The mathematical vernacular, a language for mathematics with typed sets

The Mathematical Vernacular, A Language for Mathematics with Typed Sets

N.G. de Bruijn

1. INTRODUCTION

1.1. The body of this paper is from an unpublished manuscript ("Formalizing the Mathematical Vernacular") that was started in 1980, had a more or less finished form in the summer of 1981, and a revision in July 1982. [*The Sections 1 to 17 were published for the first time in* [*de Bruijn 87a (F.3)*]. *At that occasion the (very essential) Sections 12–17 were revised in order to adapt them to typed set theory, and the Introduction was extended. For this 1994 version the old Sections 18–22 have been revised in order to let them match the revised Sections 12–17.*]

1.2. The word "vernacular" means the native language of the people, in contrast to the official, or the literary language (in older days in contrast to the latin of the church). In combination with the word "mathematical", the vernacular is taken to mean the very precise mixture of words and formulas used by mathematicians in their better moments, whereas the "official" mathematical language is taken to be some formal system that uses formulas only.

We shall use MV as abbreviation for "mathematical vernacular".

This MV obeys rules of grammar which are sometimes different from those of the "natural" languages, and, on the other hand, by no means contained in current formal systems.

1.3. It is quite conceivable that MV, or variations of it, can have an impact on computing science. A thing that comes at once into mind, is the use of MV as an intermediate language in "expert systems". Another possible use might be formal or informal specification language for computer programs.

1.4. Many people like to think that what really matters in mathematics is a formal system (usually embodying predicate calculus and Zermelo-Fraenkel set theory), and that everything else is loose informal talk *about* that system. Yet the current formal systems do no adequately describe how people actually think, and, moreover, do not quite match the goals we have in mathematical education. Therefore it is attractive to try to put a substantial part of the mathematical vernacular into the formal system. One can even try to discard the formal system altogether, making the vernacular so precise that its linguistic rules are sufficiently sound as a basis for mathematics.

An attempt to this effect will be made in this paper. We shall try to do more than just define what the formalized vernacular is: much of our effort (certainly in Sections 2, 3, 4) will go into showing its relation to standard mathematical practice.

1.5. Putting some kind of order in such a complex set of habits as the mathematical vernacular really is, will necessarily involve a number of quite arbitrary decisions. The first question is whether one should feel free to start afresh, rather than adopting all pieces of organization that have become more or less customary in the description of mathematics. We have not chosen for a system that is based on what many people seem to have learned to be the only reasonable basis of mathematics, viz. classical logic and Zermelo-Fraenkel set theory, with the doctrine that "everything is a set". Instead, we shall develop a system of typed set theory, and we postpone the decision to take or not to take the line of classical logic to a rather late stage.

1.6. The idea to develop MV arose from the wish to have an intermediate stage between ordinary mathematical presentation on the one hand, and fully coded presentation in Automath-like systems on the other hand. One can think of the MV texts being written by a mathematician who fully understands the subject, and the translation into Automath by someone who just knows the languages that are involved. [*For general information on Automath the following paper may be adequate:* [*de Bruijn 80 (A.5)*].]

Experience with teaching MV was acquired in a course "Language and Structure of Mathematics", given yearly since 1979 at the Eindhoven University of Technology. This course was designed on the claim that MV does not only serve to *present* mathematics better, but also to *understand* it better. The course is a part of the curriculum for mathematics teachers.

1.7. Even a superficial inspection of mathematical literature shows that it is very hard to get anywhere as long as we take the term "mathematical vernacular" so wide as to contain all language mathematicians use for convincing one another. We shall try to isolate a fragment of the language and polish it up so as to turn it into a basis for mathematics. It is this fragment that is called MV.

The rules of MV do not just explain how mathematical sentences have to be formed, but also how they have to be manipulated in order to build new correct material. In particular they will help us to disclose the rules of the game of axioms, definitions, theorems and proofs.

1.8. Roughly speaking, the MV part of a piece of mathematics will be the rigorous part. In order to make a bit clear at this stage what this MV part is, we mention a few things that we do *not* want to belong to it. Without being very systematic, we mention:

(i) Argumentation in the form of references to previous material, and indications of the kind of reasoning. Typical of what we mean here is: "Replacing x by p in Theorem 25 we find ...".

(ii) Indications for reconstructing pieces of texts that have been omitted. Example: "The second part of the proof can be given by interchanging the roles of x and y".

(iii) References to the syntactical form of presented material, like "the left-hand side of this equation".

(iv) Interpretation in terms of notions that belong to an entirely different area, like the use of geometrical terminology for discussing the graph of a function, in a case where the rigorous part of the text has no geometry at all.

(v) Remarks about the relation between the human writer or reader and the text. Example: "It is easy to see that ...". It may help the reader to draw a figure of this situation.

(vi) Commands, like "Show that ...".

(vii) Surveys of what is to be expected in later parts of the text.

(viii) Historical remarks.

It would not be hard to extend this list of non-MV items.

Quite often non-MV components and MV-components occur in one and the same sentence. Example: "Obviously we have $f(x) > 1$ for all x, but that does not help us to prove the lemma". The only MV-part here is "$f(x) > 1$ for all x".

1.9. In a system where we expect to have our mathematics checked by a machine it will certainly be worth while to take both the MV-part and the argumentations as essential parts of the formal language, as has been done in Automath. But even if that is considered as a sound basis for mathematical communication, it is questionable whether it can ever *replace* that communication. It has the disadvantage that it makes sense only for texts that have been elaborated in every silly little detail. For communication this is rather inconvenient. We wish to write in a style in which we omit what we think is trivial.

What things can be considered to be trivial depends on the experience the reader is expected to have. Therefore we shall define correctness of MV in such a way that proofs where pieces of the derivation are omitted, can be considered as still correct. A text would become incorrect if we omit definitions of notions that are used in later parts of the book.

A proof written in MV may be restricted to showing a sequence of resting points only. The derivation from point to point may be suppressed, or at least be treated quite informally. This seems to come close to the current ideal of mathematical presentation: impeccable statements, connected by suggestive remarks.

1.10. In contrast to what one might expect at first sight, the grammar of the mathematical vernacular is not harder, but very much easier than the one of natural language. We can get away with only three grammatical categories (the sentence, the substantive and the name), because mathematicians can take a point of view that is very different from the one of linguists. The main thing is that mathematical language allows mathematical notions to be *defined*; it can even define words and sentences. In choosing these new words and sentences we have almost absolute freedom, just like in mathematical notation. We hardly need linguistic rules for the formation of new words and new sentences. It usually pleases us to form them in accordance with natural language traditions, but it is neither necessary nor adequate to set linguistic rules for them.

1.11. The language definition of MV will be presented in two rounds.

In the first round we express the general framework of organization of mathematical texts. It is about books and lines, introduction of variables, assumptions, definitions, axioms and theorems. All this is condensed in the rules BR1–BR9 in Sections 9 and 10.

In the second round we get the rules about validity. These cover Sections 11–17.

These two rounds describe a language for mathematics. It would go too far to call them the foundation of mathematics. The language of mathematics allows us to write mathematical books, and in these books we can axiomatize the rest of what we call the foundation of mathematics. Part of that axiomatic basis might be considered as foundation of mathematics as a whole, but other sets of axioms just serve for particular mathematical theories. The dividing line between the two is traditional, not essential. Part of the axiomatic basis in the book may be of logical nature, and that part will certainly be considered to belong to the foundation.

Most of the validity rules of the second round have been put in that second round since they cannot be expressed in the books. In other words, they cannot be expressed in MV itself. But a very large part of what is called the foundation of mathematics can just be written in the books, more or less to our own taste. As examples we mention here: falsehood, negation, conjunction, disjunction, the law of the excluded middle, existential quantification, the empty set, the axioms for the system of natural numbers, the axiom of choice.

One might try to reduce the second round to an absolute minimum and to put as much as possible in the MV books. We have not gone that far, in most cases because it seems to be nicer to keep things together that belong together. In the case of Section 15 (rules for cartesian products) the reason to keep it in the second round may seem peculiar. It is just because of the fact that if we want to refer to elements (a, b) of the cartesian product of A and B, we would hate to have to mention A and B as parameters all the time. We would have to, if that section would be shifted to the book.

1.12. Let us try to compare MV and Automath. In the first place it must be said that MV has been inspired by the structure of Automath as well as by the tradition of writing in Automath. In that tradition elementhood, i.e. the fact that an object belongs to a set, is expressed by the typing mechanism available in Automath. So in order to say that p is an element of the set S, this is coded as $p : S$, so S is the type of p. This is in accordance with the tradition in standard mathematical language. If we say that p is a demisemitriangle, one does not think of the set or the class of all demisemitriangles in the first place, but rather thinks as "demisemitriangle" as the type of p. It says what kind of thing p is.

In order to keep this situation alive, MV does not take sets as the primitive vehicles for describing elementhood, but substantives (in the above example semidemitriangle is a substantive).

It is important to see the difference between *substantives* and *names*. Grammatically they play different roles. If we say that $a + b$ is an integer, then "integer" is a substantive and $a + b$ plays the role of the name of an object.

Coming to a situation like $a \in b \in c \in d$, the Automath style does not allow to write this as a chain of typings like $a : b : c : d$. If b is a set, then let us write $b\downarrow$ for the substantive "element of b". The chain becomes $a : b\downarrow$, $b : c\downarrow$, $c : d\downarrow$.

An important difference between Automath and MV is that in Automath typings are unique (up to definitional equivalence), and in MV they are not. MV is adapted to the tradition of ordinary mathematical language in which 5 is a real number and at the same time the same 5 is an integer. One does not feel a conflict since "integer" is just a special kind of "real number". In Automath it is always a bit troublesome to express that an object belongs to a subtype: The fact that 5 is a positive real number is described in Automath by two consecutive typings. The first one says that 5 is a real number, the second one says that some particular expression u is a proof for the statement that 5 is positive. This is often felt as a burden.

A consequence of the way we treat typing by means of substantives in MV is that a typing like "5 : real number" has the nature of a proposition. This is one of the rules of MV (see T1 in Section 12), but is not done in Automath.

Another difference between Automath and MV, already mentioned in 1.9, is that Automath has exact proof references inside the formal text, whereas MV either does not have them at all or has them informally in the margin. This provides a serious (but quite clear) task for those who implement MV into Automath.

There is another trouble with the implementation. In MV we have quite strong equality rules, more or less corresponding to the standard feeling that "between two equal things there is no difference at all, they are just the same". In Automath it may cause quite some work to show the equality of two expressions whose only difference is that, somewhere inside, the first one has p and the second one has q, and where p is equal to q. One has to bring the equality from the inside to the outside, and that may cause a lot of Automath text. Fortunately the writing of that text can be automated. In our version of MV we have a strong set of equality axioms (in particular EQ10a–EQ10c of Section 13.2) which make all this much easier.

1.13. One might think of direct machine verification of books written in MV, but this will be by no means so "trivial" as in Automath. Checking books in MV may require quite some amount of artificial intelligence. In the first place MV allows us to omit parts of proofs, at least as long as no definitions are suppressed (see Section 1.9).

But even if the steps in an MV book are ridiculously small, a checker may have a hard time, since in MV proof indications are not given in the formal text itself. To make a book in MV better readable, one can provide the text with proof references in the form of hints, so to speak in the margin. In order to make automatic checking of MV books feasible, one has to invent some system to pass those informal hints to the artificially intelligent machine.

1.14. The formation rules of MV allow us to form sub-substantives to a given substantive. The relation is denoted by $\ll$, like "square $\ll$ rectangle" in geometry. Once we have the substantive "rectangle", we can form smaller ones, but our rules do not allow us to form bigger ones. The effect is that for every object in an MV book one can find a "largest" substantive it is typed by. Let us call that one the *archetype* of the object. Likewise, this largest substantive can also be called the archetype of all the substantives it contains in the sense of $\ll$.

These archetypes can be on the back of our minds, but they are never mentioned explicitly in the MV book. Moreover, archetypes are nowhere mentioned in the language rules. One advantage of this system of "anonymous archetypes" is that we are never obliged to state the archetypes as a kind of parameters (actually this is what we have to do in Automath). Another advantage is that the MV text we produce can also be appreciated by readers who have bigger archetypes in mind. For example, a book on real numbers where complex numbers are never mentioned, can be used by anyone who started from the complex numbers, and wants to see the reals as special cases. In other words, our MV books can always be embedded into book with bigger archetypes.

Since all objects, all substantives and all sets have an archetype, we can refer to our MV as a kind of typed set theory.

1.15. Our effort in describing a large part of the language in terms of both substantives and sets, instead of sets only, gives some duplication in the language rules that might be considered as superfluous. We of course would like to try to eliminate one of the two, and deal with sets only, or with substantives only. Both can be done, of course, but none of the two seems to give anything that looks more satisfactory than what we have in our MV.

It has some advantage to describe both: it liberates us from the nasty decision to discard either of them. Substantives (like point, number, function) seem to be the things we handle in our natural language, and sets are things we have learned, more or less artificially, to use instead of substantives.

Before the advent of New Math (or should we say, before the rise and fall of New Math) talking by means of substantives was general, and set language was only introduced if strictly necessary. This is roughly what we have done in our presentation of the MV rules.

Talking and thinking in terms of substantives is so strongly traditional that one might even call it "natural".

1.16. In MV it is not true that "every object is a set". If we introduce a substantive as a variable or as a primitive, then the objects which are typed by that substantive can not be considered as sets. Elements of a cartesian product (see Section 15) are no sets.

1.17. Some of the decisions we have to take about MV involve questions about what to put in the language and what in the metalanguage. In particular we have a kind of meta-typing (the "high typing", see Section 3.6) in the language, whereas most other systems would have such things in the metalanguage. The high typing is used for saying that something is a substantive or that something is a statement. We note here that "statement" is a synonym for "sentence". We use "statement" in MV, but there would be no harm in replacing it by "proposition". Linguists would probably dislike the use of the word "sentence" for phrases which are no full sentences.

The distinction between high typing and low typing corresponds in Automath to the distinction between typing by means of expressions of degree 1 and typing by means of expressions of degree 2.

1.18. It is customary to make the distinction between sets and classes. Roughly speaking, sets are classes over which we allow quantification. Usually we think of the classes which are no sets as those which are just too big to be sets, like the class of all sets. In our MV we allow quantification over every substantive, and substantives directly correspond to sets. Classes over which we cannot quantify are not discussed in MV itself, neither by means of low typing nor by means of high typing. We can discuss them in the metalanguage: the class of all statements, the class of all substantives.

In Automath the class of all proofs for a given proposition is treated as a type. There is nothing of that kind in MV.

1.19. Let us devote some attention to the role played by adjectives. An adjective belongs to a substantive, and serves a double purpose: (i) to form a new substantive, and (ii) to form a new sentence. Example: Having the substantive "triangle", we can form the adjective "isosceles". With this one we can form the new substantive "isosceles triangle" as well as the new sentence "... is isosceles".

It may be just because of this double usage that mathematicians like to express things by means of adjectives. Many definitions in mathematics are in the form of the introduction of a new adjective.

A warning must be given: an adjective belongs to a substantive, but not automatically to the archetype of that substantive.

2. SOME TERMINOLOGY

2.1. Before we start explaining what MV really is, we say something about the terminology to be used in this paper.

We of course distinguish between the *language* MV and *texts* written in the language. Instead of "texts" we shall speak of "books". The language MV has to be defined by stating its *rules of grammar*, or, as we might say, its *language definition*.

A certain amount of usage of MV in an MV-book will depend on constructions and notions that are introduced in the language definition. This kind of usage will be called *primary* MV. All the rest of MV that is used somewhere in the book depends on terms and phrases that were chosen previously in that book. Such book material will be called *secondary MV*. We quote some examples. A phrase like "for all x" will conveniently belong to the language definition, but "$\log x$", "normal subgroup", will be things we prefer to define in some book. There are things on the borderline for which the language design has to choose between introduction in grammar or in book, and that choice might be a matter of taste. Examples of things about which one might hesitate are: use of the equality sign, elementary notions about logic, and the treatment of sets and functions.

2.2. A mathematics book contains what we called (in Section 1.8) an MV-fragment, and all the rest is non-MV. A part of the non-MV fragment consists of material that speaks *about* the MV-fragment. One might think of words and phrases like "left-hand side", "equation", "notation", "symbol", "formula", "substitution", "unknowns", "integration", "algebraic", and of the material mentioned in Section 1.8, (i) and (ii). We shall not study this kind of language systematically, but just vaguely refer to it as *metalanguage*.

It should be noted that the borderline between language and meta-language has shifted over the centuries. There was a time when "set", "function" were definitely metalanguage, and right now there are things on the threshold between language and metalanguage which might shift into the language in the next few decades. As such, one might suspect terms like "proposition", "condition", "proof", "algorithm".

Quite often a word occurs in two different languages, with related but different meanings. This has its reasons: it is always hard to find new words for new notions, and we like to have words with some suggestive power rather than new words that do not mean anything to us. For example, the word "algebra" is used in metalanguage to indicate a branch of mathematics, and in the language itself to denote a special kind of ring. And if we say "this system of three equations has two solutions", then "three" and "two" may be on different sides of the border.

In this paper we shall develop some new terminology. Some of it will be incorporated in the primary part of the language MV, some of it will be meta-MV. We have to be explicit about this, especially since we shall shift things into MV that are commonly considered metalanguage. In particular words like "substantive", "statement" will become (primary) MV.

2.3. Let us give a survey of the abbreviations for the various languages to be referred to in this paper. Quite often a sentence does not belong to any of these, simply because the sentence is intended to relate these languages to each other. But for separate words in this kind of discussion it may be possible to state more precisely to what language they belong. In many cases we shall indicate this, and we use the abbreviations OMV, gL, MV, pMV, sMV, mMV, smMV, imMV.

(i) OMV. This stands for "ordinary mathematical vernacular". This is the language today's mathematicians actually use when they want to be precise but not absolutely formal.

(ii) gL stands for "general language". This refers to words and phrases outside mathematics, in our case usually as a kind of meta-OMV. Since we do not suggest an exact definition of OMV, the boarderline between OMV and gL will be vague.

(iii) MV stands for the "stylized" form of OMV, and is to be described in this paper. This MV is a language that can be defined completely (see Sections 3–17).

(iv) pMV stands for primary MV, as explained in Section 2.1. Words and other constructs of pMV appear for the first time in the language definition. Authors of MV books do not have the right to deviate from pMV.

(v) sMV stands for secondary MV, as explained in Section 2.1. Words, symbols and phrases of sMV are chosen by the author of an MV-book.

(vi) mMV. This stands for meta-MV. We shall consider two kinds, smMV and imMV.

(vii) smMV stands for syntax-oriented meta-MV. It is the language we use for expressing the rules of MV.

(viii) imMV stands for interpretation-oriented meta-MV. It is the language we use for discussion of the relation between MV and its popular variant OMV. We use imMV for explaining how pieces of existing mathematics can be expressed in MV.

Many of the words we introduce in pMV, sMV, smMV will be borrowed from gL, and the usage will be related to their usage in gL.

In order to give an impression what distinctions can be made, we give a list of terms which all mean something like "group of words or symbols that express something", and in each case we indicate the possible languages. In several cases the set of possible languages might be extended, and in some cases it is not very clear what that list should be. What is important for us here is the question which cases are pMV, sMV, imMV, smMV. The matter of what belongs to gL is more vague, of course.

word		gL				
sentence		gL				
proposition	OMV			sMV		
statement			pMV			
substantive			pMV			
formula	OMV				smMV	
expression		gL			smMV	
phrase		gL				
theorem	OMV				smMV	imMV
assertion	OMV	gL			smMV	imMV
assumption	OMV	gL			smMV	imMV
condition	OMV	gL			smMV	imMV
definition	OMV	gL			smMV	imMV
predicate	OMV				smMV	imMV
clause					smMV	
name		gL			smMV	imMV

3. INGREDIENTS OF MV

3.1. We shall point out some of the characteristics of MV, with special emphasis on those aspects which might be considered novelties. We shall comment on the following points: context indication (Section 3.2), mixing natural language and formulas (Section 3.3), grammatical categories (Sections 3.4 and 3.5), typing (Section 3.6).

3.2. The context indication system, as described in Section 4, is little more than a systematic description of what all mathematicians are aware of when they are talking or writing mathematics. It is certainly worth while to give it a predominant place in the description of what mathematics is. In particular it gives insight in what "variables" are. Moreover it opens the way to natural deduction as a basis for mathematical reasoning.

3.3. Mixing natural language and formulas is a very typical aspect of a mathematician's lingo (both OMV and MV). In most cases formulas become part of a sentence as if they were just words or sequences of words, in complete accordance with the grammar of the natural language. When we say "if $a//b$ then

$p \in V$", then "$a//b$" and "$p \in V$" play grammatically the role of sub-sentences, just like in "if it rains then we get wet". And when we say "b satisfies ..." then b is the subject of the sentence just as if b were a person. But there are also cases where the mathematician's lingo does not follow the rules of the natural language. We mention "for all integers x we have ...", where the x does not play a role that any ordinary word can play in a natural language sentence. And we mention "for all x we have ...", where x does seem to play such a role, but the wrong one. For these little irritations we offer as explanation that our natural languages do not admit anything corresponding to bound variables.

3.4. In natural languages one analyses the structure of a sentence by attaching *grammatical categories* to words or word groups. Such categories can be "sentence", "noun", "verb", etc. In our discussion of MV we shall restrict ourselves to a rather small number of categories. We shall only use the following:

statement, *substantive*, *name*.

There might be a case for the *adjective* (cf. Section 1.19) as a fourth category, but we ignore them in our presentation of MV.

The reason why we do not need to go into the finer shades of grammatical analysis lies in the fact that in the MV-book we can introduce words, symbols and other kinds of phrases by means of *definitions* (see Section 7.2). As far as the defined things are words or phrases, we usually choose them according to what sounds right in ordinary language, and that is why they seem to ask for linguistic analysis. But such an analysis is unnecessary. The only thing we do with the new words and phrases is to *repeat them* in other circumstances. As an example we quote a definition: "We say that the vectors p and q are locally independent in the sense of Prlwtzkowsky if ...". Later we just repeat this phrase, with p and q replaced by other names. The fact that the words "in the sense of" have been taken just in this order, does not play a role we consider to be essential for MV. It plays a role in readability, memorizability and possibly in parsability (cf. Section 23). It is like choosing notation: as a function symbol for the hyperbolic cosine we might select "cos hyp" or "cosh" or "csh" but not "ggrrr", since that would not be very suggestive, and certainly not "gg?(rg[?" since that would be asking for trouble with parsing.

3.5. Let us say a few informal things (expressed in gL) on the categories "statement", "substantive", "name" and "adjective".

A *statement* is a group of words or formulas that might play the role of a complete sentence, although it can occur as just a part of some other phrase (the word "phrase" is used here to indicate any sequence of words or formulas that somehow is considered in its entirety at some moment). Example: In the phrase "if $a > b$ then p is divisible by 5" the parts "$a > b$" and "p is divisible by 5" are statements, and the whole phrase is a statement.

A *substantive* is a generic term for a class. Examples: "circle", "positive integer with exactly three divisors", "point". A generic term for a class is not the same as a name for that class. The difference is small: it is only the way we use them. If C is the class of all circles, then the phrases "P is a circle" and "P is an element of C" are intended to mean the same thing.

A warning: sometimes a phrase has the grammatical form of a substantive without playing that role in a mathematical text. In the phrase "P is the orthocenter of triangle ABC", the word "orthocenter" is not to be considered as a generic name for a class. One should not think that it had first been explained what an orthocenter is, and that later it was proved that a triangle has just one orthocenter, so that finally we can speak of "the orthocenter". No: the phrase "the orthocenter of triangle ABC" can be used by virtue of a previous definition in the book, where it was introduced as a name with the same status as a name like $\mathrm{Oc}(A, B, C)$ would have had. Therefore there is no question of parsing it into separate components like "the", "orthocenter", etc.

A *name* is a phrase we consider as a sufficient indication of an object. Without going into the question whether we have or do not have objects in mathematics, we note that our linguistic handling of mathematics seems to treat mathematical names as if they *were* names of objects. Examples of names are "the center of the unit circle", "the point M", "M", "$a + b$".

As to *adjectives* we mention that adjectives are always attached directly or indirectly to a substantive. Once we know what a triangle is we can say what it means that a triangle is "isosceles". It can be used in two ways: (i) in statements like "triangle ABC is isosceles", and (ii) in order to form the new substantive "isosceles triangle".

This humble role of adjective does not seem to suffice as a reason for taking them as building blocks in our rudimentary grammar of MV. Nevertheless there is a reason to take them seriously: mathematicians seem to like them so much. They seem to like definitions where a new notion is presented in the form of a new adjective.

We shall say more about adjectives in Section 22.14.

3.6. In our version of MV we use typing on two levels: *low typing* and *high typing*.

Low typing is used to express that some "object" is of a certain "kind" like "p is an integer". In MV we have a preference for writing a colon instead of "is a", so we write "p : integer". This colon is the notation for a kind of relation between "p" (which is grammatically a name) and "integer" (which is grammatically a substantive). In the metalanguage smMV we say that "p : integer" is a (low) typing.

High typing is a thing that in most other systems would be put into the metalanguage rather than into the language itself. We denote it by a double colon. On the right we have either "statement" or "substantive". Examples of high typings are "integer :: substantive", "$x > y$ :: statement".

We can as well say right here that low typings "$p : q$" will occur only in cases where the high typing "q :: substantive" has been established already. In this connection we mention that one might say in the metalanguage smMV that p is a name, or that p is the name of some q, but we do not express this in MV itself. We mention a question that often turns up among mathematicians: is 3 a number or is 3 the name of a number? We can agree to both alternatives, depending on the language we use. In MV we say "3 : number", but in smMV we say that 3 is a name, and more precisely that 3 is the name of a number.

For a moment we consider the word "object". There is the old philosophical question whether mathematical objects exist. Those who believe in the existence are called platonists. One might suspect that all mathematicians are platonists, even those who fiercely deny it. The matter is clear for those who consider it as their job to provide useful communication language for mathematicians: platonism is not right or wrong, platonism is irrelevant. At least it is irrelevant for matters of truth and falsehood of mathematical statements. It may be relevant for mathematical taste, but that is a personal matter anyway.

The most important thing to say about platonism is possibly that platonism is dangerous. It may seduce mathematicians in thinking that they can get away with incomplete definitions of objects since these objects exist anyway. And it might give the false suggestion that slightly different definitions of a mathematical object are not harmful since after all they refer to one and the same platonic object. Another danger of the idea of platonic existence is that many people find it hard to understand the meaning of existence in mathematics. The statement in OMV that "there exists a positive number whose square equals 9" has nothing to do with the platonic existence of the number 3.

We shall give a kind of linguistic interpretation to the word "object". We take it as a word used in smMV. If S is a substantive, and if we have in MV that $p : S$, then we might say in smMV that "p is the name of an object". Continuing in smMV, one might ask "what kind of object?". The answer to this will be in smMV that p is the name of an S, and in MV itself that $p : S$ (which expresses that p is an S). We of course have not expressed here what the word "object" means, but only how the word is used.

3.7. In the next section we will use the word *clause* (smMV). It will get its exact description in the language definition (from Section 6 onward), but we may as well say right here that a clause is either a typing or a statement. This cannot serve as a definition of the word "clause", however. We even note that "typing" and "statement" belong to different languages.

In order to give a preliminary idea, we say here that a clause will be either a high typing,

$$A :: \text{substantive}$$

or

$$P :: \text{statement}$$

or something of the form

$$P \tag{3.7.1}$$

in situations where

$$P :: \text{statement} \tag{3.7.2}$$

had already been recognized as a valid clause. The interpretation of (3.7.2) is "P is a well-formed statement", and the one of (3.7.1) is "P is true".

Note that in (3.7.1) and (3.7.2) P itself can be a low typing like "$a : A$", where "A :: substantive" has already been recognized as a valid clause. There will be cases where we establish that "$a : A$" is a well-formed statement, and there will be cases where we establish that "$a : A$" is true.

High typings will be different from low typings in the sense that they cannot be considered as statements. There will be no valid clauses of the form

$$(A :: \text{substantive}) :: \text{statement}$$
$$(P :: \text{statement}) :: \text{statement}$$

4. STRUCTURE OF MV BOOKS

4.1. In this section we give a first outline of what a *book* is. The following terms will be all smMV: "book", "line", "older", "younger", "context", "context item", "declarational", "assumptional", "body of a line", "context length", "empty context", "flag", "flagstaff", "flagstaff form", "flagless form", "block", "block opener", "nested blocks", "sub-block".

A *book* is a finite partially ordered set of *lines*. The order relation is called "older than" ("line p is older than line q" and "line q is younger than line p" are synonymous).

A *line* consists of two parts: a context and a body.

A *context* is a finite sequence of *context items*. There are two kinds of context items: *declarational items* and *assumptional items*. The sequence of context items may be empty, and in that case we speak of the *empty context.* In general, the number of items in the sequence is called the *length* of the context; the empty context has length zero. This is all we say here about context items; for further information we refer to Section 6.

We refer to Section 7 for a description of the *body* of a line; for the time being we do not need such a description.

4.2. We sketch the interpretation of the words introduced in Section 4.1 in terms of gL. A book is to be interpreted as any connected piece of mathematics that starts from scratch. Lines are primitive building blocks of books. One aspect of lines is that if we omit the last line of a book then it is still a book, but if we omit just a part of that line then it is no longer a book. Usually we think of a book as a linearly ordered set (i.e., a sequence of lines, and we were thinking that way when using the words "last line" in the previous sentence), where the first line is "the oldest" and the last line is "the youngest", but we need not go so far as to prescribe this linearity. The meaning of old and young is that younger material may make use of older material, but not the other way round. Since every finite partially ordered set can be put into a linear order that is consistent with the partial order, we see that the generalization from linear to partial is a very superficial one. Nevertheless, the presentation in a non-linear form may make a book easier to understand. In particular, if two pieces A and B are logically independent of each other, then this independence would be muffled if, only for the sake of typography, we would proclaim A to be older than B.

Saying that a book remains a book if we omit the last line (or in the case of non-linear order, if we omit a line that is not older than any other line in the book), means that it remains a book in the sense of syntactic structure; it need not be an interesting book.

An assumptional context item is to be interpreted as an assumption, like "assume $p > q$". A declarational context item is to be interpreted as the introduction of a variable of a specified type, like "let y be a real number". A context is to be interpreted as a sequence of such items, arranged in the order in which they were introduced. As an example (in OMV) we give a context of length 4: "let n be a positive integer, let S be a subset of the set of real numbers, assume that S has n elements, let s be an element of S".

The body of a line is interpreted as a piece of true information we provide in the considered context. As an example (in OMV) we give, with the above context of length 4, "if $n > 1$ then S contains an element different from s".

4.3. In this section we present examples of the structure of a book. Throughout Sub-sections 4.3, 4.4, 4.5 we think of a linearly ordered book. The examples are abbreviated, in the sense that context items are replaced by symbols $I_1, I_2, I_3, \dots$, line bodies by $b_1, b_2, b_3, \dots$. Contexts are represented as sequences of items separated by commas, and we write an asterisk between context and body of the line. Now a book can look like this:

$$\begin{array}{ll} I_1, I_2, I_3 & * \; b_1 \\ I_5, I_2 & * \; b_2 \\ I_6, I_7, I_8, I_9 & * \; b_3 \\ I_6, I_8, I_9 & * \; b_4 \end{array}$$

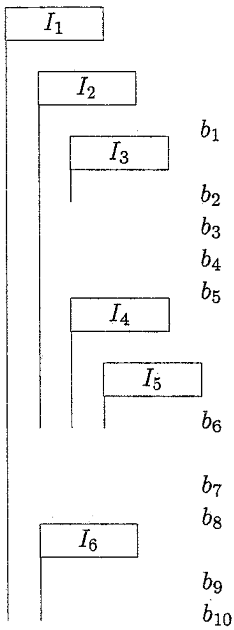
The contexts in this example look a bit untidy. In a mathematics text the contexts usually do not change from line to line, but are constant over a larger piece of text. And if the context changes, it is either by adding a few context items on the right or by deleting one or more from the right. So contexts grow and shrink on the right in the course of a mathematical discussion. Assumptions that were once introduced are no longer valid from a certain point onwards, and the same thing holds for variables: a variable is born, is alive during some time, and then dies. In OMV it is customary to announce birth of assumptions and of variables, but it is left to the reader to guess (possibly on the basis of the typographical layout, possibly on the basis of "understanding the author's intentions") at what point in the text they are dismissed.

For the sake of further discussion we give a typical example:

$$\begin{array}{lll} I_1, I_2 & * & b_1 \\ I_1, I_2, I_3 & * & b_2 \\ I_1, I_2 & * & b_3 \\ I_1, I_2 & * & b_4 \\ I_1, I_2 & * & b_5 \\ I_1, I_2, I_4, I_5 & * & b_6 \\ I_1 & * & b_7 \\ I_1 & * & b_8 \\ I_1, I_6 & * & b_9 \\ I_1, I_6 & * & b_{10} \end{array}$$

4.4. The information contained in a book is completely preserved if we write it in what we call its flagstaff form. In contrast to this, the form presented in Section 4.3 is called the *flagless form*.

In the *flagstaff* form, the context items are written on flags. The staff of a flag is vertical, and marks the set of lines where the flag's item is a part of the context. The following example, where the second example of Section 4.3 has been put into flagstaff form, speaks for itself.

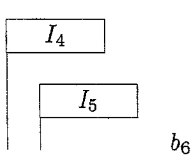


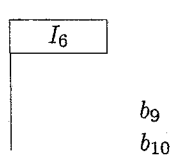
Needless to say, the way back from flagstaff form to flagless form is immediate. For every one of the bodies $b_1, ..., b_{10}$ we get the context if we assemble the items on the flags carried by the flagstaffs we see on the left of that body.

Later we shall use rectangular flags for assumptions and pointed flags for declarations, in order to make a clear distinction between those two kinds of context items. We did not do it here, since the relation between flagless form and flagstaff form is independent of such a distinction.

In our formal presentation of MV we handle the flagless form; in examples we may switch to the flags (see Section 18).

4.5. Sometimes we use the word *block* (smMV) to denote the material to the right of a flagstaff, including the flag itself. So every flag determines a block, and the item on the flag is called a *block opener* (smMV). As an example we quote that





are blocks of the book of Section 4.4. The block openers are I_4 and I_6, respectively.

The blocks are always *nested*, that is to say that if two blocks are not disjoint then one of the two is a sub-block of the other one.

5. IDENTIFIERS

5.1. In this section the following terms of smMV will be introduced: "identifier", "fresh identifier", "constant", "parametrized constant", "modified parametrized constant", "variable", "dummy", "variables of a context".

Note that a word like "variable" is smMV, but that the variables themselves are sMV. Similar things hold for the other notions.

5.2. An *identifier* is a symbol or a string of symbols to be considered as an atomic piece of text. We might say that an identifier is a symbol, but since we need a very large number of symbols we use strings of symbols instead, taken from a relatively small collection. It is a matter of parsing how to isolate these identifiers in a given piece of text. We shall not go into these parsing questions since they are not very essential here: if we had an unlimited amount of useful identifiers the matter would not have arisen at all. We refer to Section 23 for further remarks.

Examples of identifiers in OMV are "x", "2", "the complex number field", "parallelogram".

We note that if we describe "the complex number field" as a string of symbols, then we have to consider the empty spaces between the separate words as symbols too. These are produced by key strokes on a typewriter just like the letters, but they do not leave a visible imprint on the paper. Therefore it is better to replace the empty spaces by a visible character that is not used otherwise. One can take the underlining symbol for this, and write "the_complex_number_field". In our examples we shall not do this, however. It is one of the aspects in which the paper remains informal.

5.3. An identifier is called *fresh* at some specified place of the book if it has not appeared yet at older places of that book. In MV and in OMV we often need fresh identifiers, but in practice this is taken with a grain of salt. Since the number of short identifiers is rather small, we are inclined to use some of them repeatedly, in different circumstances, with different meanings. We shall not pay attention to this matter and act as if there were an unrestricted amount of easily recognizable symbols.

5.4. In an MV book there are various kinds of identifiers. First there are the pMV symbols that occur in the definition of MV, like

"substantive", "statement", ":", "::", ":=".

Another class of identifiers is the class of *variables* (the word "variable" is smMV, the variables in the book will be sMV). A variable is an identifier that occurs for the first time in an MV book in a declarational context item (see Section 6.3). Other identifiers are *bound variables* (also called *dummies*), for which we refer to Section 20.

Finally we have identifiers that are called *constants*. They are the identifiers whose first occurrence in an MV-book is on the left of a symbol ":=". The interpretation is that a constant is the name given to a defined object like "2", "e" (e is the basis of natural logarithms).

5.5. Related to the constants are the *parametrized constants*, which are not identifiers in the proper sense. A parametrized constant is a kind of finite sequence of symbols in which there occur variables at various places. The notion is relative with respect to a context.

A context has a number of variables, i.e., the variables introduced in the declarational items of that context. These variables will be referred to as the "*variables of the context*".

It is essential that each one of the context variables occurs at least once in the parametrized constant. The constants of Section 5.4 can be considered as parametrized constants for the case that there are no declarational items in the context.

If x and y are the variables of the context, then the following things may be parametrized constants:

"$f(x, y)$", "$x + y$", "the distance from x to y".

A parametrized constant is called *fresh* (smMV) somewhere in the book if it has not appeared at older places in that book, not even with different variables.

Parametrized constants can be used later in the book by repeating them, with the variables replaced by other expressions. We do not say here what kind of expression we have in mind, but just mention as examples

"$f(a + b, 3)$", "$(a + b) + 4$",
"the distance from P to the center of c".

In smMV such modified repetitions will be called *modified parametrized constants*. Clearly, these modified constants will generate new parsing problems, but again we lightheartedly neglect these.

The condition that in a parametrized constant *all* variables of the context occur, is usually taken with a grain of salt. We return to this in Section 21.6.

5.6. Many of the parametrized constants in our examples will have the form $b(x_1, ..., x_n)$, where $x_1, ..., x_n$ are all the variables of the context, in the order in which they are introduced in the declarational context items. In these cases we often take the liberty to write just b instead of $b(x_1, ..., x_n)$ on the left-hand side of the definitional line (and sometimes at other places where it is obvious what the abbreviation b stands for).

6. STRUCTURE OF CONTEXT ITEMS

6.1. A *context item* is a pair, consisting of a *clause* and a *label*. For a first orientation on what a clause is, we refer to Section 3.7. The label is either "(asm)" or "(dcl)". See Section 6.4 for the reason why these labels are used.

The phrases "context item", "clause" and "label" are smMV; both "(asm)" and "(dcl)" are pMV.

6.2. An assumptional context item has the form "P (asm)". As a first orientation we say that this P is a clause, but that not every clause will be admitted: "P (asm)" will only be allowed in cases where the high typing "P :: statement" can be established in the book, at least in the context formed by the sequence of context items preceding this item "P (asm)". For details we refer to Section 9.

6.3. Declarational context items have one of the following forms:

$x : P$	(dcl)
x :: substantive	(dcl)
x :: statement	(dcl)

where x is a fresh identifier and P is some expression. As said in Section 5.4, x is called the variable of the context item. Not every P will be admitted here, but only those P for which the high typing "P :: substantive" can be established in the book, in the context formed by the sequence of context items preceding this item "$x : P$ (decl)". For details we refer to Section 9.

6.4. It is essential that context items are explicitly labeled as being either declarational or assumptional. In the flagstaff form this can be done by using pointed flags for declarations and rectangular flags for assumptions (see Section 18).

The reason for the use of labels that distinguish between the two kinds of items, is the fact that the form of the context item does not always reveal to which one of the two categories it belongs.

The following example of a context of length 2 in OMV shows what we mean: "Let p be a quadrilateral, assume that p is a rectangle" (of course no one would say this in one breath, but quite often the various items of a single context are pages apart). The labels "let be" and "assume that" are no luxury, for if we would say "p is a quadrilateral, p is a rectangle" then it would not have been made clear that in the first item p is introduced and that in the second item p is a thing we already know about.

7. STRUCTURE OF LINE BODIES

7.1. There are four kinds of line bodies:

(i) definitional line bodies (Sections 7.2 and 7.6)

(ii) primitive line bodies (Sections 7.3 and 7.10)

(iii) assertional line bodies (Sections 7.4 and 7.11)

(iv) axiomatic line bodies (Sections 7.5 and 7.12)

(all these terms are smMV).

7.2. The interpretation in OMV of lines of the type (i) is that they represent definitions. That word has to be taken in a wide sense, and contains much more than what a text in OMV would label as "Definition". Whenever we select a new symbol to represent a longer expression, usually for the sake of brevity, we essentially have a definition.

We consider three kinds of definitions, according to the syntactic category of the things to be defined. There are "name definitions", where a new name is introduced for an "object". Next there are "substantive definitions", where a new substantive is introduced, and finally "statement definitions", where a new phrase is coined to represent a statement. As examples of the three kinds of definitions we quote

"the orthocenter of triangle t is ...",
"A rhombus is ...",
"We say that the sequence s converges to the real number c if ...".

In MV these three categories will be represented by (7.6.1), (7.6.2), (7.6.3), respectively. For further examples and comments see Sections 7.7–7.9.

Many definitions in OMV have the form of the introduction of a new *adjective*. We shall not put these into MV since they can be circumvented (cf. Sections 3.5 and 22.14).

7.3. The interpretation (in OMV) of lines with bodies of the type (ii) is that they introduce primitive notions. Such lines are rare in mathematics, and have the same status as axioms. Together with the axioms they may form the basis of a theory. As an example we quote from Hilbert's axioms for plane geometry, which state "there are things we call points and things we call lines", where the words "point" and "line" are introduced as new substantives but, in contrast to the substantive definitions of Section 7.6, without explanation in terms of known things. In MV the introductions of these primitives get the form (7.10.2).

An example of a primitive of the form (7.10.3) is that, after points and lines have been mentioned, the notion "point A lies on line q" is introduced without explanation.

Finally we give an example of what is expressed in (7.10.1). One of Peano's axioms is: "there is a special natural number which we shall denote by the symbol 1". Here the new object is introduced without definition. Instead of defining it we just say of what kind it is.

7.4. The interpretation of lines with bodies of the type (iii) is that assertions are made that follow from previous material. Some of these are called "theorems", others "lemmas", but most of them (in particular the assertions inside proofs) do not get such a stately name. And it is certainly not common practice to apply words like "theorem", "lemma" to cases with high typings like "A :: substantive", "P :: statement", which are likewise admitted here (see Section 7.11).

When saying that theorems and lemmas follow from previous material, we have to interpret the habit in OMV to print a proof after the announcement of the theorem instead of before. If we wish to have a similar announcement in MV, we might give a name to the thing stated in the theorem, claiming that it is a statement P, like in (3.7.2). The proof will end with the assertional line body P (see (3.7.1)). The interpretation of the first line in OMV is "P is a well-formed proposition", and of the last one "P is true".

7.5. Lines with a body of the type (iv) are to be interpreted as *axioms.* They can be applied in the same way as theorems, but in the case of axioms we do not require that their content follows from previous material.

7.6. In MV, a *definitional line body* has one of the forms

$$P := Q : R \tag{7.6.1}$$
$$P := Q :: \text{substantive} \tag{7.6.2}$$
$$P := Q :: \text{statement} \tag{7.6.3}$$

where P stands for a parametrized constant. Note that "substantive" and "statement" are pMV, as well as the symbols ":=", ":", "::", but that Q is definitely *not* the pMV symbol "PN".

7.7. We remark that it will be a consequence of our later rules that (7.6.1) appears only in situations where "R :: substantive" is valid.

The interpretation of (7.6.1) is that the definition provides a new (possibly short) name P for an object of which the full description is Q. Example (in OMV): "Let $S(n)$ denote the real number $\exp(1) + ... + \exp(n)$". In this example $S(n)$ plays the role of P, "$\exp(1) + ... + \exp(n)$" the role of Q, and "real number" the role of R. In MV we write it as "$S(n)$:= ... : real number" (we do not attempt right now to write $\exp(1) + ... + \exp(n)$ in official MV).

7.8. The interpretation of (7.6.2) is that it provides a (usually short) expression P for a (usually longer) description Q that represents a substantive. Example: the role of Q can be played by the substantive "positive integer with exactly two divisors" and the role of P by the new substantive "prime number".

7.9. The interpretation of (7.6.3) is that the definition provides a new (usually short) expression for a (usually longer) statement. Example: "We say that p is orthogonal to q if the inner product of p and q is zero". Here "p is orthogonal to q" plays the role of P, and "the inner product of p and q is zero" plays the role of Q.

7.10. In MV a *primitive line body* has one of the forms

$$P := \text{PN} : R \tag{7.10.1}$$
$$P := \text{PN} :: \text{substantive} \tag{7.10.2}$$
$$P := \text{PN} :: \text{statement} \tag{7.10.3}$$

We note that ":=", "PN", ":", "::", "substantive" and "statement" are all pMV ("PN" has been chosen at mnemonic for the OMV-term "primitive notion"). P stands for a parametrized constant. In the case of (7.10.1) it will be required that "R :: substantive" is valid in the context in which (7.10.1) is written.

7.11. An *assertional line body* in MV is nothing but a single clause (cf. Section 3.7).

7.12. An *axiomatic line body* has the form

$$c \text{ [Axiom]} \tag{7.12.1}$$

where c is a clause, and the symbol "[Axiom]" is a pMV term. By virtue of language rules still to be formulated, there are two differences between axiomatic and assertional line bodies. In the first place, the assertional line body has to "follow" from the previous part of the book, and secondly, in the axiomatic case the clause c has to be restricted to cases for which the high typing "c :: statement" can be established in the book. The latter is similar to the restriction made on assumptional context items (Section 6.2).

7.13. We introduce the notion *clause of a line body* (smMV).

In the cases of Sections 7.10–7.12 the body has just one clause. In an assertional line body that clause is the line body itself. In lines with axiomatic line body "c [Axiom]" the clause of the line is just that c. In lines with bodies (7.10.1), (7.10.2), (7.10.3) the clause of the body is "$P : R$", "P :: substantive", "P :: statement", respectively.

A definitional line body has *two* clauses. The *old clauses* of the lines of Section 7.6 are "$Q : R$", "Q :: substantive" and "Q :: statement", respectively. The *new clauses* of these lines are "$P : Q$", "P :: substantive" and "P :: statement", respectively.

8. GENERAL REMARKS ON RULES OF MV

8.1. In Section 4, 6 and 7 we have explained the structure of books, contexts and lines. The question is now: what contexts and what lines are allowed? It will not be trivial to state a complete set of rules for this. A part of these rules will be felt as rules for language manipulation; these rules will be explained in Sections 9 and 10. Another part (Sections 12–17) will be more like a piece of the foundation of mathematics. However, the rules of MV will not contain *all* of what is usually called the foundation of mathematics. Once we have reached a certain level, the language is strong enough to allow us to write the rest of the foundation of mathematics in an MV book.

It is attractive to put as little as possible in the language definition and as much as possible in the books, but we shall not aim at extremes in this respect.

The state of affairs can be compared to the way a ship is built. The ship is constructed ashore only until the stage that it is just able to float. Then it is launched, and after that, the construction goes on. The reason for this is, of course, that a ship cannot be launched if it is too heavy. In the case of MV the reason is different. The MV ship can be used by many different customers in different ways. After MV is launched, every customer can finish the construction according to his own wishes.

After the launching of a ship, two things happen: (i) the construction is completed, and (ii) the ship will be sailing the seas. Here our analogy is less satisfactory. The action of the ship's construction in the water near the shipyard is very different from the action of sailing the seas. In the case of MV these two actions are alike. To (i) there corresponds the writing of the fundamental portions of the book, and what corresponds to (ii) is writing a (possibly long) book or set of books based on that fundamental chapter. But all the time the action consists of writing books and nothing else.

8.2. As said before, our MV will be modelled after OMV, i.e. the way mathematicians write and speak today, but we cannot just copy OMV. There is no consensus in OMV about how things should be said. We are not in a position to derive all rules of MV by observation of OMV. We have to invent new rules, and that may mean making arbitrary choices. We have to give definitive shape to things which are not properly revealed in OMV. In particular this refers to the fact that this paper tries to interpret OMV as a typed language. One might argue that such an interpretation is not really called for, and that it is about as arbitrary as interpreting OMV as a non-typed language.

The most-favored method of coping with life without types is to maintain that "everything is a set". One might try to arrange a typed language in such a way that this set-loving point of view can be obtained by just creating a single type, viz. the type "set". Yet we have not taken the trouble to keep this possibility open in our presentation of MV.

Conversely, one might try to code typed material in terms of a non-typed language, but this seems to be very unattractive.

8.3. We first say something about the notion of *validity* (smMV). The word *valid* (smMV) means: having been built according to the grammar of MV (and that grammar has still to be disclosed). The rules of that grammar will be production rules, in the sense that they all describe ways to extend a valid book by adding a new line. In the course of the description of how the new line is to be built, we have certain resting-points where certain phrases are discussed as being acceptable ingredients of the line to be added. Important resting points are clauses (see Section 3.7 and Section 7.13). In a given context there is a set of such clauses which are called *valid* clauses. The production rules explain how our knowledge about that set can be extended, describing how by means of a number of elements of that set a new one can be constructed.

8.4. Validity is expressed with respect to a book.

As already said in Section 4, a book is a partially ordered set of lines. For any given line we can consider the set of all lines which are older than the given line; this is again a partially ordered set of lines and therefore a book. We shall refer to the given line as "the new line" and to that book as "the set of old lines". Whether a new line will be called valid, depends on the set of old lines, and not on what happens in other lines. The same remark applies to *parts* of new lines, like clauses and contexts. Only for the identifiers, and more generally for the parametrized constants we have a condition that goes beyond the set of old lines: we have to stipulate that they are all different throughout the whole book. We usually think of a book as having been written line by line, where older lines precede newer lines in time. If this is the case, then the condition for the parametrized constants is that at each moment the parametrized constant introduced in a line (on the left of a sign :=) is different from all the parametrized constants used before.

We still have to say how the context for the new line has to be built, and what clauses are valid in that context. This will be said in Sections 9 and 10.

8.5. From Section 9 onwards we shall give a formal definition of the notion of an MV book in flagless form. Except for the syntactic matters referred to in Section 8.6, we shall not make use of what was said in the preceding sections. Those sections were intended to give interpretations, and to help the reader to get an insight into the complex set of definitions we shall display in the next sections.

First a few things about the terminology. The smMV-terms "MV book" and "valid book" are synonymous. We shall not define notions "context" and "clause" as such, but we shall define "valid context with respect to a set of lines", "valid clause with respect to a valid context and a set of lines" (in both cases that set is referred to as the set of "old lines", and in the second case it will be required that the valid context is a valid context with respect to that same set of old lines), "valid book", "line", "body of a line", "clause of the body of a line", "context of a line".

The pMV symbols to be used are

"$:$", "$::$", "$:=$", "substantive", "statement", "PN", "[Axiom]"

and

"(dcl)", "(asm)", "$*$".

(The symbols of the second row do not appear in the flagstaff form: their role is taken over by the pointed and rectangular flags and flagstaffs.)

A few general things can be said here about the format of things. A book is a finite (possibly empty) partially ordered set of lines. A line is a pair consisting of a (valid) context and a line body. A (valid) context is a finite (possibly empty) sequence of context items.

8.6. We mention some things that should have been formally discussed in the next sections, but are nevertheless treated very superficially. They are of a syntactical nature. We mention:

(i) substitution,

(ii) variables of a context,

(iii) fresh identities and fresh parametrized constants,

(iv) parsing.

We take it that Section 5 (and 23) are sufficiently clear as an indication of how these notions are to be formalized. A complete formalized treatment of them would not quite fit into the general style of this paper.

9. VALID CONTEXTS AND VALID CLAUSES

9.1. Everything that has been said thus far is to be considered as introduction, providing an orientation about what we are going to describe. It also served to build up a feeling for the interpretation.

From now on, however, we shall attempt a more complete and more formal description. Many things that have been referred to earlier in vague terms, will now get a more serious treatment.

The rules are about books, lines and validity. They will get their content by means of rules BR1–BR9 (BR stands for "basic rule"). We need not say beforehand what these notions mean.

These rules BR1 to BR9 are hardly of a logical or a mathematical nature. Or, rather, they describe how to *handle* logic and mathematics. In order to get to logic and mathematics themselves we have to add a number of rules in Sections 12–17 that describe more ways to produce valid clauses. As to the production of valid contexts and books no rules will be issued beyond these BR1–BR9.

9.2. The symbols c, C, I_1, I_n, P, A, x, x_1, x_k, X_1, X_k that are used in this section for explaining language rules are meta-variables. They are used in smMV in order to denote expressions occurring in an MV book.

In the rules BR1–BR7 there is a set S of lines ("the set of old lines"), and "valid" stands for "valid with respect to S".

9.3. BR1. If an old line has context C, and if c is a clause of the body of that line, then C is a valid context, and c is a valid clause in that context.

9.4. BR2. The empty context is valid.

9.5. BR3. If $I_1, ..., I_n$ is a valid context (if $n = 0$ we mean the empty context), and if

P :: statement

is a valid clause in that context, then $I_1, ..., I_n, I_{n+1}$ is a valid context, where I_{n+1} stands for "P(asm)". (As already explained in Section 6.3, the additional "(asm)" serves to label it as an assumptional context item; it is not superfluous since P may have the form of a typing.)

9.6. BR4. If $I_1, ..., I_n$ is a valid context (if $n = 0$ it is the empty context), and if x is a fresh identifier, then the following contexts of length $n + 1$

$I_1, ..., I_n, x$ :: substantive (dcl)
$I_1, ..., I_n, x$ :: statement (dcl)

are valid contexts. If, moreover,

A :: substantive

is a valid clause in the context $I_1, ..., I_n$, then

$I_1, ..., I_n, x : A$ (dcl)

is a valid context.

9.7. BR5. If $I_1, ..., I_n$ is a valid context, and if one of these n items is $x : A$ (dcl), then $x : A$ is a valid clause in that context. Similarly, if one of the n items is x :: statement (dcl), then x :: statement is a valid clause in the context. If one of the items is P(asm), then P is a valid clause in the context.

9.8. BR6. Let C and C_0 be valid contexts, let $x_1, ..., x_k$ be the variables of the context C_0 (this notion was explained informally in Section 5.5), and let c be a valid clause in the context C_0. Let $X_1, ..., X_k$ be expressions with the property that if we replace $x_1, ..., x_k$ by $X_1, ..., X_k$, then all context items of C_0, with the labels "(dcl)" and "(asm)" deleted, become clauses which are valid in the context C. Then the clause we get if we replace x's by X's in c becomes a clause that is valid in the context C.

9.9. BR7. If $I_1, ..., I_n$ is a valid context, and if $k < n$, then $I_1, ..., I_k$ is a valid context. If c is a valid clause in the latter context, then c is a valid clause in the context $I_1, ..., I_n$.

10. VALID BOOKS

10.1. The notion of a valid book is obtained by saying that the empty book is valid and by explaining how a valid book can be extended.

10.2. BR8. The empty book is valid.

10.3. BR9. Consider a valid book, and take any set of lines as set of old lines. Let C be a valid context with respect to this set. The following list indicates what line bodies can be used to form, together with the context C, a line that produces a valid book again if it is added to the book, making the new line younger than all the old lines. The line bodies are on the right. In the cases (iii), (iv), (v), (vi), (vii), (viii) we require, as an extra condition, that the clause on the left is valid in the context C with respect to the set of old lines.

(i)	----	P := PN :: statement
(ii)	----	P := PN :: substantive
(iii)	Q :: statement	P := Q :: statement
(iv)	Q :: substantive	P := Q :: substantive
(v)	c :: statement	c [Axiom]
(vi)	c	c
(vii)	R :: substantive	P := PN : R
(viii)	Q : R	P := Q : R

In all cases P stands for some fresh parametrized constant, containing the variables of the context and no others.

As the "clause of the line body" we take, in the cases (i) to (vii), respectively,

(i) P :: statement

(ii) P :: substantive

(iii) both P :: statement and Q :: statement

(iv) both P :: substantive and Q :: substantive

(v) c

(vi) c

(vii) P : R

(viii) both P : R and Q : R.

10.4. We have a comment on case (ii) of BR9. Some people may say it is not customary to use, or to admit the use of, lines like this in any arbitrary context. They might like to admit them in the empty context only. Essentially this comes down to starting a mathematics book with the creation of a number of types, and then off we go. This restricted use of case (ii) has the advantage that it becomes much easier to describe the collection of all types that can occur in a book.

Nevertheless we keep this rule (ii) as it stands, i.e. we allow it in any context. We leave it to the user of the language to make or not to make the more restricted use of the rule. It is as with roads: one can build a road that technically admits speeds of 200 mph, the legal authorities may prescribe a speed limit of 100, and the individual user may restrict himself to a maximum of 60.

We note that if a substantive is introduced as primitive by means of a line of the type (ii), then this substantive is an archetype (see Section 12.1). And if the line has a context like the line

$$x : A(\text{dcl}) * P(x) := \text{PN} :: \text{substantive},$$

then the only way to make special instances $P(u)$ and $P(v)$ comparable, is to require $u = v$.

11. COMMON STRUCTURE OF FURTHER RULES

11.1. All further rules are about the validity of clauses, where "validity" is taken with respect to a set of old lines and with respect to a context (which in its turn is assumed to be valid with respect to that set of old lines). In the simplest case such a rule will be of the form

....

$$P \qquad\qquad Q \tag{11.1}$$

....

and will express the following: if P is a valid clause, then Q is a valid clause too. A variation on the scheme (11.1) is

....

P_1	Q_1
P_2	Q_2
	Q_3

....

which means to express the rule that if P_1 and P_2 are both valid, then Q_1, Q_2, and Q_3 are valid.

11.2. Some of the rules will be slightly more intricate in the sense that they deal with context extension. This can happen in entries on either side. We take a case where it happens on the left only:

....

$$\begin{array}{lcl} & P_1 & \\ J * & P_2 & \qquad\qquad\qquad Q \\ & P_3 & \end{array}$$

....

(the $*$ is an smMV symbol here). The meaning of this is as follows. We are dealing with a set of old lines (which is not going to be changed in this rule), and a context C. Assume that P_1 is a valid clause in the context C, that P_2 is a valid clause in the extended context C, J (if $C = I_1, ..., I_n$, then C, J represents the context $I_1, ..., I_n, J$), and finally that P_3 is a valid clause in the context C. Then Q is a valid clause in the context C.

The validity of J as a context item will not be open to doubt in the cases we present. This validity will always follow from the assumptions.

In rule T6 there is a case where the role of J will be taken over by two context items (separated by a comma) instead of a single one.

As remarked in Section 11.1, a rule like the one above is intended to hold in any context. If I is such a context, this means that the rule also includes the following one:

....

$$\begin{array}{rcl} I * P_1 & & \\ I, J * P_2 & \qquad\qquad\qquad & I * Q \\ I * P_3 & & \end{array}$$

....

11.3. In all our rules, the phrases that were represented above by P, P_1, P_2, P_3, Q, Q_1, Q_2, Q_3, J will be expressions in terms of one or more meta-variables. Actually all symbols that have not been introduced explicitly as pMV, are to be considered as meta-variables in this kind of rules. For example, in rule T8* the letters A and B are meta-variables. In applications of that rule, they may be replaced by any pair of expressions.

11.4. Except for rule EQ11, all rules to be presented in the next sections have the form sketched in Sub-sections 11.1 and 11.2.

11.5. We sometimes use the term *derived rule* (smMV). Derived rules are rules whose validity follows from earlier rules. In other words, what such a rule proclaims to be valid, can be shown to be valid already because of the other rules. We shall use a rule number with asterisk if we claim that the rule is a derived rule. The remaining rules are called *fundamental rules*, although we do not claim our set of fundamental rules to be minimal.

In some cases one might be able to write such derived rules as theorems in an MV book, and then it is a matter of taste whether we present them as language rules or as theorems.

There are derived rules whose derivation requires induction over the length of the book. As an example we take the observation that $a : A$ can appear in the book only when A :: substantive. Another example is the observation that if A and B are substantives, and $(A =)B$ :: statement, then there is a substantive C such that both $A \ll C$ and $B \ll C$. We shall not actually use this kind of derived rules, but rather consider them as part of the metatheory of MV books. Therefore we shall refer to such rules as metatheory rules.

11.6. In some rules we deal with *substitution* (smMV). We use as smMV notation $[[\ /\ ]]$. If x is an identifier, if P and Q are expressions, then $[[x/P]]\,Q$ denotes the expression we get if every occurrence of x in Q is replaced by the expression P. Since P may also contain x, there may arise new occurrences of x, but these new ones are not to be replaced by P, of course. Example: $[[x/g(x,y)]]\,f(x,y)$ stands for $f(g(x,y),y)$.

12. RULES ABOUT TYPING

12.1. We start with an informal introduction on the relation "$\ll$", a relation that can hold between substantives. If A and B are substantives then "$A \ll B$" is to be interpreted as "every A is a B". Example: "square $\ll$ rectangle", "rectangle $\ll$ rectangle". In smMV we say that "$A \ll B$" is a clause, and that it intends to express that "A is a *sub-substantive* of B". The clause "$A \ll B$" can appear in books also in cases where $A \ll B$ is not *true*. For example, although "rectangle $\ll$ square" is not true, it can still be considered as a well-formed statement. Here we speak smMV, but in MV it would be expressed as

rectangle $\ll$ square :: statement.

Our rules will have the effect that $A \ll B$ can only appear in our books if A and B have a common ancestor E, i.e., a substantive E such that $A \ll E$ and $B \ll E$ are both true. If there is not such a common ancestor, then $A \ll B$ will not be a statement. Our production rules for valid clauses will never produce such a thing as "rectangle $\ll$ complex number :: statement".

One might derive in the metalanguage that "$A \ll B$ :: statement" is an equivalence relation on the set of all substantives, and that every equivalence class is completely characterized by an "*archetype*" E, i.e., a substantive E with the property that all A of the class are sub-substantives of E. Similarly, if an "object" x is typed by $x : A$, then x has an archetype E.

These archetypes do not appear in our rules, and neither in our MV books. For a comment on why the rules of MV were designed without explicit archetypes we refer to Section 1.14.

Since formula $A \ll B$ will not be a statement for arbitrary substantives A and B, we will be unable to state it as an assumption in an MV book. We cannot say in such a book: "let A and B be substantives and assume that $A \ll B$". The fact that we do say such things in some of our rules (like in T4) is quite a different matter. These rules are not written in an MV book. In T4 it means the assumption "$A \ll B$ is a valid clause", and this is said in smMV, not in MV.

In none of our rules there is a conclusion drawn from the mere fact that $A \ll B$ :: statement. We refrain from such rules in the philosophy that in such cases we will always have some substantive C with $A \ll C$ and $B \ll C$.

Formulas $a : A$ will only appear in our books in situations where A is a substantive. Likewise, $A \ll B$ will only appear when both A and B are substantives. Therefore we need not add assumptions like A :: substantive on the left in the rules T1–T6.

As to use of the substantive binder S in rule T6 we refer to Section 20.4.

12.2. Rules T1–T11. (See Section 11 for notational conventions.)

....

T1	$a : A$	$a : A$ :: statement

....

T2	$A \ll C$ $B \ll C$	$A \ll B$ :: statement

....

T3	$A \ll C$ $B \ll C$ $a : A$ :: statement	$a : B$ :: statement

....

T4	$A \ll B$	$x : A(\text{dcl}) * x : B$

....

T5	$x : A(\text{dcl}) * x : B$	$A \ll B$

....

T6	$x : A(\text{dcl}) * P$ :: statement	$S_{x:A}\, P$ :: substantive $S_{x:A}\, P \ll A$ $y : S_{x:A}\, P(\text{dcl}) * [[x/y]]\, P$ $y : A(\text{dcl}), [[x/y]]\, P(\text{asm}) * y : S_{x:A} P$

....

12.3. We mention some derived rules. Indications for derivations are:

T7*: from BR5 and T5,
T8*: from T7* and T2,
T9* and T10*: from T3 and T7*,
T11*: from T4 and BR5,
T12*: from T4, BR7, T11* and T5,
T13*: from T6 and T11*.

....

T7*	A :: substantive	$A \ll A$

....

T8*	$A \ll B$	$A \ll B$ :: statement

....

T9*	$A \ll B$ $a : A$ :: statement	$a : B$ :: statement

....

T10*	$A \ll B$ $a : B$ :: statement	$a : A$ :: statement

....

T11*	$A \ll B$ $a : A$	$a : B$

....

T12*	$A \ll B$ $B \ll C$	$A \ll C$

....

T13*	$x : A(\text{dcl}) * P$ :: statement $y : S_{x:A}\, P$	$y : A$ $[[x/y]]\, P$

....

13. RULES ABOUT EQUALITY

13.1. We shall consider equality between names of objects, between statements and between substantives. The effect of our rules will be that objects will be comparable by equality (being "comparable by equality" means that their equality is a statement) only if they have the same archetype (cf. Section 12.1), and substantives will be comparable by equality only if they are sub-substantives of a common archetype.

Any two statements will always be comparable by equality.

As a metatheory rule (which we shall never explicitly use) we mention that if $p = q$ appears in a book, then p and q are either both objects typed by a substantive, or both substantives, or both statements.

In the (quite strong) rules EQ10a–10c the symbols p and q stand for phrases that possibly show one or more occurrences of the identifier t.

13.2. We first present the fundamental rules of the form displayed in Section 11.

....

EQ1	$a : A$ :: statement $b : A$ :: statement	$a = b$ :: statement

....

EQ2	$a : A$	$a = a$

....

EQ3	$a : A$ $a = b$	$b : A$

....

EQ4	$A \ll C$ $B \ll C$	$A = B$ :: statement
EQ5	A :: substantive B :: substantive $A = B$	$A \ll B$ $B \ll A$
EQ6	$A \ll B$ $B \ll A$	$A = B$
EQ7	P :: statement Q :: statement	$P = Q$ :: statement
EQ8	P :: statement Q :: statement P(asm) $*$ Q Q(asm) $*$ P	$P = Q$
EQ9	P :: statement Q :: statement $P = Q$	P(asm) $*$ Q Q(asm) $*$ P
EQ10a	$u : A$ $v : A$ $u = v$ $t : A$(dcl) $*$ $p = q$ (for notation of substitution see Section 11.6)	$[[t/u]]\, p = [[t/v]]\, q$

EQ10b As EQ10a, but with ":: substantive" instead of ": A".

EQ10c As EQ10a, but with ":: statement" instead of ": A".

13.3. The following rule EQ11 is not of the general form described in Section 11.

. . . .

EQ11 If the set of old lines contains a line of one of the forms

$$C * P := Q : R$$
$$C * P := Q :: \text{substantive}$$
$$C * P := Q :: \text{statement}$$

(cf. the first three cases of Section 10.3), then $P = Q$ is a valid clause in the context C.

. . . .

13.4. We mention the following derived rules, mainly on reflexivity, symmetry and transitivity of equality. Hints for derivation are:

EQ12*: from EQ5 and T11*,
EQ13*: from T9 and EQ1,
EQ14*: from T7* and EQ6,
EQ15*: from EQ5 and EQ6,
EQ16*: from EQ5, T12* and EQ6,
EQ17*: from BR5 and EQ8,
EQ18*: from EQ9 and EQ8,
EQ19*: from EQ9 and BR6,
EQ20*: from EQ9, BR6 and EQ8,
EQ21*: from EQ1, EQ17*, EQ10a, EQ2 and EQ18*,
EQ22*: from EQ1, EQ17*, EQ10a,
EQ23*: from EQ6, T6, T13*, BR6, EQ19*, T5 and EQ6,
EQ24*: from T6, EQ12* and EQ8,
EQ25*: from T5 and EQ6.

. . . .

EQ12*	A :: substantive	
	B :: substantive	$a : B$
	$a : A$	
	$A = B$	

. . . .

EQ13*	$a : A$ :: statement $b : B$ :: statement $A \ll C$ $B \ll C$	$a = b$ :: statement
EQ14*	A :: substantive	$A = A$
EQ15*	A :: substantive B :: substantive $A = B$	$B = A$
EQ16*	A :: substantive B :: substantive C :: substantive $A = B$ $B = C$	$A = C$
EQ17*	P :: statement	$P = P$
EQ18*	P :: statement Q :: statement $P = Q$	$Q = P$
EQ19*	P :: statement Q :: statement $P = Q$ P	Q
EQ20*	P :: statement Q :: statement R :: statement $P = Q$ $Q = R$	$P = R$

EQ21*	$a : A$ $a = b$	$b = a$

....

EQ22*	$a : A$ $a = b$ $b = c$	$a = c$

....

EQ23*	A :: substantive $x : A(\mathrm{dcl}) * P$ :: statement $x : A(\mathrm{dcl}) * Q$ :: statement $x : A(\mathrm{dcl}) * P = Q$	$S_{x:A}\, P = S_{x:A}\, Q$

....

EQ24*	A :: substantive $x : A(\mathrm{dcl}) * P$ :: statement $x : A(\mathrm{dcl}) * Q$ :: statement $S_{x:A}\, P = S_{x:A}\, Q$	$x : A(\mathrm{dcl}) * P = Q$

....

EQ25*	A :: substantive B :: substantive $x : A(\mathrm{dcl}) * x : B$ $x : B(\mathrm{dcl}) * x : A$	$A = B$

....

14. RULES ABOUT SETS

14.1. In this section we shall provide rules that take care of the translation of the language of substantives into the language of sets. This translation is not very essential, and whether we prefer sets over substantives is partly a matter of fashion. But one thing is really important for us: we want to be able to speak of the collection of all subsets of a set, and to quantify over that collection.

The symbols "-set", "$\uparrow$", "$\downarrow$" are pMV.

We use "A-set" as a substantive formed from the substantive A (like "point-set" is derived from "point"). The notation "$A\uparrow$" can be pronounced as "the set of all A's", and if T is an A-set, then $T\downarrow$ can be pronounced as the substantive "element of T".

14.2. Fundamental rules.

S1	A :: substantive	A-set :: substantive $A\uparrow$: A-set
S2	A :: substantive T : A-set	$T\downarrow$ :: substantive $T\downarrow \ll A$ $(T\downarrow)\uparrow = T$
S3	A :: substantive	$A = (A\uparrow)\downarrow$
S4	A :: substantive B :: substantive $A \ll B$	A-set $\ll$ B-set

14.3. In order to get to the ordinary notations about sets we have to introduce some typographical abbreviations (for this notion we refer to Section 21):

$$a \in T \quad \text{stands for} \quad a : T\downarrow$$
$$T_1 \subset T_2 \quad \text{stands for} \quad T_1\downarrow \ll T_2\downarrow$$
$$\{x \in T \mid P(x)\} \quad \text{stands for} \quad (S_{x:A}\, P(x))\uparrow, \text{ where } A = T\downarrow.$$

14.4. We mention a number of derived rules. Hints for derivation are:

S5*: from S1, S4 and T11*,
S6*: from S4, T2, EQ4, S1, T1, T3,
S7*: from S2, T11*, BR5, EQ20*, EQ24*,
S8*: from BR5, S1, EQ14*, EQ10b, S1, EQ2, EQ10b,
S9*: from S4, S3, S2, EQ14*, EQ10a, S2, EQ15*, EQ16*,
S10*: from S1, EQ14*, EQ10a,
S11*: from S2, S8*, EQ23*, EQ24*.

S5*	A :: substantive B :: substantive $A \ll B$	$A\uparrow$: B-set

S6* A :: substantive
B :: substantive
C :: substantive
$A \ll C$
$B \ll C$

A-set $\ll$ B-set :: statement
A-set $=$ B-set :: statement
$A\uparrow$: B-set :: statement

....

S7* A :: substantive
T_1 : A-set
T_2 : A-set
x : $A(\text{dcl}) * (x \in T_1) = (x \in T_2)$

$T_1 = T_2$

....

S8* A :: substantive
B :: substantive
$A = B$

A-set $=$ B-set
$A\uparrow = B\uparrow$

....

S9* A :: substantive
B :: substantive
C :: substantive
$A \ll C$
$B \ll C$
$A\uparrow = B\uparrow$

$A = B$

....

S10* A :: substantive
T_1 : A-set
T_2 : A-set
$T_1 = T_2$

$T_1\downarrow = T_2\downarrow$

....

S11* A :: substantive
T_1 : A-set
T_2 : A-set
$T_1\downarrow = T_2\downarrow$

$T_1 = T_2$

....

15. RULES ABOUT PAIRS

15.1. If we have two substantives, "point" and "line", say, we want to speak of *pairs*, the first component of which is a point, the second one a line. Such pairs are called "point-line-pairs". The symbols "-pair", "proj1", "proj2", "the pair" are pMV.

15.2. Fundamental rules about pairs.

P1	A :: substantive B :: substantive	A-B-pair :: substantive
P2	$a : A$ $b : B$	the pair (a, b) : A-B-pair
P3	A :: substantive B :: substantive u : A-B-pair	proj1(u) : A proj2(u) : B the pair (proj1(u),proj2(u)) $= u$
P4	$a : A$ $b : B$	proj1(the pair (a, b)) $= a$ proj2(the pair(a, b)) $= b$

15.3. We mention two derived rules. Hints for derivation are:

P5*: from P3, T11*, P2, EQ22*, T5,
P6*: from P5*, T2, EQ4.

P5*	$A \ll C$ $B \ll D$	A-B-pair $\ll$ C-D-pair
P6*	$A \ll E$ $C \ll E$ $B \ll F$ $D \ll F$	A-B-pair $\ll$ C-D-pair :: statement A-B-pair $=$ C-D-pair :: statement

16. RULES ABOUT FUNCTIONS

16.1. If A and B are substantives we shall introduce a new substantive "mapping of A's to B's". We write this in MV as "$A \to B$"; the symbol $\to$ is pMV. The fact that the same arrow is used for implication (Section 17) will give no confusion. Actually the rules for the two are alike (compare F2 and F3 with L2 and L3).

For the value of the function f at the point p we shall write in MV "val(f, p)", instead of the usual $f(p)$. The notations val and λ are pMV.

16.2. Fundamental rules about functions.

F1	A :: substantive B :: substantive	$A \to B$ :: substantive
F2	A :: substantive B :: substantive $f : A \to B$ $p : A$	$\text{val}(f, p) : B$
F3	A :: substantive B :: substantive $x : A(\text{dcl}) * F : B$	$\lambda_{x:A} F : A \to B$
F4	A :: substantive B :: substantive $x : A(\text{dcl}) * F : B$ $y : A$	$\text{val}(\lambda_{x:A} F, y) = [[x/y]]\, F$
F5	A :: substantive B :: substantive $f : A \to B$ $g : A \to B$ $x : A(\text{dcl}) * \text{val}(f, x) = \text{val}(g, x)$	$f = g$

16.3. Here are two derived rules. Hints for derivation are:

F6*: from F2, F4, F5,
F7*: from T11*, EQ12*, F3, F6*, EQ22*, T5.

....

F6*	A :: substantive B :: substantive $f : A \to B$	$\lambda_{x:A}\,\mathrm{val}(f,x) = f$

....

F7*	A :: substantive B :: substantive C :: substantive D :: substantive $A = C$ $B \ll D$	$A \to B \ll C \to D$

....

16.4. Many mathematicians would prefer to express the notion of function by means of a graph in a cartesian product, which has the advantage to reduce the number of basic rules. On the other hand the function concept seems to be such a natural one, and the way we think of functions is usually so far from cartesian products, that it is attractive to describe the function concept independently.

17. RULES ABOUT LOGIC

17.1. The only things to be presented in this section are the rules for implication and for universal and existential quantification. Treatment of negation, conjunction and disjunction can be postponed to the MV book. For this possibility we refer to Section 18.1.

The symbol "$\to$" is pMV; the same arrow was used in Section 15 for the notation of mappings.

17.2. We first present the fundamental rules L1, L2, L3.

....

L1	P :: statement Q :: statement	$P \to Q$ :: statement

....

L2	P :: statement Q :: statement $P \to Q$	$P(\text{asm}) * Q$

....

L3	P :: statement Q :: statement $P(\text{asm}) * Q$	$P \to Q$

....

17.3. We mention some simple derived rules about implication. Hints for proofs are:

L4*: from BR5, L3,
L5*: from L2, EQ8,
L6*: from L4*, EQ10b.

....

L4*	P :: statement	$P \to P$

....

L5*	P :: statement Q :: statement $P \to Q$ $Q \to P$	$P = Q$

....

L6*	P :: statement Q :: statement $P = Q$	$P \to Q$ $Q \to P$

....

17.4. We finally present derived rules on universal quantification. Let P be an expression (possibly containing the identifier x). Then we take "$\forall_{x:A} P$" as typographical abbreviation (see Section 21) for "$A = S_{x:A} P$".

For this new quantifier the following rules can be derived.

....

L10*	A :: substantive $x : A(\text{dcl}) * P$ :: statement	$\forall_{x:A} P$ :: statement

....

L11* A :: substantive
$x : A(\mathrm{dcl}) * P$ :: statement $\forall_{x:A} P$
$x : A(\mathrm{dcl}) * P$

...

L12* A :: substantive
$x : A(\mathrm{dcl}) * P$ :: statement
$\forall_{x:A} P$ $[[x/a]]\, P$
$a : A$

...

17.5. The reader might have expected a treatment of existential quantification too. This can easily be postponed to the MV book. It can be built upon axioms for a statement exist(A), where A is a substantive. It seems to be nicer to postpone that to the book, since it is of the same nature as the axioms for disjunction in propositional calculus.

18. EXAMPLE OF AN MV BOOK

18.1. Having completed our presentation of the rules of MV, we can now start writing books. In the beginning of an MV book we still have to write a number of fundamentals in the form of primitive notions and axioms. These might have been taken as language rules too, but we would rather leave it to the user of the language to have it in his own way. Moreover, the language definition is simplified if we shift to the book whatever we can.

Nevertheless there are several things in our language rules that have a form that would enable us to write them in the book. As examples we mention EQ2, EQ3, S3, F1, F2, F5, L1, L2, L3 as far as "fundamental" rules are concerned. In the case of S1, F1, L1 we had serious reasons for not shifting them to the book: they were needed for the formulation of further rules that had to stay in the language definition because of their form. For the others there is no other reason than the wish to keep related material together. The fact that some of them play a role in the derivation of derived rules (like EQ2 is used in the derivation of EQ21*), is not a serious reason. The derived rules do not belong to the definition and theory of MV, and might as well be postponed until after a piece of the book has been written.

In Section 19.4 we show that some book material might also have been put in the form of rules of the type of Section 17.

18.2. In the following MV book with pointed and rectangular flags (cf. Section

6.4) we have numbers (1), (2), ... on the left. These do not belong to the book, but serve as labels for our comments in Section 19.

```
(1)   | a :: statement >
(2)   | | b :: statement >
(3)   | | if a then b := a → b :: statement
(4)   | | | [b]
(5)   | | | | [a]
(6)   | | | | b
(7)   | | | a → b
(8)   | | if b then a → b
(9)   | | | [a]
(10)  | | | | [a → b]
(11)  | | | | b
(12)  | | a and b := PN :: statement
(13)  | | a or b  := PN :: statement
(14)  | | | [a]
(15)  | | | a or b [ Axiom ]
(16)  | | | | [b]
(17)  | | | | a and b [ Axiom ]
(18)  | | | | b and a
(19)  | | | [b]
(20)  | | | a or b [ Axiom ]
(21)  | | | [a and b]
(22)  | | | a [ Axiom ]
(23)  | | | b [ Axiom ]
(24)  | | | b and a
(25)  | | | [a or b]
(26)  | | | | c :: statement >
(27)  | | | | | [a → c]
(28)  | | | | | | [b → c]
(29)  | | | | | | c [ Axiom ]
(30)  | | | a → (b or a)
(31)  | | | b → (b or a)
(32)  | | | b or a
(33)  contradiction := PN :: statement
(34)  no := contradiction :: statement
(35)  | a :: statement >
```

(36) $\text{not}(a) := a \to \text{no}$ :: statement
(37) $\text{no} \to a$ [Axiom]
(38) $((a \to \text{no}) \to a) \to a$ [Axiom]
(39) $\text{not}(\text{not}(a))$
(40) $\text{not}(a)$
(41) no
(42) a
(43) $(a \to \text{no}) \to a$
(44) a
(45) $\text{not}(\text{not}(a)) \to a$
(46) b :: statement
(47) $b \to a$
(48) $\text{not}(b) \to a$
(49) $\text{not}(a)$
(50) b
(51) a
(52) no
(53) $\text{not}(b)$
(54) a
(55) $\text{not}(a) \to a$
(56) a
(57) a and b = $\text{not}(a \to \text{not}(b))$
(58) a or b = $\text{not}(a) \to b$
(59) a or $\text{not}(a)$
(60) A :: substantive
(61) $\text{exist}(A) :=$ PN :: statement
(62) $a : A$
(63) $\text{exist}(A)$ [Axiom]
(64) $\text{exist}(A)$
(65) c :: statement
(66) $\forall_{a:A}\, c$
(67) c [Axiom]
(68) $\forall_{x:A}\, \text{no}$
(69) no
(70) $\text{not}(\forall_{x:A} \text{no})$
(71) $\text{not}(\forall_{x:A} \text{no})$
(72) $\text{not}(\text{exist}(A))$
(73) $x : A$

(74) exist(A)
(75) no
(76) $\forall_{x:A}$no
(77) no
(78) not(not(exist(A)))
(79) exist(A)
(80) exist(A) $=$ not($\forall_{a:A}$ no)
(81) there is exactly one A := exist($S_{a:A}\forall_{b:A}$ $(b = a)$) :: statement
(82) there is exactly one A
(83) the A := PN : A
(84) $b : A$
(85) $b =$ the A
(86) $a : A$
(87) $\{a\}_A :=\uparrow_{x:A}$ $(x = a)$: A-set
(88) X : A-set
(89) $a : X\downarrow$
(90) $\{a\}_{X\downarrow} = \{a\}_A$
(91) $b : A$
(92) $\{a, b\}_A :=\uparrow_{x:A}$ $(x = a$ or $x = b)$: A-set
(93) $a \in \{a, b\}_A$
(94) $b \in \{a, b\}_A$
(95) $c : A$
(96) $c \in \{a, b\}_A$ and not $(c = a)$
(97) $c = b$
(98) A :: substantive
(99) B :: substantive
(100) $f : A \to B$-set
(101) union(A, B, f) $:=\uparrow_{b:B} \exists_{a:A} b \in$ val(f, a) : B-set
(102) A :: substantive
(103) $a : A$
(104) $b : A$
(105) P :: statement
(106) selected(A, a, b, P) :=
$S_{x:A}(((x = a)$ and $P)$ or $((x = b)$ and not $P))$ :: substantive
(107) there is exactly one selected(A, a, b, P)
(108) selection(A, a, b, P) := the selected(A, a, b, P) : A
(109) if P then (selection(A, a, b, P) $= a$)
(110) if not(P) then (selection(A, a, b, P) $= b$)

(111) natural number := PN :: substantive
(112) nat := natural number :: substantive
(113) N := nat ↑: nat-set
(114) 1 := PN : nat
(115) [n : nat ⟩
(116) successor of n := PN : nat
(117) suc(n) := successor of n : nat
(118) not(suc(n) = 1) [Axiom]
(119) [m : nat ⟩
(120) if suc(n) = suc(m) then $n = m$ [Axiom]
(121) [S : nat-set ⟩
(122) start(S) := $1 \in S$ :: statement
(123) propagate(S) := $\forall_{n:N}((n \in S) \rightarrow (\text{suc}(n) \in S))$ :: statement
(124) if start(S) and propagate(S) then $S = N$ [Axiom]
(125) [n : nat ⟩
(126) [m : nat ⟩
(127) $n \cdot m$:= PN : nat
(128) m divides n := $\exists_{k:\text{nat}}\ (k \cdot m = n)$
(129) divisor of n := $S_{k:\text{nat}}$ (k divides n) :: substantive
(130) (divisor of n) $\ll$ nat
(131) prime number := $S_{p:\text{nat}}(\text{not}(p = 1) \text{ and } \forall_{k:\text{divisor of } p}(k = 1 \text{ or } k = p))$
:: substantive
(132) prime number $\ll$ nat
(133) [p : prime number ⟩
(134) [q : prime number ⟩
(135) not ($p \cdot q$: prime number) :: statement

(136) point := PN :: substantive
(137) line := PN :: substantive
(138) [A : point ⟩
(139) [b : line ⟩
(140) b goes thr. A := PN :: statement
(141) [B : point ⟩
(142) [b : line ⟩
(143) b goes thr. A and B := (b goes thr. A) and (b goes thr. B)
(144) [not($A = B$)]
(145) there is exactly one ($S_{c:\text{line}}$ (c goes thr. A and B)) [Axiom]
(146) [C : point ⟩

(147) b : line

(148) A, B, C on $b :=$ (b goes thr. A and B) and (b goes thr. C) :: statement

(149) $\exists_{A:\text{point}}\ \exists_{B:\text{point}}\ \exists_{C:\text{point}}$ not($\exists_{d:\text{line}}$ (A, B, C on d)) [Axiom]

19. COMMENTS ON THE EXAMPLE OF AN MV BOOK

19.1. In Section 18.2 the text from (1) to (59) represents a piece of propositional logic. It is not complete in the sense that it contains everything one might ever need, but it is sufficiently representative for showing the following things.

(i) Some logic can indeed be developed in the book, so we do not need to put all of it in the language rules.

(ii) The logical statements we derive in the book can be applied later as inference rules, in the same way as mathematical theorems are applied. So there is hardly a borderline between logic and mathematics. A much more prominent borderline is the one between language definition and book material.

(iii) The book starts with minimal propositional logic (lines (1) to (32)): just the rules for introduction and elimination of implication, conjuction and disjunction, without any negation. Next there is the introduction of contradiction and negation (lines (33) to (36)) and the "falsum rule" (37) as a logical axiom. This part of the book ((1) to (37)) might be used as a basis for intuitionistic mathematics. It is not very likely, however, that our system would satisfy all intuitionists. They might dislike some of the things we have preferred to put in the language rules, like the possibility to introduce arbitrary substantives, and the rules S1-S4 (which make it possible to talk about the set of all subsets of a set).
The extra axiom (38) takes us into classical logic, with the double negation rule (44) and the rule of the excluded third (59) as results. From there on we are in classical logic. It has to be admitted that if classical logic had been our only goal, the number of primitives and axioms might have been reduced considerably. But reducing the number of axioms, important as it might be for metatheory, is completely irrelevant from the practical point of view: later in the book old axioms and primitives are applied in exactly the same way as old theorems and old defined notions.

(iv) The spirit of the treatment is natural deduction. There is no trace of treating logic by means of truth values. Contrary to popular opinion, it is hard, if possible at all, to explain logical reasoning by means of truth values, unless one cheats by using a priori knowledge of logical reasoning in order to explain what such reasoning is. But nothing is lost by discarding truth tables. Everything that can be done with truth tables, can be done in natural deduction, usually faster, and usually closer to our actual way of thinking.

19.2. We now comment on some of the details of the MV book of Section 18.2.

The book starts with implication. Line (3) introduces an alternative notation for the $\rightarrow$. Lines (4) to (8) are meant as a little exercise with the introduction of implication, according to rule L3. Note that we have (6) by the fact that we had "b" in an old line in a smaller context (BR5 and BR7). Now (7) follows by L3. Similarly the little theorem (8) follows from (7).

Lines (9) to (11) apply rule L2 for the elimination of implication, the so-called "modus ponens" rule. Actually it makes the rule available as a book theorem: any further case of modus ponens can be seen as an application of this theorem (11).

A similar remark holds for many mathematical theorems. We need not always transform results obtained inside a block into results with a shorter context by means of L3 and L11*. We can just leave them quietly in their context, ready for the application of the powerful rule BR6.

Lines (12) and (13) introduce the conjunction and disjunction as primitives. The introduction rule for the conjunction is given in (17) and (18); note that (18) need not be labeled as an axiom since it can be obtained from (17) by substitution: b for a and a for b. The elimination rule for the conjunction is given by (22) and (23), and here both have to be axioms.

The introduction rule for the conjunction is expressed by the two axioms (15) and (20). Elimination of disjunction is achieved by the more complex rule (29). It is the basis for "proof by cases": if we want to prove c and we know that a or b, then it suffices to derive c from a and from b separately.

In (33) the primitive notion "contradiction" is introduced. Thus far there was no question of "falsehood", since all validity rules in the language definition are formulated in a positive way. In an MV book there is no reason to say that a thing is true: a man is not more honest because he *says* that he is honest. But we do say sometimes that a thing is false, by saying that it implies a contradiction. This is expressed in (36). In (34) we just abbreviated "contradiction" to "no". This has the same function as the typographical abbreviations (Section 21), but we prefer to restrict the use of typographical abbreviations to cases that can *not* be expressed in the book. The same remark applies to the notation $\neg a$ instead of not(a) : it can be introduced in a definitional line in the MV book if we wish.

Classical logic is obtained by adding the double negation law (45): it says that if the negation of a is not true, then a is true. In the present text it is a theorem instead of an axiom, derivable from the two axioms (37) and (38). The first one is the intuitionistic "falsum rule", the second one is a special case of Peirce's law $((a \to b) \to a) \to a$.

We have taken (38) as an axiom since it gives rise to interesting exercises in natural deduction. Let us assume that we did not have the rules (12) to (32) in our book. Then we can still say that the rule (38) (holding for all statements a) has the *effect* of a disjunction, viz. the disjunction of b and not(b) (for all b). Line (56) shows that a can be proved by cases, just as if "b or not (b)" had been available.

In (45) we have the double negation law: the negation of the negation of a statement implies that statement itself.

If we would have taken the double negation rule (45) as an axiom, we might have derived both (37) and (38) as theorems. In a certain sense (37) and (38) form an orthogonal decomposition of (45). This is to be interpreted with the following notion of orthogonality: statements p and q are called *orthogonal* if both $(p \to q) \to q$ and $(q \to p) \to p$. In a way that means that q and p do not give any information about each other: if we can prove q under the assumption p then we can prove q all by itself, and if we can prove p under the assumption q then we can prove p all by itself. And indeed, without using the book from (1) to (32) it can be shown directly after line (35) that no $\to a$ and $((a \to \text{no}) \to a) \to a$ are orthogonal.

A few lines about the derivation of (45). By modus ponens we have

$$\text{not}(\text{not}(a))\ (\text{ass}),\ \text{not}(a)\ (\text{ass})\ *\ \text{no}\ ,$$

so the falsum rule leads to

$$\text{not}(\text{not}(a))\ (\text{ass}),\ \text{not}(a)\ (\text{ass})\ *\ a\ .$$

Therefore

$$\text{not}(\text{not}(a))\ (\text{ass})\ *\ (a \to \text{no}) \to a\ ,$$

so applying (38) with modus ponens we get

$$\text{not}(\text{not}(a))\ (\text{ass})\ *\ a\ .$$

Line (59) is the so-called rule of the excluded third. Abbreviating "a or not(a)" to c, we get it from (56), with a replaced by c and b by a. Notice that both a and not(a) lead to c, because of (15) and (20).

The basic rules BR1-BR7 and the equality rules EQ1-EQ25* soon become second nature to us, and therefore we shall hardly notice their application any more.

19.3. At this point we note that it is hard to give a satisfactory description of the word "proof" as an smMV term. Looking ahead from a line (i) to a line (j) (where i<j) one might say that the text between (i) and (j) is a *proof* of (j). Others might only count the material between (i) and (j) as far as it is actually *used* for the derivation of (j). More important is the question whether explanations of the type given in Section 19.2 belong to the proof. If the steps in the MV book are so small that each line requires just a single application of a single rule, then most people would call it a very detailed proof, even if it is not mentioned in the text what rules were applied. We can omit lines here and there such that most people will be able to find intermediate steps themselves by mental exercise. In the case of (24) most people will be able to find the missing link in a split second. In the case of (57) there is a sequence of missing links, and we will feel the need for scrap paper, even if we are experienced in this kind of natural deduction.

We can go quite far in this respect. It was pointed out already in Section 1.9 that the rules of MV allow to omit intermediate steps, even to such an extent that readers may find that the essence of the proof is lacking. This is caused by the fact that validity in MV is defined recursively. Something can be valid because of the *existence* of a sequence of intermediate steps; it is *not* required that these steps have actually been written down in the book.

19.4. Some of the logical parts of the text of Section 18.2 give us the same rights as if we had added a number of further rules in Section 17. We shall display them in that form here, labeled with two asterisks instead of one, since they are not derived from the fundamental rules of Sections 9–17 but from material of the particular book of Section 18.

L13**	P :: statement Q :: statement P Q	P and Q

L14**	P :: statement Q :: statement P and Q	P Q

L15**	P :: statement Q :: statement P	P or Q

L16**	P :: statement	
	Q :: statement	P or Q
	Q	

....

L19**	P :: statement	
	Q :: statement	
	R :: statement	R
	$P \to R$	
	$Q \to R$	
	P or Q	

....

19.5. In (60)–(97) we show a few things about sets, sufficiently representative for showing how one should go on.

Lines (60)–(67) introduce the notion of existence on a negation-free basis. Once we have classical logic, we can do more. In (64)-(80) we show the equivalence of "exist(A)" and "not($\forall_{x:A}$ no)", which is the basis for expressing existential quantifiers into universal ones, and vice versa. We mention that (69) is obtained by application of (67) with c replaced by no; with this value of c assumption (66) is valid because of (68). Next, (70) rests on L3, (69) and, of course, (36). One gets (74) from (63), replacing a by x, and (75) from L2, using (72) and (74). Then (76) follows from L11*, using (75), next (77) from L2 with (71), (76), and (78) from L3 and (77). Finally we get (79) from (78) and (45), and (80) from (70), (79) by virtue of L5*.

In (83) we introduce (as a primitive notion) the definite article "the" in front of a substantive, if that substantive has the uniqueness property assumed in (82). In (85) we assert that if the uniqueness property holds then every A equals "the A".

A detailed proof of (85) can be given as follows. Abbreviate $K(A) := S_{a:A}\forall_{b:A}\ (b = a)$. Then (82) says exist($K(A)$). We want to apply (67), replacing A by $K(A)$ and c by "$b =$ the A". To that end we have to satisfy what the condition (66) amounts to in this case, i.e., $\forall_{a:K(A)}(b =$ the $A)$. In order to prove the latter statement, we extend the context by means of $a : K(A)$ (dcl). In the extended context we have to show $b =$ the A. In this context we have $a : K(A)$, and therefore $\forall_{d:A}\ (d = a)$. Using L12* we get both $b = a$ and the $A = a$, so by EQ21* and EQ22* we infer $b =$ the A.

Note that this derivation uses no classical logic. It is entirely negation-free.

In (87) we define the singleton $\{a\}_A$ as an A-set. In usual untyped set theory the subscript A is superfluous, but here it is not. Nevertheless, having to write the subscript is a formal duty only, for if the same a also satisfies $a : B$ then $\{a\}_A = \{a\}_B$. We know (by metatheory) that the typings $a : A$ and $a : B$ can hold simultaneously only if A and B are sub-substantives of a common K (which might be the archetype of a), and therefore (90) helps us out.

Note that in (87) and (92) the typographical abbreviation $\uparrow_{x:A}$ us used (see (20.7.2)).

In the context of (87) we can define the empty set too:

$$A :: \text{substantive(decl)} \; * \; \text{emptyset}_A := \uparrow_{x:A} \; \text{no} : A\text{-set} \; .$$

Note that we do not have a universal empty set: every archetype has one of its own.

In (90) we wanted to express that if A and C are substantives with $C \ll$A and $a : C$ then $\{a\}_C = \{a\}_A$. But unfortunately, our language rules do not allow us to open a context expressing that a new identifier C represents an arbitrary sub-substantive of A. We have to introduce an arbitrary A-set X (see (88)), to derive the substantive $X\downarrow$ from it, and this $X\downarrow$ can play the role of the C we wanted.

One might dislike this, but it would not do any harm to extend the language rules by admitting declarational context items of the form "$C \ll A$ (dcl)" for expressing directly "let C be an arbitrary sub-substantive of A".

From (91) to (97) we have a few things on the set notation $\{a, b\}$. One might go on with $\{a, b, c\}$,$\{a, b, c, d\}, \ldots$, but in this way we cannot get to the general notation. There are two essentially different ways to do this, but these two ways are often confused.

(i) We define in the MV book something that can be seen as $\{a_1, \ldots, a_n\}$, where n is a variable in the MV book, with n:nat (it can be done only after the introduction of natural numbers, see (111)-(124), and may use the unions defined in (101)).

(ii) As typographical abbreviations (cf. Section 21) we define $\{a_1, a_2, a_3\}$, $\{a_1, a_2, a_3, a_4\}, \ldots$. A way to explain these is to take

$$(S_{x:A}(x = a_1) \text{ or} \cdots \text{or } (x = a_n))\uparrow$$

as the thing denoted by $\{a_1, \ldots, a_n\}$. In this case n is a variable natural number in the metalanguage smMV.

The two ways (i) and (ii) can be connected in the MV book for every single value of n, but not for all n simultaneously. We note that the length of the derivation (i.e. the number of applications of language rules) is more or less proportional to n.

In (101) we define the union of an indexed collection of sets.

Lines (102)–(110) present the "if-then-else" selector, which can be used as a basis for definition of functions by cases. We omit indications of the proofs. Note that (108) uses the "the" of (83).

19.6. The text from (111) to (135) deals with the natural number system. The Peano axioms are (112), (113), (114), (116), (118), (124). In (127) we have presented the notion of the product of two natural numbers by means of a PN. This can of course be avoided (the product can be defined), but the text would become lengthy, and it is our present purpose to get rapidly to divisibility. In (129) we define a substantive "divisor of n", in (130) it is noted that it is a sub-substantive of "nat", and (131) gives the definition of "prime number". In (135) we form the statement that the product of two primes is not a prime. Note that this contains an example of a typing ($p \cdot q$: prime number) playing the role of a statement, which is allowed by virtue of (132) since $p \cdot q$: nat. It would be wrong to claim "not ($p \cdot q$: prime number)" as a theorem here: a proof would require more information about products than what is expressed in (127).

19.7. The text from (136) to (149) is based on the beginning of Hilbert's axiomatization of geometry. Hilbert starts by saying "We conceive three different systems of things: the things of the first system are called "points", those of the second system are called "lines", those of the third system "planes"." Hilbert does not make any use of his "systems" as systems: what he actually does is just handling the words "point", "line", "plane" as new substantives. Therefore we interpreted his words in MV by taking them as PN's.

As to (143) one might hesitate. Does this really require a mathematical definition or is it just one of the linguistic transformations we want to admit anyway? We refer to Section 22 for such matters.

20. BINDERS

20.1. We have not spent much attention on quantification by means of bound variables and quantifiers. Formal treatment of quantification in lambda calculus is well known, of course. One of the hard things in quantification is the treatment of the names of bound variables, which have to be refreshed occasionally. In this section we do not go into these standard matters of non-typed lambda calculus. Instead, we shall indicate a number of points in which our present proposal of MV differs from more usual ways to treat quantification.

20.2. First we mention that our MV is a typed language, and that, accordingly, the bound variables in the quantifications run over a certain range. The range can be indicated by a substantive A, using a typing "$x : A$", or by a set S, and then we use $x \in S$. By 14.3 we can easily pass from one to the other, and so we restrict our discussions to the case "$x : A$".

20.3. In OMV the "value" of a result of quantification is either a statement or an object. Examples of the first kind are

$$\forall_{n:\text{natural number}} P(n) \;, \qquad \exists_{n:\text{natural number}} P(n) \;,$$

where $P(n)$ is a statement. Examples of the second kind are

$$\sum_{n=1}^{m} f(n) \;, \qquad \bigcup_{n \in S} V(n) \;, \qquad \{x \in S | P(x)\} \;,$$

where $f(n)$ is a number, $V(n)$ is a set, $P(x)$ a statement.

20.4. In MV, where substantives are taken seriously, we can also admit quantification where the value of the result is a substantive. One of the possibilities is given by the binder S, introduced in rule T6, Section 12.2. Its meaning is shown in the following example. We consider "quadrilateral with the property that its diagonals are perpendicular to each other" as a new substantive. It can be written by means of a quantifier as

$$S_{x:\text{quadrilateral}}(\text{the diagonals of } x \text{ are perpendicular}) \;.$$

20.5. The following example demonstrates a second case where quantification leads to a new substantive. We have the name "square of p" if p is a prime number. We want to *despecify* the variable p, and get to the substantive "square of a prime", or "prime square". For this we can use the binder "despo" and write

$$\text{despo}_{p:\text{prime}} \; (\text{the square of } p)$$

(despo is short for "despecified object"). This binder can be considered as a typographical abbreviation (see Section 21.2).

20.6. Many quantifiers can be expressed once we have the *functional binder* (Church's λ). If A and B are substantives, if $F(...x...)$ is an expression containing x at one or more places, with the property that $x : A$ implies $F(...x...) : B$, then

$$\lambda_{x:A} \; F(...x...)$$

is the function that attaches, for each x, the value $F(...x...)$ to x. Example:

(20.6.1) $\lambda_{x:\text{positive integer}}(x^2 + x)^{-2}$.

Some other quantifiers can be expressed in terms of this one. For example, the sum

(20.6.2) $\sum_{n:\text{positive integer}} (n^2 + n)^{-2}$

might be written as

(20.6.3) $\text{sum}(\lambda_{n:\text{positive integer}}(n^2 + n)^{-2})$.

This means that the quantification (20.6.2) can be obtained by application of the unary operator "sum" to the function (20.6.1). In contrast to (20.6.2), all the binding in (20.6.3) is in the function, and nothing of it in the operation.

20.7. The binder S of rule T6, Section 12.2, plays a similar central role as Church's λ. In Section 20.6 we had the unary operator "sum", acting on an expression quantified by a λ, in the present case we take as examples the unary operators "exist" (from Section 18.2, formula (61)) and "$\uparrow$" (from Section 14.2). The operator "exist" maps substantives into statements, the "$\uparrow$" maps substantives into names. If P is an expression containing x, such that for all $x : A$ we have P :: statement, then we can form a new statement $\exists_{x:A}$ by agreeing that

(20.7.1) $\exists_{x:A}\ P(x) \ = \ \text{exist}(S_{x:A}P(x))$

and the new name

(20.7.2) $\uparrow_{x:A}\ P(x) \ = \ (S_{x:A}P(x))\uparrow$.

This (20.7.2) is usually written on OMV as $\{x : A|P(x)\}$.

We get the standard rules for introduction and elimination of the quantifiers $\exists_{x:A}$ and $\uparrow_{x:A}$ from T6 (Section 12.2) in combination with the rules for the unary operator "exist" (Section 18.2, (60)-(67)) and the rules for the unary operator $\uparrow$ (Section 14.2).

20.8. At this point it should be noted that we have not provided facilities in MV to deal explicitly with predicates. If A is a substantive then we cannot express in mV that something is a predicate over A. Instead, we have to use the rules of Section 14.

A predicate is usually seen as a mapping from objects to statements. As an ~~example, we consider the property of a~~ natural number to be > 5. One might suggest $\lambda_{x:\text{nat}}(x > 5)$ that sends any natural number x into the statement $x > 5$. It seems attractive, but we have not incorporated this into MV. It would not not quite fit into our system to attach a type to such a thing, and therefore it would not help us to create arbitrary predicates.

The set-forming rules of Section 14 help us out. Instead of discussing a predicate in MV, we discuss the set of all objects satisfying that predicate. So instead of that $\lambda_{x:\mathbf{nat}}(x > 5)$ we talk about the set $\Uparrow_{x:A} P(x)$ (cf. (20.7.2). And instead of taking arbitrary predicates we take an arbitrary A-set, as in formula (88) of Section 18.2.

20.9. MV does not allow the introduction of new quantifiers in the book. The reason is that the language is not equipped with means for saying "an expression containing x". Have only two basic quantifiers in the language definition, viz. the substantive binder S and the functional binder λ. All other binders have to be expressed in terms of these two in the way indicated in Section 20.6.

If we insist on using notations like (20.6.2), (20.7.1), (20.7.2), we have to treat them as typographical abbreviations, to be discussed in Section 21.

21. TYPOGRAPHICAL ABBREVIATIONS

21.1. Of course we like to use OMV notations in MV as much as possible. We can do this in an informal way by using what we shall call *typographical abbreviations.*

We can agree that if we write (20.6.2) in an MV book, this is just an informal abbreviation for (20.6.3). The agreement to use that abbreviation cannot be made in the book itself, but has to be written in the margin somehow. A similar remark applies to (20.7.1) and (20.7.2).

21.2. As a further example we take the "despo" of Section 20.5. If A and B are substantives, and if $P(...x...)$ is an expression containing x such that for all $x : A$ we have $P(...x...) : B$, then

$$\text{despo}_{x:A}\ P(...x...)$$

can be considered as typographical abbreviation of

$$S_{y:B}\ \exists_{x:A}\ (y = P(...x...))\ .$$

21.3. The words "typographical abbreviations" indicate unofficial abbreviations, usually (but not always) invented for the sake of typography. When reading a text that uses such abbreviations we first have to translate them into what they stand for, and only after that translation we are assumed to be able to understand the text as an MV book.

Typographical abbreviations are superficial, when compared to the abbreviations we introduce in the MV book itself by means of definitional lines (see

Section 7.6). Definition and theory of the language have to take these definitional lines as essential parts of the language, but never deal with typographical abbreviations.

As an example of a typographical abbreviation outside the world of quantifiers we quote the notation $\{1, \ldots, n\}$ for the set of integers from 1 to n. We refer to the discussion in Section 19.5.

21.4. When studying mathematical notation, we discover many other cases of abbreviations that we would prefer to consider as informal. Some examples are:

(i) We write $a = b < c = d$ instead of the conjunction of $a = b$, $b < c$, $c = d$.

(ii) We write $a(1), a(2), \ldots$ to denote an infinite sequence.

(iii) We have baroque notations for 17-th and 18-th century mathematics, on integrals, derivatives, differential equations, etc.

(iv) We have many unwritten conventions by which we omit things that, strictly speaking, would be necessary for parsing. These may be local conventions in a certain area of mathematics. In trigonometry one interprets sin x cos y as the product (sin x)(cos y); the alternative sin (x cos y) occurs in the theory of Bessel functions but not in trigonometry.

21.5. Many formulas in OMV are written in a form that do not fit into a single line. A simple example is the old notation for quotients by means of a horizontal bar. If we think of MV having to be processed by a computer it seems that such notations have to be avoided, but it is not a matter of principle. Formats which do not have the form of a string of characters might be admitted in MV just as well.

As a simple example we note that sometimes the value that a function takes at the point n is denoted by using the letter n as a subscript, like in c_n. This should not be confused with the habit of using $c_1, c_2, \ldots$ as new identifiers. The difference between the two is of the same type as the difference between the two ways to look at $a_1, \ldots, a_n$, discussed in Section 19.5.

21.6. There will be many cases where one is easily tempted to interpret the rigid rules of MV with a little grain of salt. At least, as long as the texts are intended to be read by human beings only. If we present them to machines, we have to be much more careful.

It was already mentioned in Section 5.3 that we often cheat with the rules that require identifiers to be fresh. But it is not necessary to be so rigid in cases of variables introduced by means of declarations. It suffices to have them different from all previous variables of that same context and different from all previously introduced constants (either by definition or by PN). But the rule that all introduced constants should have different identifiers, will often be felt as a burden: in mathematics constants occurring in distinct subjects will often have the same name. When writing for human readers, there does not seem to be any harm. When writing for machines, we should do something like the paragraph system that was used in Automath, where identifiers are always interpreted in the sense given to them in the local paragraph. An old constant with the same name can only be referred to if we add some kind of paragraph indication.

Another case for grains of salt was already mentioned in Section 5.5. The condition that all context variables occur in the name of a defined notion is quite different from habits in OMV.

In Automath there is a systematic way to weaken this rule: in parentheses expressions like $F(A_1, \ldots, A_n)$ we may just omit the first k entries $A_1, \ldots, A_k$ if they are identical to the first k variables of the context. In particular that has the effect that in a definitional line in a context with variables $x_1, \ldots, x_n$ the parametrized constant on the left of sign ":=" can be written as a single identifier (cf. Section 5.6). But it should be noticed that in MV and in OMV not all parametrized constants have the form of such parentheses expressions. That makes it harder to formulate what liberties can be taken in the matter of omission of a number of context variables.

22. GETTING CLOSER TO NATURAL LANGUAGE

22.1. In the examples of Section 18.2 we inserted a small piece of what one might call natural language. On a small scale it showed how mathematics can be described in words and sentences, not just in symbols and formulas. If we want to insert more natural language, or even if we want our MV book to look like an ordinary mathematics book, we can do this on three levels:

(i) the primary MV level (pMV),

(ii) the secondary MV level (sMV),

(iii) the level of typographical abbreviations.

22.2. As to primary MV we can discuss how some of the basic notations on typing, on context and on quantification are to be expressed in terms of words. In several cases it is not quite clear how to do this: there may arise ambiguities, in particular just those which the MV notations intended to avoid. Right now we do not try to suggest how to solve all these difficulties. We hardly go beyond a first orientation.

22.3. The typing "p:prime number" can of course be pronounced "p is a prime number". A definitional line body "$p := Q$: prime number" can be pronounced as "denote the prime number Q by p" (in this case Q is an expression and p is an identifier). The declaration "p:prime number" (the case of a pointed flag) is "let p be a prime". The assumption "p:prime number" can be "assume that p is a prime number" (the case of a rectangular flag). The case of "assume..." is essentially different from "let...": in the case of "let..." the p is a new variable, in the case of "assume..." it is a variable or a constant that was introduced earlier in the book (cf. the assumption (89) in Section 18.2).

Unfortunately there is not a very strong feeling in english OMV that "let..." is to be restricted to the case of declaration. One might consider to replace the "let..." by something that cannot be misinterpreted, like "take any prime number p".

In OMV there is no clear way to tell where the flagstaffs end. Quite often it is suggested by the typographical layout, mainly by the subdivision of the text into sentences, paragraphs, sub-sections, sections and chapters. There is certainly a need for explicit rules for this. Right now there are just some unwritten conventions. One might express a rule like this: If an assumption is a part of a sentence, then it does not reach beyond that sentence. If it is the first sentence of a paragraph, and not a paragraph of its own, then it does not reach beyond its paragraph. Similarly, if a sequence of assumptions form the first sentence of a paragraph, and not a whole paragraph, then these assumptions are intended just for this paragraph. The rules for declarations are the same as those for assumptions. As an example we quote

(22.3.1) If x is a real number then $\sin x < 2$.

It is considered to be bad manners to refer to x in the next sentence. Actually one may doubt what (22.3.1) means. It can be

(i) a block, opened by the declaration "x:real number",

(ii) the universal statement $\forall_{x:\text{real number}} \sin x < 2$.

Fortunately the two are equivalent by virtue of the language rules of MV. But sometimes (22.3.1) means an implication: just imagine that the sentence (22.3.1) is preceded by "let x be any complex solution of the equation $4 \cos x - 1 = 0$". In that case we can consider (22.3.1) as an implication, but also as a block (starting with the assumption "let x be a real number"). Fortunately again, the two possibilities are equivalent because of the language rules of MV.

22.4. Expressing quantification in natural language is reasonably established: "for all points $P \cdots$", "for every point $P \cdots$", "there exists a point P such that $\ldots$", are sufficiently clear, also if the quantifier "for all points P" is shifted to the end of the sentence: a machine should be able to translate them into $\forall$'s and $\exists$'s. Nevertheless there is something wrong from the linguistic point of view: the name P does not play a role any ordinary word could ever play in a sentence. Writing "for every point, P say" does not make it any better. Natural language simply does not have anything corresponding to dummy variables! We (and the linguists) just have to learn to live with the strange P in "for every point P".

22.5. In our natural languages it is often possible to express quantification without the use of a dummy. The sentence "all dogs sleep" is equivalent to "for every dog P, P sleeps". This can be done because of the fact that P is the subject of the sentence "P sleeps". In other cases it can be done by means of pronouns. "There is a dog whose master trims its hair every day" will (although it is not very elegant English) mean: "there is a dog d such that the master of d trims d's hair every day".

Correct interpretation of such sentences may depend on subtleties (see what happens if "d's hair" is replaced by "his hair"), in particular in cases of more than one quantification in a single sentence.

Special care should be taken with the words we use when applying the rule for elimination of the existence quantifier, mentioned at the end of Section 20.7. One usually says: "We know the existence of an $x : A$ such that $P(x)$. Take such an x". Then one starts deriving a statement Q (which does not contain x). That is, one derives

$$x : A(\text{decl}),\ P(x)(\text{ass})\ *\ Q$$

and then the existence of an $x : A$ such that $P(x)$ guarantees that Q holds outside the block too.

It would be nice to have a standard way of saying these things in a world where it is not customary to indicate the end of a block. A suggestion: open the argument with "We may assume that we have an x with the property $P(x)$". The word "assume" stresses the fact that the life of x is short!

22.6. The situation with the substantive binder (see T6 in Section 12) is similar to the situation with logical quantifiers. The example in Section 20.4, viz. "quadrilateral with the property that its diagonals are orthogonal to each other" again shows that names of dummies can sometimes be avoided by means of a pronoun (in this case "its"). Another example is "prime number dividing n", where the gerund "dividing n" is derived from the statement "p divides n".

22.7. It is not hard to get good translations for our formalism describing sets. If A is a substantive, we can pronounce "$A\uparrow$" as "the set of all A's". If S is an A-set then the substantive "$S\downarrow$ " can be pronounced as "element of S". So $s : S\downarrow$ is to be pronounced as "s is an element of S", and this amounts to the same thing as "$s \in S$ ".

22.8. Now coming to secondary MV, we of course depend on the special book we prefer to write. If it contains the material of Section 18.2, we first get to the discussion of natural language for negation.

If P is a statement, the statement not(P) can be written as "it is not true that P", and actually such a thing could be written as a book definition like this:

$$P \ :: \text{statement(decl)} \ * \ \text{it is not true that } P := \text{not}(P) \ :: \text{statement}.$$

Nothing much is gained by this, of course (and some people might object that "not true" has too much of a metalinguistic flavor), but it seems to be the only construction that works the same way in all possible cases. In many sentences the negation can be worked into the statement P, usually by putting "does not" in front of the main verb, but that is impossible if P has the form of a quantified statement or of an implication.

22.9. Getting deeper into an MV book, we are no longer dealing with fundamentals, and this may imply that we are mainly sticking to the kind of sentences we have invented ourselves in definitions, especially since it is so easy for us to introduce new terminology in the book.

We discuss the example of line (140) in Section 18.2. In natural language we like to have some synonyms available for a phrase like "b goes through A", and there is no objection against codifying some of these in the MV book. We might insert directly after (140) a definition like

$$A \text{ lies on } b := b \text{ goes through } A \ :: \text{statement}.$$

And we may introduce a new substantive "point of b" as "point lying on b" (the substantive binder with suppressed dummy, cf. Section 22.5). And from now on we have another way to say that "b goes through A" by means of the typing statement "A is a point of b".

22.10. Natural language can have productive rules for getting synonyms. The rules we discuss here are connected with possession, with conjunction and with disjunction.

Possession (to have or not to have) seems to be overwhelmingly important for human beings, and therefore they have made facilities for expressing it in many different ways, so as to fit in every linguistic situation. In the name "the derivative of g" the "of" suggests that g possesses something, and we rapidly accept that "f is the derivative of g" and "g has f as its derivative" are synonymous. Therefore we feel that in the MV book it is sufficient to define "the derivative of g" and that we can take the other constructions as implicitly defined by it. The phrases for describing possession of course depend on the question whether somebody can possess more that one thing of the kind mentioned, and also on the question whether he is or is not the sole proprietor. For example: "A is a point of b" turns into "b has A as one of its points", and the existence statement "exist(point of b)" is synonymous to "b has at least one point".

In conjunctive statements synonyms can lead to shorter sentences. The statement "A lies on b and B lies on b" is synonymous to "A and B lie on b", and similarly "A lies on b and A lies on c" is synonymous to "A lies on b and c". Still the matter is tricky: "a implies c and b implies c" can *not* be contracted to "a and b imply c".

There are similar contractions for disjunction. "A lies on b or on c" is synonymous to "A lies on b or A lies on c".

In general one will go as far as one can with such contractions as long as there is no danger of ambiguities. Try "P is the only point of S and P is the only point of T", "P is a point of the limit set of S and P is a point of the limit set of T".

Quite often synonym production rules are not safe enough for mathematics. "I hear John and Mary" can be considered to be synonymous to "I hear Mary and John". But in "the cartesian product of X and Y" we cannot just interchange the X and the Y. And a rule that "not for all points P we have Q" is synonymous to "there is a point P such that not(Q)" is a thing we want to be able to discuss as a logical rule; we do not want to be forced to accept it for the sake of linguistics.

22.11. From Section 22.10 we may conclude that it is not easy at all to decide about synonym production rules. Forbidding them will make our language inelegant, admitting them makes it unsafe. Obviously the best thing we can do is that as long as we do not fully understand the synonym production rules, we refuse to consider them as a part of "official" MV. We can put them on the list of the "grains of salt", if we wish. In many cases we can take the effects of synonym production rules seriously without proclaiming them as rules. An example of this was presented in Section 18, line (143), where a synonym was provided by a book definition. Many things can be developed that way, although it may become monotonous on the long run.

22.12. It requires quite some mathematical experience to understand sentences involving indefinite articles ("a" and "an"). Quite often such sentences are ambiguous, and their interpretation may depend on whether they are labeled as "definition" or as "theorem". Example:

(22.12.1) a rhombus is a quadrilateral with property P

has three interpretations, viz.

(22.12.2) rhombus $\ll$ quadrilateral with property P

(22.12.3) rhombus := quadrilateral with property P

(22.12.4) rhombus = quadrilateral with property P .

If (22.12.1) is labeled with "theorem" or "lemma", we choose for (22.12.2), but if the label is "definition", we choose (22.12.3). If instead of (22.12.1) we would have had "rhombuses are quadrilaterals with property P", with a theorem-like label, we might have hesitated between (22.12.2) and (22.12.4). If we really want to express (22.12.4) we might prefer "A quadrilateral is a rhombus if and only if it has property P".

We may test our abilities for understanding mathematical sentences by trying cases where some of the words have been replaced by symbols that conceal the meaning of the words. As an example we consider

Definition. An A of a B is a C whose D intersects that B.

This we rapidly interpret as

$$x : B \text{ (dcl)} \;*\; A \text{ of } x :=$$

$$:= S_{y:C}((\text{the } D \text{ of } y) \text{ intersects } x) \;::\; \text{substantive} .$$

A second example is

Definition. We say that the A of a B hibernates if it is skew to that B".

This we interpret as

$$x : B \text{ (dcl)} \;*\; \text{the } A \text{ of } x \text{ hibernates} :=$$

$$:= \text{the } A \text{ of } x \text{ is skew to } x \;::\; \text{statement} ,$$

at least if the terms "the A of x" and "is skew to" have been defined in the book.

There is an objection against the choice of the phrase "the A of x hibernates" in the above definition. It asks for trouble with parsing. Let us be more specific by taking as an example: "We say that the orthocenter of triangle d hibernates if it lies inside d". Now take two triangles d and e with common orthocenter P. We might argue: if P lies inside d then P hibernates, and therefore P lies inside e. This is obviously false! The thing is that there never has been a definition explaining what it means that a point hibernates. Nevertheless this is suggested by the phrase "We say that the orthocenter of d hibernates", since "the orthocenter of d" is the name of a point. Therefore it is in vain to appeal to EQ10a (Section 13) in order to say that in this phrase "the orthocenter of d" may be replaced by "the orthocenter of e". We cannot say that

$$t : \text{point (dcl)} * t \text{ hibernates } :: \text{statement} .$$

A way to avoid this inconvenience is to define "the orthocenter of triangle d hibernates with respect to d". Or, still simpler, we define hibernation of a point with respect to a triangle, and then apply it to the orthocenter.

We note that the definite article "the" is used in two different ways. In a case like "the orthocenter of d" it originates from a line in the book where in a context "d:triangle (dcl)" we have defined the name "the orthocenter of d". But a case like "the positive root of f" has to be parsed as a substantive ("positive root of f") preceded by the "the" of Section 18.2 (83), which requires a proof of the uniqueness statement (82).

In the case of "the orthocenter of d" there has been no previous introduction of a substantive "orthocenter of d", and therefore it can not be parsed as a substantive preceded by a definite article.

22.13. A line in an MV book can be labeled "theorem" or "lemma" if the line body is a statement (case (vi) of BR9 (Section 10)). That is, if it is considered "important" enough for such a stately name. Otherwise it can just be considered as a stepping stone in a proof, or as a minor remark.

Lines with a body of the form (iii), (iv) or (viii) of BR9, can be labeled "definition", although very often (in particular in case (iii)) we prefer to call them abbreviations. These cases (iii), (iv), (vii) can be called "statement definition", "substantive definition" and "name definition", respectively, and each case has its own phraseology in OMV.

Very many definitions in OMV are definitions of adjectives; we shall discuss the use of adjectives in Section 22.14.

22.14. In our grammar of MV we have not discussed adjectives thus far, but they are easily incorporated. We should always bear in mind that an adjective is to be defined with respect to a substantive. Let us take the substantive "triangle" as an example. If

$$x : \text{triangle (dcl)} \ * \ P \ :: \text{statement}$$

is valid, then P (which may be an expression containing x) expresses a property of x. In natural language such a property may be expressed by an adjective. Let us choose the word "blue" for it. Then we can consider the new substantive "blue triangle" and the new statement "x is blue". This statement can be considered as an abbreviation for "x is a blue triangle". The substantive can be defined as

(22.14.1) $\text{blue triangle} := S_{x:\text{triangle}}\, P \ :: \text{substantive}\ .$

The statement "x is blue" can be introduced by

$$x : \text{triangle (dcl)} \ * \ x \text{ is blue} \ := \ (x : \ \text{blue triangle}) \ :: \text{statement}\ .$$

We can agree that we express both (22.14.1) and (22.14.2) by the single line

Definition. A triangle x is called blue when P.

A nice way to write this with a new binder "Adj" is as follows:

(22.14.3) $\text{blue} := \text{Adj}_{x:\text{triangle}}\, P\ .$

Working with adjectives has some other nice features. One is, that if an adjective like "yellow" is defined with respect to the substantive A, and if $B \ll A$, then we can speak of yellow B's too. And if we have defined both "yellow" and "round" on the substantive A, then we can use the new substantive "yellow round A", and that is synonymous with "round yellow A". Misunderstandings can arise if the adjective "round" was not defined with respect to the substantive A but with respect to the substantive "yellow A".

23. REMARKS ON PARSING

The situation about parsing is like this. What we really want to say in a sentence or a formula, has the structure of a tree (to be more precise, of a planted planar tree). Such a tree is a finite directed graph where (i) every point has just one incoming edge (except for a single point, the root, which has none), (ii) at each point a linear order of the outgoing edges is given (the order from left to right), (iii) every point can be reached by means of a finite path that starts at the root. Finally we mention that to the points of the tree there may be attached letters or words, i.e., identifiers.

The difficulty is that we want to put such tree-shaped information in a linear form, in order to be expressed in speech or writing, and that we want this linearized form to reveal the original tree structure. Mathematicians have solved this problem centuries ago, coding their trees in linearized form with the aid of sets of nested pairs of parentheses. In natural languages, however, this has never been done. It is quite probable that parsing trouble had its influence on our natural languages before writing was invented, in particular in the shaping of inflexions and conjugations. Quite some effort in learning languages, and in studying their structure, is connected with parsing and with the constructs people invented for the benefit of parsability.

In spite of the fact that parsing is an immense problem in the study of natural languages, we can be very casual about it when discussing MV. As long as we are able to *create* a language it is no serious problem at all. We can just be generous with the use of parentheses or any other means for describing the tree structure in a linear format. Admittedly, we do not want to go all the way: we want our MV to look like natural language as much as possible. This can be achieved to a large extent by sensible choice of the terms and phrases we introduce in our books (in the form of sMV material). Combined with the use of just a few parentheses, this can reduce parsing trouble to a bearable extent. Making a serious study of this would, for the moment, be a waste of time, since the trouble can so easily be eliminated by adding enough parentheses.