} = (x)Q^{\star}((y)F^{\star}(x,y)), \\ \mathrm{OBJ}(A,B,F,Q)^{\circ}[n] = F^{\circ}[n]\_Q^{\circ}. \end{array}}$$

A transitive verb can be formed from a verb taking a preposition by attaching the preposition to it. Notice that prepositions in syntax trees are not words but formal expressions, elements of the set Prep, from which words are obtained by the function PREP.

V2 $\rightarrow$ V4(r) r.

$$\frac{A : \mathrm{dom} \quad r : \mathrm{Prep} \quad B : \mathrm{dom} \quad F : \mathrm{V4}(A,r,B)}{\begin{array}{l} \mathrm{TRANS}(A,r,B,F) : \mathrm{V2}(A,B), \\ \mathrm{TRANS}(A,r,B,F)^{\star} = F^{\star}, \\ \mathrm{TRANS}(A,r,B,F)^{\circ}[n] = F^{\circ}[n]\_\mathrm{PREP}(r). \end{array}}$$

A two-place adjective taking a preposition r can be turned into a verb taking the same preposition by prefixing the copula be to it. The complex verb is inflected by inflecting the copula, for which we have the operator VF.

V4(r) $\rightarrow$ *is* A4(r).

$$\frac{A : \mathrm{dom} \quad r : \mathrm{Prep} \quad B : \mathrm{dom} \quad F : \mathrm{A4}(A,r,B)}{\begin{array}{l} \mathrm{COMPLADPRED}(A,r,B,F) : \mathrm{V4}(A,r,B), \\ \mathrm{COMPLADPRED}(A,r,B,F)^{\star} = F^{\star}, \\ \mathrm{COMPLADPRED}(A,r,B,F)^{\circ}[n] = \mathrm{VF}(n,\mathrm{be})\_F^{\circ}. \end{array}}$$

The following two rules concern collective predication. Notice that we have restricted collective predication to proper names, excluding quantifier phrases, since the interpretation of English sentences like *every line and some line are parallel* is obscure.

S $\rightarrow$ PN *and* PN V5.

$$\frac{A : \mathrm{dom} \quad B : \mathrm{dom} \quad a : \mathrm{PN}(A) \quad b : \mathrm{PN}(B) \quad F : \mathrm{V5}(A,B)}{\begin{array}{l} \mathrm{COLL}(A,B,a,b,F) : \mathrm{S}, \\ \mathrm{COLL}(A,B,a,b,F)^{\star} = F^{\star}(a^{\star},b^{\star}), \\ \mathrm{COLL}(A,B,a,b,F)^{\circ} = a^{\circ}\_and\_b^{\circ}\_F^{\circ}[\mathrm{PL}]. \end{array}}$$

V5 $\rightarrow$ *are* A5.

$$\frac{A : \mathrm{dom} \quad B : \mathrm{dom} \quad F : \mathrm{A5}(A,B)}{\begin{array}{l} \mathrm{COLLADPRED}(A,B,F) : \mathrm{V5}(A,B), \\ \mathrm{COLLADPRED}(A,B,F)^{\star} = F^{\star}, \\ \mathrm{COLLADPRED}(A,B,F)^{\circ}[n] = \mathrm{VF}(n,\mathrm{be})\_F^{\circ}. \end{array}}$$

Let us summarize all these ways of building sentences in a diagram.

$$
\begin{tikzcd}
\mathrm{A4}(A,r,B) \arrow[d, "\mathrm{COMPLADPRED}"'] & & & \\
\mathrm{V4}(A,r,B) \arrow[d, "\mathrm{TRANS}"'] & & & \mathrm{A5}(A,B) \arrow[d, "\mathrm{COLLADPRED}"] \\
\mathrm{V2}(A,B) \arrow[dr, "\mathrm{OBJ}"'] & & & \mathrm{V5}(A,B) \arrow[ddll, "\mathrm{COLL}"] \\
& \mathrm{V1}(A) \arrow[dr, "\mathrm{SUBJ}"'] & & \\
& & \mathrm{S} &
\end{tikzcd}
$$

(The diagram does not show that OBJ, SUBJ, and COLL require noun phrase arguments as well.)

8 Combinators for Noun Phrases

Noun phrases can be formed both by combining quantifiers with common nouns and by raising proper names in the style of Montague.

NP $\rightarrow$ Q1 CN.

$$
\frac{R : \mathrm{Q1} \quad A : \mathrm{CN}}{\begin{array}{l}\mathrm{QUANT}(R,A) : \mathrm{NP}(A^\star),\\ \mathrm{QUANT}(R,A)^\star \ = \ R^\star(A^\star),\\ \mathrm{QUANT}(R,A)^\circ \ = \ R^\circ\_A^\circ.\end{array}}
$$

NP $\rightarrow$ PN.

$$
\frac{A : \mathrm{dom} \quad a : \mathrm{PN}(A)}{\begin{array}{l}\mathrm{RAISE}(A,a) : \mathrm{NP}(A),\\ \mathrm{RAISE}(A,a)^\star \ = \ (Y)Y(a^\star),\\ \mathrm{RAISE}(A,a)^\circ \ = \ a^\circ.\end{array}}
$$

All noun phrases defined in this section are of the singular number.

Proper names can be formed by applying functional expressions to given proper names.

PN $\rightarrow$ *the* F1(r) r PN.

$$
\frac{r : \mathrm{Prep} \quad A : \mathrm{dom} \quad B : \mathrm{dom} \quad f : \mathrm{F1}(r,A,B) \quad a : \mathrm{PN}(A)}{\begin{array}{l}\mathrm{APPLY}(r,A,B,f,a) : \mathrm{PN}(B),\\ \mathrm{APPLY}(r,A,B,f,a)^\star \ = \ f^\star(a^\star),\\ \mathrm{APPLY}(r,A,B,f,a)^\circ \ = \ \mathit{the}\_f^\circ\_\mathrm{PREP}(r)\_a^\circ.\end{array}}
$$

$\mathrm{F1}(s) \rightarrow \mathrm{F2}(r,s)\ r\ \mathrm{PN}.$

$$\frac{r : \mathrm{Prep} \quad A : \mathrm{dom} \quad s : \mathrm{Prep} \quad B : \mathrm{dom} \quad C : \mathrm{dom} \qquad f : \mathrm{F2}(r,A,s,B,C) \quad a : \mathrm{PN}(A)}{\begin{array}{l}\mathrm{PARTAPPLY}(r,A,s,B,C,f,a) : \mathrm{F1}(s,B,C), \\ \mathrm{PARTAPPLY}(r,A,s,B,C,f,a)^{\star} = (y)f^{\star}(a^{\star},y), \\ \mathrm{PARTAPPLY}(r,A,s,B,C,f,a)^{\circ} = f^{\circ}\_\mathrm{PREP}(r)\_a^{\circ}.\end{array}}$$

PN $\rightarrow$ *the* F4 *of* PN *and* PN.

$$\frac{A : \mathrm{dom} \quad B : \mathrm{dom} \quad C : (A)(B)\mathrm{prop} \quad D : \mathrm{dom} \qquad f : \mathrm{F4}(A,B,C,D) \quad a : \mathrm{PN}(A) \quad b : \mathrm{PN}(B) \quad c : C(a^{\star},b^{\star})}{\begin{array}{l}\mathrm{DRAW}(A,B,C,D,f,a,b,c) : \mathrm{PN}(D), \\ \mathrm{DRAW}(A,B,C,D,f,a,b,c)^{\star} = f^{\star}(a^{\star},b^{\star},c), \\ \mathrm{DRAW}(A,B,C,D,f,a,b,c)^{\circ} = \mathit{the}\_f^{\circ}\_\mathit{of}\_a^{\circ}\_\mathit{and}\_b^{\circ}.\end{array}}$$

9 Lexicon

The rules that we stated in the previous sections lay down a general grammatical framework, in which particular fragments of English can be defined by means of lexical entries. A lexical entry introduces an expression of some category and tells how it is interpreted and sugared, just as we did for the combinators above. This is usually done in an environment in which a mathematical theory is given, so that the new expressions are interpreted in that theory. (The combinators, as we saw, are interpreted in the logical framework, independently of particular mathematical theories.) Most of the expressions we have given as examples are interpretable in von Plato's constructive axiomatization of geometry. (See von Plato. to appear. The theory, as well as a lexicon giving English versions of all its basic expressions, has been implemented in ALF.)

To illustrate the format of the lexicon, let us assume a fragment of constructive geometry: the sets of points and lines, the distinctness predicats for points and lines, the convergence (=intersection) predicate of lines, the parallel construction, and the intersection point construction.

$\mathrm{Pt} : \mathrm{dom},$
$\mathrm{Ln} : \mathrm{dom},$
$\mathrm{DiPt} : (\mathrm{Pt})(\mathrm{Pt})\mathrm{prop},$
$\mathrm{DiLn} : (\mathrm{Ln})(\mathrm{Ln})\mathrm{prop},$
$\mathrm{Con} : (\mathrm{Ln})(\mathrm{Ln})\mathrm{prop},$
$\mathrm{par} : (\mathrm{Ln})(\mathrm{Pt})\mathrm{Ln},$
$\mathrm{int} : (x : \mathrm{Ln})(y : \mathrm{Ln})(\mathrm{Con}(x,y))\mathrm{Pt}.$

Lexical entries for common nouns look as follows. Notice that the sugaring rule must show the formation of both singular and plural forms. The operator REGNOUN forms the plural of a word C by adding the suffix -*s*.

$$\text{point} : \text{CN},$$
$$\text{point}^{\star} = \text{Pt},$$
$$\text{point}^{\circ}[n] = \text{REGNOUN}(n, \mathit{point}).$$

The common noun line interpreted as Ln is introduced in the same way.

To express distinctness in English, we have two lexical entries,

$$\text{distinctP} : \text{A4}(\text{Pt}, \text{from}, \text{Pt}),$$
$$\text{distinctP}^{\star} = \text{DiPt},$$
$$\text{distinctP}^{\circ} = \mathit{distinct},$$

$$\text{distinctL} : \text{A4}(\text{Ln}, \text{from}, \text{Ln}),$$
$$\text{distinctL}^{\star} = \text{DiLn},$$
$$\text{distinctL}^{\circ} = \mathit{distinct}.$$

Notice that both of the adjectives distinctP and distinctL are sugared to the same English word *distinct*, which thereby becomes overloaded. It is clear that no formal expression can be overloaded, but all overloading (as well as other cases of ambiguity) is an effect of sugaring.

Several syntactic categories corresponding to two-place predicates have basic expressions for convergence:

$$\text{intersect} : \text{V2}(\text{Ln}, \text{Ln}),$$
$$\text{converge} : \text{V4}(\text{Ln}, \text{with}, \text{Ln}),$$
$$\text{convergeC} : \text{V5}(\text{Ln}, \text{Ln}),$$
$$\text{convergent} : \text{A4}(\text{Ln}, \text{with}, \text{Ln}),$$
$$\text{convergentC} : \text{A5}(\text{Ln}, \text{Ln}).$$

To express the proposition

$$\text{Con}(l^{\star}, m^{\star}),$$

where $l, m : \text{PN}(\text{Ln})$, we have thus five expressions of the category S,

$$\text{SUBJ}(\text{Ln}, \text{RAISE}(\text{Ln}, l), \text{OBJ}(\text{Ln}, \text{Ln}, \text{intersect}, \text{RAISE}(\text{Ln}, m))),$$
$$\text{SUBJ}(\text{Ln}, \text{RAISE}(\text{Ln}, l),$$
$$\quad \text{OBJ}(\text{Ln}, \text{Ln}, \text{TRANS}(\text{Ln}, \text{with}, \text{Ln}, \text{converge}), \text{RAISE}(\text{Ln}, m))),$$
$$\text{COLL}(\text{Ln}, \text{Ln}, l, m, \text{convergeC}),$$
$$\text{SUBJ}(\text{Ln}, \text{RAISE}(\text{Ln}, l), \text{OBJ}(\text{Ln}, \text{Ln}, \text{TRANS}(\text{Ln}, \text{with}, \text{Ln},$$
$$\quad \text{COMPLADPRED}(\text{Ln}, \text{with}, \text{Ln}, \text{convergent})), \text{RAISE}(\text{Ln}, m))),$$
$$\text{COLL}(\text{Ln}, \text{Ln}, l, m, \text{COLLADPRED}(\text{Ln}, \text{Ln}, \text{convergentC})),$$

which sugar into the sentences

l° *intersects* m°,
l° *converges with* m°,
l° *and* m° *converge*,
l° *is convergent with* m°,
l° *and* m° *are convergent*,

respectively. Notice that the word *converge* results from both of the formal verbs converge and convergeC. These formal expressions have the same interpretation, but as they belong to different syntactic categories, they must be treated as distinct. The overloading of the word *convergent* is similar.

To finish the illustration of the lexicon, let us give entries for expressing the parallel construction and the intersection point.

$$\begin{aligned}
&\text{parallel} : \text{F2}(\text{to}, \text{Ln}, \text{through}, \text{Pt}, \text{Ln}),\\
&\text{parallel}^{\star} = \text{par},\\
&\text{parallel}^{\circ} = \textit{parallel},
\end{aligned}$$

$$\begin{aligned}
&\text{intersection-point} : \text{F4}(\text{Ln}, \text{Ln}, \text{Con}, \text{Pt}),\\
&\text{intersection-point}^{\star} = \text{int},\\
&\text{intersection-point}^{\circ} = \textit{intersection point.}
\end{aligned}$$

10 Logical Constants

A theory that is used as the basis of many further theories is monomorphic set theory, providing the definitions of the logical constants. The following expressions are interpreted in terms of the quantifiers Π and Σ.

$$\begin{aligned}
&\text{every} : \text{Q1},\\
&\text{every}^{\star} = \Pi,\\
&\text{every}^{\circ} = \textit{every},
\end{aligned}$$

$$\begin{aligned}
&\text{some} : \text{Q1},\\
&\text{some}^{\star} = \Sigma,\\
&\text{some}^{\circ} = \textit{some},
\end{aligned}$$

$$\begin{aligned}
&\text{if} : (A : \text{S})(B : \text{S})\text{S},\\
&\text{if}(A, B)^{\star} = \supset(A^{\star}, B^{\star}),\\
&\text{if}(A, B)^{\circ} = \textit{if}_A^{\circ}_B^{\circ},
\end{aligned}$$

$$\begin{aligned}
&\text{and} : (A : \text{S})(B : \text{S})\text{S},\\
&\text{and}(A, B)^{\star} = \&(A^{\star}, B^{\star}),\\
&\text{and}(A, B)^{\circ} = A^{\circ}_\textit{and}_B^{\circ},
\end{aligned}$$

where $\supset(A, B) = \Pi(A, (x)B)$ and $\&(A, B) = \Sigma(A, (x)B)$. Now we can form the sentence

$$\text{SUBJ}(\text{Ln}, \text{QUANT}(\text{every}, \text{line}), \text{OBJ}(\text{Ln}, \text{Ln}, \text{intersect}, \text{QUANT}(\text{some}, \text{line}))),$$

which is interpreted as the proposition

$$\Pi(\text{Ln}, (x)\Sigma(\text{Ln}, (y)\text{Con}(x, y)))$$

and sugared into the string

every line intersects some line.

Notice that we cannot yet form sentences to express all quantified propositions, but only propositions in which each quantifier binds just one occurrence of a variable, or none at all. If there is a variable, it must occur in a "subject or object position" in a formula and occupy an argument place all by itself.

To express propositions with other kinds of bindings, we could either introduce more combinators or include a mechanism for binding variables in the grammar. A solution of the former kind is to define the conjunction for syntactic categories other than S. Thus we have the conjunction of intransitive verbs,

V1 $\rightarrow$ V1 *and* V1.

$$\frac{A : \mathrm{dom} \quad F : \mathrm{V1}(A) \quad G : \mathrm{V1}(A)}{\begin{array}{l}\mathrm{CONJV1}(A, F, G) : \mathrm{V1}(A), \\ \mathrm{CONJV1}(A, F, G)^{\star} = (x)\&(F^{\star}(x), G^{\star}(x)), \\ \mathrm{CONJV1}(A, F, G)^{\circ}[n] = F^{\circ}[n]\_\mathit{and}\_G^{\circ}[n].\end{array}}$$

Assuming the verb contain : V2(Ln, Pt), we can now form the sentence

$$\begin{array}{l}\mathrm{SUBJ}(\mathrm{Ln}, \mathrm{QUANT}(\mathrm{every}, \mathrm{line}), \mathrm{CONJV1}(\mathrm{Ln}, \\ \qquad\qquad\qquad\qquad \mathrm{OBJ}(\mathrm{Ln}, \mathrm{Ln}, \mathrm{intersect}, \mathrm{QUANT}(\mathrm{some}, \mathrm{line})), \\ \qquad\qquad\qquad\qquad \mathrm{OBJ}(\mathrm{Ln}, \mathrm{Pt}, \mathrm{contain}, \mathrm{QUANT}(\mathrm{some}, \mathrm{point})))),\end{array}$$

which is interpreted as the proposition

$$\Pi(\mathrm{Ln}, (x)\&(\Sigma(\mathrm{Ln}, (y)\mathrm{Con}(x, y)), \Sigma(\mathrm{Pt}, (y)\mathrm{Inc}(y, x)))),$$

where Inc : (Pt)(Ln)prop is the incidence predicate, and sugared into the string

every line intersects some line and contains some point.

In the same way, we can define a conjunction in each of the categories corresponding to two-place predicates. In the next section, we shall give several examples of sentences built by conjoining verbs and adjectives. As we shall see, the general requirement for conjoining two verbs or adjectives is that they share both the syntactic category and the argument types—in other words, they must be expressions of the same domain-relative syntactic category.

11 Alternative Combinations

In the present system of combinators, it usually takes several steps to build a sentence (cf. the diagram at the end of Section 7). Thus for an adjective of category A4(r), we have to take four steps: prefix the copula, postfix the preposition, postfix the object, and prefix the subject. This analysis of the process is not made only for the sake of economy, to minimize the number of combinators needed, but also because all the intermediate stages produce expressions that occur as constituents of English sentences. For instance, the adjective

$$\mathrm{convergent} : \mathrm{A4}(\mathrm{Ln}, \mathrm{with}, \mathrm{Ln})$$

is conjoined with another adjective of the category A4(Ln, with, Ln) in the sentence

$$l \textit{ is } ((\textit{convergent}) \textit{ and } (\textit{coincident})) \textit{ with } m.$$

(We use parentheses for exposition, to indicate the relevant constituent structure.) At the next stage, V4(Ln, with, Ln), we have the verb *is convergent*, which occurs in conjunction with another verb of that category in the sentence

$$l\ ((\textit{is convergent}) \textit{ and } (\textit{coincides})) \textit{ with } m.$$

By adding the preposition *with*, we obtain a verb of the category V2(Ln, Ln) that sugars into the string *is convergent with*, which is conjoined with another transitive verb in the sentence

$$l\ ((\textit{is convergent with}) \textit{ and } (\textit{intersects}))\ m.$$

Finally, the intransitive verb phrase *is convergent with* m is conjoined with another phrase of the category V1(Ln) in the sentence

$$l\ ((\textit{is convergent with } m) \textit{ and } (\textit{intersects } k)).$$

With similar considerations—that is, by testing occurrence in conjunctions—we could find even more categories and expressions. By starting from A4(A, r, B) and first adding the preposition instead of the copula, we get expressions like *convergent with*, which occurs in the conjunction

$$l \textit{ is } ((\textit{convergent with}) \textit{ and } (\textit{distinct from}))\ m.$$

To recognize this expression, we introduce a new category A2(A, B) of two-place adjectives without prepositions. Now we can either add the copula and get to V2(A, B), or add an object and obtain the expression *convergent with* m of a new category A1(A) of one-place adjectives, occurring in

$$l \textit{ is } ((\textit{convergent with } m) \textit{ and } (\textit{vertical})).$$

From A1(A), we can get to V1(A) by adding the copula, as shown by the conjunction

$$l\ ((\textit{is convergent with } m) \textit{ and } (\textit{intersects } k)).$$

On the other hand, we can start with a two-place verb and first add the subject before adding the object. The phrase l *is convergent*, obtained from the adjective convergent : A4(Ln, with, Ln) by first adding the copula and then a subject, occurs in the conjunction

$$((l \textit{ is convergent}) \textit{ and } (m \textit{ coincides})) \textit{ with } k,$$

as well as in the relative construction

$$\textit{line with which } l \textit{ is convergent.}$$

To recognize phrases like this, we introduce a new category V1$^1/_2(r, B)$ of sentences lacking objects of type B with the preposition r. But we can also add a subject to the transitive verb phrase *is convergent with*, to get the phrase l *is convergent with*, which occurs in the conjunction

$$((l \textit{ is convergent with}) \textit{ and } (m \textit{ intersects}))\ k,$$

as well as in the relative construction

$$\textit{line that } l \textit{ is convergent with.}$$

We give this new category the name $\mathrm{V}^1/_2(B)$—sentences lacking objects of type B.

We can now complete the left branch of the diagram of Section 7 to a much richer diagram:

$$\begin{tikzcd}
 & & \mathrm{A4}(A,r,B) \arrow[dl, "is"'] \arrow[dr, "r"] & & \\
 & \mathrm{V4}(A,r,B) \arrow[dl, "a"'] \arrow[dr, "r"] & & \mathrm{A2}(A,B) \arrow[dl, "is"'] \arrow[dr, "b"] & \\
\mathrm{V1}^1/_2(r,B) \arrow[dr, "r"'] & & \mathrm{V2}(A,B) \arrow[dl, "a"'] \arrow[dr, "b"] & & \mathrm{A1}(A) \arrow[dl, "is"'] \\
 & \mathrm{V}^1/_2(B) \arrow[dr, "b"'] & & \mathrm{V1}(A) \arrow[dl, "a"'] & \\
 & & \mathrm{S} & &
\end{tikzcd}$$

At each arrow, we have indicated whether the item that is added is the preposition r, the copula *is*, a subject a of type A, or an object b of type B. The combinators are not shown. Nor have we introduced explicitly all of the combinators that are needed for the transitions in this diagram, but their definitions, as well as their interpretation and sugaring rules, should be obvious in the light of the foregoing discussion.

One consequence of the recognition of all these alternative modes of combination is syntactic ambiguity. Thus a sentence of the form

$$l^\circ \textit{ is convergent with } m^\circ$$

can be analysed in six different ways. If l and m are proper names, all these analyses have the same semantical interpretation,

$$\mathrm{Con}(l^\star, m^\star).$$

But if l and m are quantifier phrases, their relative scopes depend on whether l or m enters first in the combination. Thus the sentence

$$\textit{every line is convergent with some line}$$

receives two different interpretations,

$\Pi(\mathrm{Ln},(x)\Sigma(\mathrm{Ln},(y)\mathrm{Con}(x,y))),$
$\Sigma(\mathrm{Ln},(y)\Pi(\mathrm{Ln},(x)\mathrm{Con}(x,y))).$

Whether both of these interpretations are equally acceptable in practice can be debated. But it was one of the motivations in Montague's work to reveal such ambiguities, and make them explicit in syntax.

To summarize, one can say that the above diagram commutes with respect to sugaring but does not commute with respect to interpretation if the subject and the object are quantifier phrases.

The structure of the transitions between syntactic categories could not be discussed with semantic categories alone. The corresponding diagram in pure type theory looks simply as follows.

$$\begin{tikzcd}
& (A)(B)\mathrm{prop} \arrow[dl, "a"'] \arrow[dr, "b"] & \\
(B)\mathrm{prop} \arrow[dr, "b"'] & & (A)\mathrm{prop} \arrow[dl, "a"] \\
& \mathrm{prop} &
\end{tikzcd}$$

On the other hand, the transitions between syntactic categories also follow this typing discipline. They could thus not be discussed in phrase structure grammar alone.

12 Conclusion

We have been studying the structure of simple mathematical statements in English. It has turned out that both phrase structure grammar and type theory are needed. Then we have given an interpretation and generalization of phrase structure grammar in type theory—in other words, an extension of type theory with a system of categories of phrase structure grammar. We have ended up with a formalism that comprises both informal English (that is, strings of words), a system of syntactic categories and syntax trees, and a formal mathematical language. We have shown how syntax trees are sugared into English and interpreted in the mathematical formalism.

We have not shown how to define the inverses of sugaring and interpretation, which are by no means trivial. The inverse of sugaring, that is, parsing, was briefly discussed at the end of Section 6. The inverse of interpretation would be a function that takes type-theoretical formulae into syntax trees. It could be combined with the sugaring operation that we already have, to obtain a sugaring method for the mathematical formalism. We have previously tried to define sugaring directly for the mathematical formalism (see Ranta 1994a, chapter 9, and 1994b), but then we did not have access to other categorial structure than the structure of type theory.

Acknowledgements

I am indebted to Per Martin-Löf for discussions on grammatical structure, to Matti J. Kinnunen and Petri Mäenpää for cooperation in implementation, and to Jan von Plato for advice on the language of geometry. Thorsten Altenkirch and Thierry Coquand, as well as two anonymous referees, gave comments on the first version of the paper.

This paper was partly written during a period that I was employed by the Research Unit for Computational Linguistics at the University of Helsinki, and I am grateful to Prof. Fred Karlsson for the opportunity to concentrate on this work.

References

Yehoshua Bar-Hillel. On categorial and phrase structure grammars. *The Bulletin of the Research Council of Israel* 9F, pages 1–16, 1960. Reprinted in Y. Bar-Hillel, *Language and Information*, pages 99–115. Addison-Wesley Publishing Company and the Jerusalem Academic Press, Jerusalem, 1964.

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

Noam Chomsky. *Aspects of the Theory of Syntax.* The M.I.T. Press, Cambridge (Ma.), 1965.

Thierry Coquand. Pattern matching with dependent types. Informal proceedings of the Logical Frameworks Workshop held at Båstad in 1992.

Gottlob Frege. Begriffsschrift. Louis Nebert, Halle A/S, 1879.

Petri Mäenpää. Semantical BNF. Manuscript, University of Helsinki, 1994. To appear.

Lena Magnusson and Bengt Nordström. The ALF Proof Editor and Its Proof Engine. In H. Barendregt and T. Nipkow, editors, *Types for Proofs and Programs*, pages 213–237. *Lecture Notes in Computer Science* 806, Springer-Verlag, Heidelberg, 1994.

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

Richard Montague. *Formal Philosophy.* Yale University Press, New Haven, 1974. Collected papers edited by Richmond Thomason.

Jan von Plato. The axioms of constructive geometry. To appear in the *Annals of Pure and Applied Logic.*

Aarne Ranta. *Type Theoretical Grammar.* Oxford University Press, Oxford, 1994a.

Aarne Ranta. Type theory and the informal language of mathematics. In H. Barendregt and T. Nipkow, editors, *Types for Proofs and Programs*, pages 352–365. *Lecture Notes in Computer Science* 806, Springer-Verlag, Heidelberg, 1994b.

This article was processed using the LaTeX macro package with LLNCS style