MathLang: A language for Mathematics

Fairouz Kamareddine

www.macs.hw.ac.uk/~fairouz/talks/talks2004/bham04.pdf*

Friday 28 May 2004

*Parts of this talk are based on joint work with Laan and Nederpelt (see [18, 19]) and Maarek and Wells (see [20, 21])

University of Birmingham

Do we need types for Mathematics?

- *General definition of function 1879* [9] is key to Frege's *formalisation of logic*.
- For Frege, *mathematics can be seen as a branch of logic.*
- *Self-application of functions* was at the heart of *Russell's paradox 1902* [28].
- To *avoid paradox* Russell controled function application via *type theory*.
- Russell [29] *1903* gives the first type theory: the *Ramified Type Theory* (RTT).
- But, *type theory* existed since the time of *Euclid* (325 B.C.).
- RTT is used in Russell and Whitehead's Principia Mathematica [32] 1910–1912.
- *Simple theory of types* (STT): Ramsey [26] *1926*, Hilbert and Ackermann [17] *1928*.

- Church's *simply typed λ-calculus* $\lambda{\to}$ [5] 1940 = λ-calculus + STT.
- Church's $\lambda{\to}$ is at the heart of *Montague's semantics* of natural language [8], of the *HOL* theorem prover, and of many programming languages.
- The hierarchies of types/orders as found in RTT and STT are *unsatisfactory*.
- Frege's functions $\neq$ Principia's functions $\neq$ *λ-calculus* functions (1932).
- The *notion of function adopted in the λ-calculus* is *unsatisfactory* (cf. [22]).
- Hence, birth of *different systems of functions and types*, each with *different functional power*.
- Advances were also made in *set theory* [33], *category theory* [23], etc., each being advocated as a better foundation for mathematics.
- But, why have mathematicians not taken more notice?

Prehistory of Types (Euclid)

- Euclid's *Elements* (circa 325 B.C.) begins with:

  1. A *point* is that which has no part;
  2. A *line* is breadthless length.

     ⋮

  15. A *circle* is a plane figure contained by one line such that all the straight lines falling upon it from one point among those lying within the figure are equal to one another.

- 1..15 *define* points, lines, and circles which Euclid *distinguished* between.

- Euclid always mentioned to which *class* (points, lines, etc.) an object belonged.

Prehistory of Types (Euclid)

- By distinguishing classes of objects, Euclid prevented *undesired/impossible* situations. E.g., whether two points (instead of two lines) are parallel.
- Intuition implicitly forced Euclid to think about the *type* of the objects.
- As intuition does not support the notion of parallel points, he did not even *try* to undertake such a construction.
- In this manner, types have always been present in mathematics, although they were not noticed explicitly until the late 1800s.
- If you studied geometry, then you have an (implicit) understanding of types.

Prehistory of Types (Paradox Threats)

- From 1800, mathematical systems became less intuitive, for several reasons:
  - Very *complex* or abstract systems.
  - *Formal* systems.
  - Something with *less intuition* than a human using the systems: a *computer* or an *algorithm*.
- These situations are *paradox threats*. An example is Frege's Naive Set Theory.
- *Not enough intuition* to activate the (implicit) type theory *to warn against an impossible situation*.

Prehistory of Types (formal systems in 19th century)

In the 19th century, the *need for a more precise* style in mathematics arose, *because controversial results* had appeared in *analysis*.

- 1821: Many of these controversies were solved by the work of Cauchy. E.g., he introduced *a precise definition of convergence* in his *Cours d'Analyse* [4].

- 1872: Due to the more *exact definition of real numbers* given by Dedekind [7], the rules for reasoning with real numbers became even more precise.

- 1895-1897: Cantor began formalizing *set theory* [2, 3] and made contributions to *number theory*.

Prehistory of Types (formal systems in 19th century)

- 1889: *Peano* formalized *arithmetic* [24], but did not treat logic or quantification.

- 1879: *Frege* was not satisfied with the use of *natural language in mathematics*:

  > ". . . I found *the inadequacy of language to be an obstacle*; no matter how unwieldy the expressions I was ready to accept, I was less and less able, as the relations became more and more complex, to attain the precision that my purpose required."
  >
  > (*Begriffsschrift*, Preface)

  Frege therefore presented *Begriffsschrift* [9], the first formalisation of logic giving logical concepts via symbols rather than natural language.

Prehistory of Types (formal systems in 19th century)

> "[Begriffsschrift's] first purpose is to *provide us with the most reliable test of the validity of a chain of inferences* and to point out *every presupposition* that tries to sneak in unnoticed, so that its origin *can be investigated*."
>
> (*Begriffsschrift*, Preface)

- 1892-1903 Frege's *Grundgesetze der Arithmetik* [11, 15], could handle elementary arithmetic, set theory, logic, and quantification.

Prehistory of Types (Begriffsschrift's functions)

The introduction of a *very general definition of function* was the key to the formalisation of logic. Frege defined the Abstraction Principle.

Abstraction Principle 1.

> *"If in an expression, [. . .] a simple or a compound* sign has one or more occurrences *and if we regard that sign as* replaceable *[. . .] by something else (but everywhere by the same thing), then we call* the part that remains invariant in the expression *a* function, *and* the replaceable part *the* argument *of the function."*

(Begriffsschrift, Section 9*)*

Prehistory of Types (Begriffsschrift's functions)

- Frege put *no restrictions* on what could play the role of *an argument*.
- An argument could be a *number* (as was the situation in analysis), but also a *proposition*, or a *function*.
- Similarly, the *result of applying* a function to an argument did not necessarily have to be a number.
- Functions of more than one argument were constructed by a method that is very close to the method presented by Schönfinkel [31] in 1924.

Prehistory of Types (Begriffsschrift's functions)

With this definition of function, two of the three possible *paradox threats occurred*:

1. The generalisation of the concept of function made the system more abstract and *less intuitive*.

2. Frege introduced a *formal* system instead of the informal systems that were used up till then.

Type theory, that would be helpful in distinguishing between the different types of arguments that a function might take, *was left informal*.

So, *Frege had to proceed with caution*. And so he did, at this stage.

Prehistory of Types (Begriffsschrift's functions)

Frege was *aware* of some typing rule that does *not* allow to *substitute functions for object variables or objects for function variables*:

> "if the [. . .] letter [sign] occurs as a function sign, this circumstance [should] be taken into account."
>
> (*Begriffsschrift*, Section 11)

> " Now just as functions are fundamentally different from objects, so also *functions whose arguments are and must be functions* are fundamentally different from *functions whose arguments are objects and cannot be anything else*. I call the latter *first-level*, the former *second-level*."
>
> (*Function and Concept*, pp. 26–27)

Prehistory of Types (Begriffsschrift's functions)

In *Function and Concept* he was aware of the fact that making a *difference between first-level and second-level objects is essential to prevent paradoxes*:

> "The ontological proof of God's existence suffers from the fallacy of treating existence as a first-level concept."
>
> (*Function and Concept*, p. 27, footnote)

The above discussion on functions and arguments shows that *Frege did indeed avoid the paradox in his Begriffsschrift*.

Prehistory of Types (Grundgesetze's functions)

The *Begriffsschrift*, however, was only a prelude to Frege's writings.

- In Grundlagen der Arithmetik [10] he argued that mathematics can be seen as a branch of logic.

- In Grundgesetze der Arithmetik [11, 15] he described the elementary parts of arithmetic within an extension of the logical framework of *Begriffsschrift*.

- Frege approached the *paradox threats for a second time* at the end of Section 2 of his *Grundgesetze*.

- He did *not* want to *apply a function to itself*, but to its course-of-values.

Prehistory of Types (Grundgesetze's functions)

"the function $\Phi(x)$ has the same *course-of-values* as the function $\Psi(x)$" if:

> "the functions $\Phi(x)$ and $\Psi(x)$ always have the same value for the same argument."
>
> (*Grundgesetze*, p. 7)

- This is expressed as

$$\grave{\varepsilon} f(\varepsilon) = \grave{\varepsilon} g(\varepsilon) \longleftrightarrow \forall a[f(a) = g(a)]. \tag{1}$$

In modern terminology, we could say that the functions $\Phi(x)$ and $\Psi(x)$ have the same course-of-values if they have the same *graph*.

Prehistory of Types (Grundgesetze's functions)

- The notation $\grave{\varepsilon}\Phi(\varepsilon)$ may be the *origin* of Russell's notation $\hat{x}\Phi(x)$ for the class of objects that have the property Φ.

- According to a paper by Rosser [27], the notation $\hat{x}\Phi(x)$ has been at the *basis* of the current notation $\lambda x.\Phi(x)$.

- Church is supposed to have written $\wedge x\Phi(x)$ for the function $x \mapsto \Phi(x)$: the hat $\wedge$ in front of the x distinguishes this function from the class $\hat{x}\Phi(x)$.

Prehistory of Types (Grundgesetze's functions)

- Frege treated *courses-of-values* as *ordinary objects*.
- As a consequence, *a function that takes objects as arguments could have its own course-of-values as an argument*.
- *BUT,* all essential information of a function is contained in its graph.
- A system in which a function can be applied to its own graph should have similar possibilities as a system in which a function can be applied to itself.
- Frege *excluded the paradox threats* by *forbidding self-application*
- but due to his *treatment of courses-of-values* these threats were able to *enter his system through a back door*.

Prehistory of Types (Russell's paradox in *Grundgesetze*)

- In 1902, Russell wrote a letter to Frege [28], informing him that he had *discovered a paradox* in his *Begriffsschrift.*

- *WRONG: Begriffsschrift does not suffer from a paradox*.

- Russell gave his well-known argument, defining the propositional function

$$f(x) \text{ by } \neg x(x).$$

  In Russell's words: "*to be a predicate that cannot be predicated of itself.*"

- Russell assumed $f(f)$. Then by definition of f, $\neg f(f)$, *a contradiction*. Therefore: $\neg f(f)$ holds. But then (again by definition of f), $f(f)$ holds. Russell concluded that *both* $f(f)$ *and* $\neg f(f)$ hold, a *contradiction*.

Prehistory of Types (Russell's paradox in *Grundgesetze*)

- *6 days later*, Frege wrote [14] that *Russell's derivation of paradox is incorrect*.
- Ferge explained that *self-application $f(f)$ is not possible in* Begriffsschrift.
- *$f(x)$ is a function, which requires an* object *as an argument.*
  A function cannot be an object in the Begriffsschrift.
- Frege explained that *Russell's argument could be amended to a paradox* in Grundgesetze, using the *course-of-values* of functions:

  Let $f(x) = \neg\forall\varphi[(\grave{\alpha}\varphi(\alpha) = x) \longrightarrow \varphi(x)]$
  I.e. $f(x) = \exists\varphi[(\grave{\alpha}\varphi(\alpha) = x) \wedge \neg\varphi(x)]$ hence $\neg\varphi(\grave{\alpha}\varphi(\alpha))$

- *Both* $f(\grave{\varepsilon}f(\varepsilon))$ and $\neg f(\grave{\varepsilon}f(\varepsilon))$ hold.
- Frege added an appendix of 11 pages to the 2nd volume of *Grundgesetze* in which he gave a very detailed description of the paradox.

Prehistory of Types (How wrong was Frege?)

- Due to Russell's Paradox, Frege is often depicted as the pitiful person whose system was inconsistent.
- This suggests that Frege's system was the only one that was inconsistent, and that Frege was very inaccurate in his writings.
- On these points, history does Frege an injustice.
- Frege's system was much more accurate than other systems of those days.
- Peano's work, for instance, was *less precise* on several points:
- Peano *hardly paid attention to logic* especially quantification theory;
- Peano *did not make a strict distinction* between his *symbolism* and the *objects underlying this symbolism*. Frege was much more accurate on this point (see Frege's paper *Über Sinn und Bedeutung* [12]);

Prehistory of Types (How wrong was Frege?)

- Frege *made a strict distinction* between a *proposition* (as an object) and the assertion *of a proposition*. Frege denoted a *proposition*, by $-A$, and *its assertion* by $\vdash A$. Peano did not make this distinction and simply wrote A.

Nevertheless, Peano's work was very popular, for several reasons:

- Peano had *able collaborators*, and a *better eye for presentation and publicity*.

- Peano bought *his own press* to supervise the printing of his own journals Rivista di Matematica and Formulaire [25]

Prehistory of Types (How wrong was Frege?)

- Peano used a *familiar symbolism* to the notations used in those days.
- Many of *Peano's notations*, like $\in$ for "is an element of", and $\supset$ for logical implication, are used in *Principia Mathematica*, and are actually still in use.
- *Frege's work* did not have these advantages and *was hardly read before 1902*
- When *Peano* published his formalisation of mathematics in 1889 [24] he clearly *did not know* Frege's *Begriffsschrift* as he did not mention the work, and *was not aware* of Frege's formalisation of quantification theory.

Prehistory of Types (How wrong was Frege?)

- Peano considered quantification theory to be "abstruse" in [25]:

  "In this respect my *[Frege] conceptual notion of 1879 is superior to the Peano one*. Already, at that time, I specified all the laws necessary for my designation of generality, so that nothing fundamental remains to be examined. These laws are few in number, and *I do not know why they should be said to be abstruse*. If it is otherwise with the *Peano conceptual notation*, then this is due to the *unsuitable* notation."

  ([13], p. 376)

Prehistory of Types (How wrong was Frege?)

- In the last paragraph of [13], Frege concluded:

  ". . . I observe merely that the *Peano notation* is unquestionably *more convenient for the typesetter*, and in many cases *takes up less room* than mine, but that these advantages seem to me, due to the inferior perspicuity and *logical defectiveness*, to have been paid for too dearly — at any rate for the purposes I want to pursue."

  (*Ueber die Begriffschrift des Herrn Peano und meine eigene*, p. 378)

Prehistory of Types (paradox in Peano and Cantor's systems)

- Frege's system was *not the only paradoxical* one.
- The Russell Paradox can be derived in *Peano's system* as well, by defining the class $K \stackrel{\text{def}}{=} \{x \mid x \notin x\}$ and deriving $K \in K \longleftrightarrow K \notin K$.
- In *Cantor's Set Theory* one can derive the paradox via the same class (or *set*, in Cantor's terminology).

Prehistory of Types (vicious circle principle)

When Russell proved Frege's *Grundgesetze* to be inconsistent, Frege was not the only person in *trouble*. In Russell's letter to Frege (1902), we read:

> "I am on the point of finishing a book on the principles of mathematics"
>
> (*Letter to Frege*, [28])

Russell *had to find a solution* to the paradoxes, before finishing his book.

- Type Theory was the *tool* for avoiding the paradoxes.
- Types in *Principia* have a double hierarchy: *(simple) types* and *orders*.
- Ramsey/Hilbert and Ackermann removed orders and gave us the simple types.
- Recall that the idea behind simple types was already explained by Frege.

The Goal: Open borders between mathematics, logic and computation

- Ordinary mathematicians *avoid* formal mathematical logic.
- Ordinary mathematicians *avoid* proof checking (via a computer).
- Ordinary mathematicians *may use* a computer for computation: there are over 1 million people who use mathematica (including linguists, engineers, etc.).
- Mathematicians may also use other computer forms like Maple, Latex, etc.
- But we are not interested in only *libraries* or *computation* or *text editing*.
- We want *freedeom of movement* between mathematics, logic and computation.
- At every stage, we must have *the choice* of the level of formalilty and the depth of computation.

Common Mathematical Language of mathematicians: CML

+ CML is *expressive*: it has linguistic categories like *proofs* and *theorems*.
+ CML has been refined by intensive use and is rooted in *long traditions*.
+ CML is *approved* by most mathematicians as a communication medium.
+ CML *accommodates many branches* of mathematics, and is adaptable to new ones.
– Since CML is based on natural language, it is *informal* and *ambiguous*.
– CML is *incomplete:* Much is left implicit, appealing to the reader's intuition.
– CML is *poorly organised:* In a CML text, many structural aspects are omitted.
– CML is *automation-unfriendly:* A CML text is a plain text and cannot be easily automated.

A CML-text

From chapter 1, § 2 of E. Landau's *Foundations of Analysis* [Lan51].

Theorem 6. [Commutative Law of Addition]

$$x + y = y + x.$$

Proof Fix y, and let $\mathfrak{M}$ be the set of all x for which the assertion holds.
I) We have

$$y + 1 = y',$$

and furthermore, by the construction in the proof of Theorem 4,

$$1 + y = y',$$

so that

$$1 + y = y + 1$$

and 1 belongs to $\mathfrak{M}$.
II) If x belongs to $\mathfrak{M}$, then

$$x + y = y + x,$$

Therefore

$$(x + y)' = (y + x)' = y + x'.$$

By the construction in the proof of Theorem 4, we have

$$x' + y = (x + y)',$$

hence

$$x' + y = y + x',$$

so that x' belongs to $\mathfrak{M}$. The assertion therefore holds for all x. □

LaTeX code

draft documents	✓
public documents	✓
computations and proofs	✗

```
\begin{theorem}[Commutative Law of Addition] \label{theorem:6}
  $$x+y=y+x.$$
\end{theorem}
\begin{proof}
  Fix $y$, and $\mathfrak{M}$ be the set of all $x$ for which the
  assertion holds.
  \begin{enumerate}
  \item We have $$y+1=y',$$
    and furthermore, by the construction in
    the proof of Theorem~\ref{theorem:4}, $$1+y=y',$$
    so that $$1+y=y+1$$
    and $1$ belongs to $\mathfrak{M}$.
  \item If $x$ belongs to $\mathfrak{M}$, then $$x+y=y+x,$$
    Therefore
    $$(x+y)'=(y+x)'=y+x'.$$
    By the construction in the proof of
    Theorem~\ref{theorem:4}, we have $$x'+y=(x+y)',$$
    hence
    $$x'+y=y+x',$$
    so that $x'$ belongs to $\mathfrak{M}$.
  \end{enumerate}
  The assertion therefore holds for all $x$.
\end{proof}
```

The problem with formal logic

- No logical languages has the criteria expected of a language of mathematics.
  - A logical language does not have *mathematico-linguistic* categories, is *not universal* to all mathematicians, and is *not a good communication medium*.
  - Logical languages make fixed choices (*first versus higher order, predicative versus impredicative, constructive versus classical, types or sets*, etc.). But different parts of mathematics need different choices and there is no universal agreement as to which is the best formalism.
  - A logician writes in logic their *understanding* of a mathematical-text as a formal, complete text which is structured considerably *unlike* the original, and is of little use to the *ordinary* mathematician.
  - Mathematicians do not want to use formal logic and have *for centuries* done mathematics without it.

- *So, mathematicians kept to* CML.

- We would like to find an alternative to CML which avoids some of the features of the logical languages which made them unattractive to mathematicians.

The problem with fully checked proofs (on computer)

- In 1967 the famous mathematician de Bruijn began work on logical languages for complete books of mathematics that can be *fully* checked by machine.
- People are prone to error, so if a machine can do proof checking, we expect fewer errors.
- Most mathematicians doubted de Bruijn could achieve success, and computer scientists had no interest at all.
- However, he persevered and built *Automath* (AUTOmated MATHematics).
- Today, there is much interest in many approaches to proof checking for verification of computer hardware and software.
- Many theorem provers have been built to mechanically check mathematics and computer science reasoning (e.g. Isabelle, HOL, Coq, etc.).

- A CML-text is structured differently from a computer-checked text proving the same facts. *Making the latter involves extensive knowledge and many choices:*
  - First, the needed choices include:
    * The choice of the *underlying logical system.*
    * The choice of *how concepts are implemented* (equational reasoning, equivalences and classes, partial functions, induction, etc.).
    * The choice of the *formal system*: a type theory (dependent?), a set theory (ZF? FM?), a category theory? etc.
    * The choice of the *proof checker*: Automath, Isabelle, Coq, PVS, Mizar...
  - Any informal reasoning in a CML-text will cause headaches as it is hard to turn a big step into a (series of) syntactic proof expressions.
  - Then the CML-text is *reformulated* in a fully *complete* syntactic formalism where every detail is spelled out. Very long expressions replace a clear CML-text. The new text is useless to ordinary mathematicians.

- So, *automation is user-unfriendly* for the mathematician/computer scientist.

- It is the hope that the alternative to CML may help in dividing the jump from informal mathematics to a fully formal one into smaller more informed steps.

Coq

draft documents	✗
public documents	✗
computations and proofs	✓

From Module `Arith.Plus` of Coq standard library (`http://coq.inria.fr/`).

```
Lemma plus_sym : (n,m:nat)(n+m)=(m+n).
Proof.
Intros n m ; Elim n ; Simpl_rew ; Auto with arith.
Intros y H ; Elim (plus_n_Sm m y) ; Simpl_rew ; Auto with arith.
Qed.
```

Where do we start? de Bruijn's Mathematical Vernacular MV

- *De Bruijn's Automath not just [...] as a technical system for verification of mathematical texts, it was rather a life style with its attitudes towards understanding, developing and teaching mathematics....The way mathematical material is to be presented to the system should correspond to the usual way we write mathematics. The only things to be added should be details that are usually omitted in standard mathematics.*

- MV is faithful to CML yet is formal and avoids ambiguities.

- MV is close to the usual way in which mathematicians write.

- MV has a syntax based on linguistic categories not on set/type theory.

- MV is weak as regards correctness: the rules of MV mostly concern *linguistic* correctness, its types are mostly linguistic so that the formal translation into MV is satisfactory *as a readable, well-organized text*.

Problems with MV

- MV makes many logical and mathematical choices which are best postponed.
- MV incorporates certain correctness requirements, there is for example a hierarchy of types corresponding with sets and subsets.
- MV is already *on its way* to a full formalization, while we want to remain *closer to* a given informal mathematical content.
- We want a *formal* language MathLang which •has the advantages of CML but not its disadvantages and •respects CML content.
- *MV does not respect* CML *content.*

What is the aim for MathLang?

Can we formalise a CML text avoiding as much as possible the ambiguities of natural language while still guaranteeing the following four goals?

1. The formalised text looks very much like the original CML text (and hence the content of the original CML text is respected).

2. The formalised text can be fully manipulated and searched.

3. Steps can be made to do computation (via computer algebra systems) and proof checking (via proof checkers) on the formalised text.

4. This formalisation of text is as simple a process to the mathematician as LaTeX is.

Starting point for MathLang: MV and WTT

- MV is the driving force behind MathLang. But MV fails on goal 1.
- Weak Type Theory, WTT [19], is MV minus the added logic.
- Although WTT succeeds in many ways and is a considerable improvement on MV, it still fails on goal 1. A WTT text is not close to its CML original.
- MathLang starts from WTT, extends its syntax and adds natural language as a top level.
- A *MathLang text remains* close *to its* CML *original and hence yields* reliable *formalizations.*
  The CML-text is covered exactly in its formal version in the MathLang-text.
  One can easily check this.
- We are using MathLang to translate two CML-books [Lan51, Hea56]

MathLang

draft documents	✓
public documents	✓
computations and proofs	✓

- MathLang describes the grammatical and reasoning structure of mathematical texts
- A *weak type system* checks MathLang documents at a grammatical level
- MathLang eventually should support *all encoding uses*

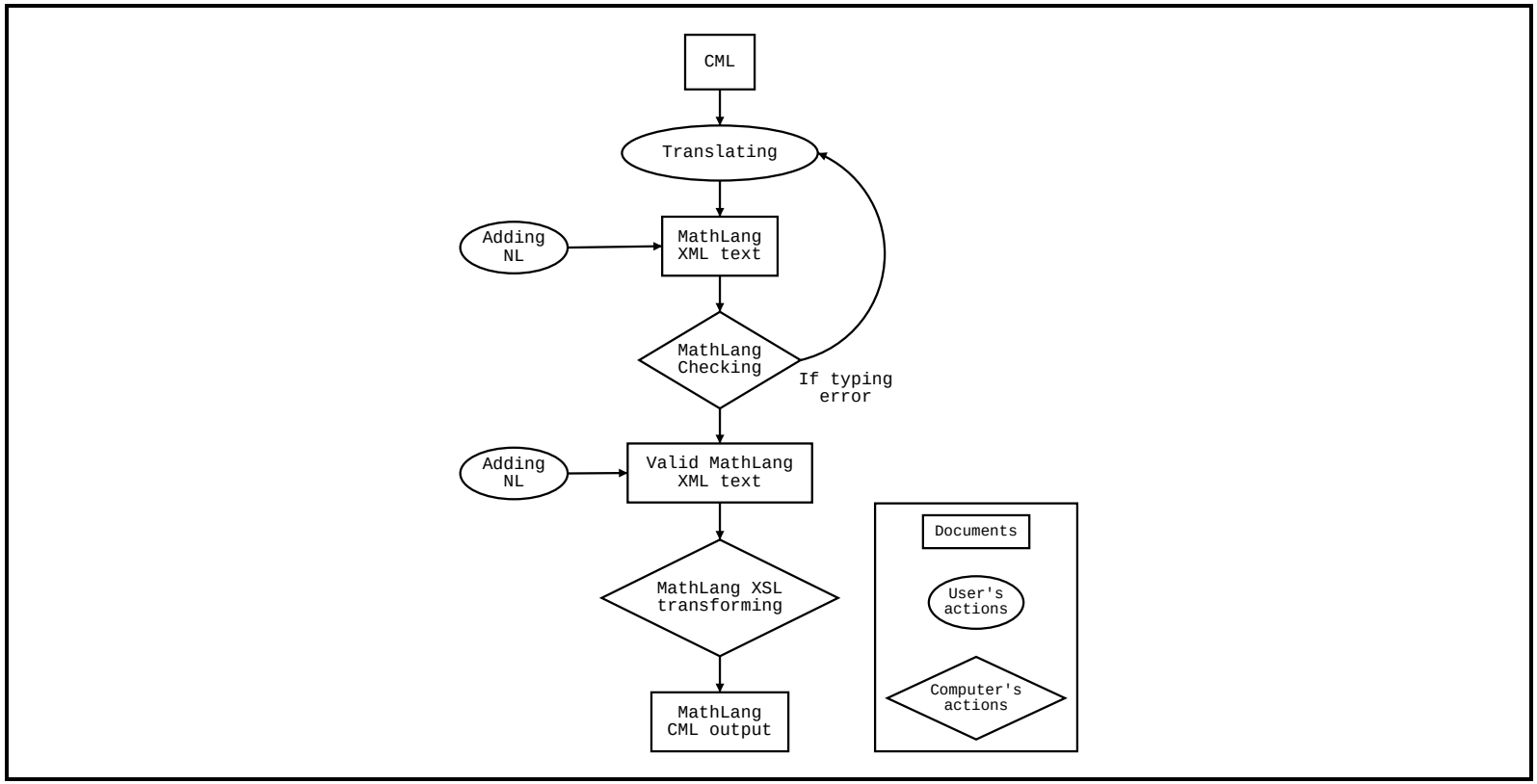


Figure 1: Translation

Weak Type Theory

In Weak Type Theory (or WTT) we have the following linguistic categories:

- On the *atomic* level: *variables*, *constants* and *binders*,
- On the *phrase* level: *terms* $\mathcal{T}$, *sets* $\mathbb{S}$, *nouns* $\mathcal{N}$ and *adjectives* $\mathcal{A}$,
- On the *sentence* level: *statements* P and *definitions* $\mathcal{D}$,
- On the *discourse* level: *contexts* $\mathbb{\Gamma}$, *lines* $\mathbf{l}$ and *books* $\mathbf{B}$.

level	category	abstract syntax	symbol
atomic	*variables*	$\mathtt{V} = \mathtt{V}^T \vert \mathtt{V}^S \vert \mathtt{V}^P$	x
	constants	$\mathtt{C} = \mathtt{C}^T \vert \mathtt{C}^S \vert \mathtt{C}^N \vert \mathtt{C}^A \vert \mathtt{C}^P$	c
	binders	$\mathtt{B} = \mathtt{B}^T \vert \mathtt{B}^S \vert \mathtt{B}^N \vert \mathtt{B}^A \vert \mathtt{B}^P$	b
phrase	*terms*	$T = \mathtt{C}^T(\overrightarrow{\mathcal{P}}) \vert \mathtt{B}^T_{\mathcal{Z}}(\mathcal{E}) \vert \mathtt{V}^T$	t
	sets	$\mathbb{S} = \mathtt{C}^S(\overrightarrow{\mathcal{P}}) \vert \mathtt{B}^S_{\mathcal{Z}}(\mathcal{E}) \vert \mathtt{V}^S$	s
	nouns	$\mathcal{N} = \mathtt{C}^N(\overrightarrow{\mathcal{P}}) \vert \mathtt{B}^N_{\mathcal{Z}}(\mathcal{E}) \vert \mathcal{A}\mathcal{N}$	n
	adjectives	$\mathcal{A} = \mathtt{C}^A(\overrightarrow{\mathcal{P}}) \vert \mathtt{B}^A_{\mathcal{Z}}(\mathcal{E})$	a
sentence	*statements*	$P = \mathtt{C}^P(\overrightarrow{\mathcal{P}}) \vert \mathtt{B}^P_{\mathcal{Z}}(\mathcal{E}) \vert \mathtt{V}^P$	S
	definitions	$\mathcal{D} = \mathcal{D}^{\varphi} \vert \mathcal{D}^P$	D
		$\mathcal{D}^{\varphi} = \mathtt{C}^T(\overrightarrow{V}) := T \vert \mathtt{C}^S(\overrightarrow{V}) := \mathbb{S} \vert$ $\mathtt{C}^N(\overrightarrow{V}) := \mathcal{N} \vert \mathtt{C}^A(\overrightarrow{V}) := \mathcal{A}$	
		$\mathcal{D}^P = \mathtt{C}^P(\overrightarrow{V}) := P$	
discourse	*contexts*	$\mathbb{\Gamma} = \emptyset \mid \mathbb{\Gamma}, \mathcal{Z} \mid \mathbb{\Gamma}, P$	Γ
	lines	$\mathbf{l} = \mathbb{\Gamma} \triangleright P \mid \mathbb{\Gamma} \triangleright \mathcal{D}$	l
	books	$\mathbf{B} = \emptyset \mid \mathbf{B} \circ \mathbf{l}$	B

Figure 2: Main categories of syntax of WTT

Other category	abstract syntax	symbol
expressions	$\mathcal{E} = T \vert \mathbb{S} \vert \mathcal{N} \vert P$	E
parameters	$\mathcal{P} = T \vert \mathbb{S} \vert P$ (note: $\overrightarrow{\mathcal{P}}$ is a list of $\mathcal{P}$s)	P
typings	$\mathbf{T} = \mathbb{S} :$ SET $\vert \mathcal{S} :$ STAT $\vert T : \mathbb{S} \vert T : \mathcal{N} \vert T : \mathcal{A}$	T
declarations	$\mathcal{Z} = \mathtt{V}^S :$ SET $\vert \mathtt{V}^P :$ STAT $\vert \mathtt{V}^T : \mathbb{S} \vert \mathtt{V}^T : \mathcal{N}$	Z

Figure 3: Categories of syntax of WTT

Constants of WTT

$\mathtt{C} = \mathtt{C}^T|\mathtt{C}^S|\mathtt{C}^N|\mathtt{C}^A|\mathtt{C}^P$.

$\mathtt{C}$ is infinite, $\mathtt{C} \cap \mathtt{V} = \emptyset$.

$(\mathtt{C}^T)$	Constants for $terms$,	$(\mathtt{C}^S)$	Constants for $sets$,
$(\mathtt{C}^N)$	Constants for $nouns$,	$(\mathtt{C}^A)$	Constants for $adjectives$,
$(\mathtt{C}^P)$	Constants for $statements$		

A constant is always $\mathtt{C}(\vec{\mathcal{P}})$. $\mathcal{P}$ is term, set or statement: $\mathcal{P} = T|\mathbb{S}|P$.

Examples of constants of WTT

($\mathtt{C}^T$) Constants for *terms*:

π,

the centre of C,

$3 + 6$,

the arithmetic mean of 3 and 6,

$d(x, y)$,

$\nabla\ f$.

($\mathtt{C}^S$) Constants for *sets*:

$\mathbb{N}$,

$V \to W$,

$A \cup B$.

($\mathtt{C}^N$) Constants for *nouns*:
a triangle,
an eigenvalue of A,
an edge of $\triangle ABC$,
a reflection of V *with respect to* l.

($\mathtt{C}^A$) Constants for *adjectives*
prime
surjective
Abelian
continuous on $[a, b]$.

($\mathtt{C}^P$) Constants for *statements*
P lies between Q and R,
$5 \geq 3$ $p \wedge q$ $\neg\, \forall_{x \in \mathbb{N}}(x > 0)$

Binders of WTT

$\mathtt{B} = \mathtt{B}^T|\mathtt{B}^S|\mathtt{B}^N|\mathtt{B}^A|\mathtt{B}^P$ where:

- $(\mathtt{B}^T)$ Binders giving *terms*,
- $(\mathtt{B}^S)$ Binders giving *sets*,
- $(\mathtt{B}^A)$ Binders giving *adjectives*,
- $(\mathtt{B}^P)$ Binders giving *statements*,
- $(\mathtt{B}^N)$ Binders giving *nouns*,

In $\mathtt{B}_{\mathcal{Z}}(\mathcal{E})$, the body $\mathcal{E}$ is one of four categories $\mathcal{E} = T|\mathbb{S}|\mathcal{N}|P$.

Examples:

- $\mathtt{B}^T_{\mathcal{Z}}(\mathcal{E}) = \min_{\mathcal{Z}}(T)|\sum_{\mathcal{Z}}(T)|\lim_{\mathcal{Z}}(T)|\int_{\mathcal{Z}}(T)|\lambda_{\mathcal{Z}}(T)|\lambda_{\mathcal{Z}}(\mathbb{S})|\iota_{\mathcal{Z}}(P)|\ldots$
- $\mathtt{B}^S_{\mathcal{Z}}(\mathcal{E}) = \mathtt{Set}_{\mathcal{Z}}(P)|\bigcup_{\mathcal{Z}}(\mathbb{S})|\iota_{\mathcal{Z}}(P)|\ldots$
- $\mathtt{B}^N_{\mathcal{Z}}(\mathcal{E}) = \mathtt{Noun}_{\mathcal{Z}}(P)|\mathtt{Abst}_{\mathcal{Z}}(T)|\mathtt{Abst}_{\mathcal{Z}}(\mathbb{S})|\mathtt{Abst}_{\mathcal{Z}}(\mathcal{N})|\ldots$
- $\mathtt{B}^A_{\mathcal{Z}}(\mathcal{E}) = \mathtt{Adj}_{\mathcal{Z}}(P)|\ldots$
- $\mathtt{B}^P_{\mathcal{Z}}(\mathcal{E}) = \forall_{\mathcal{Z}}(P)|\ldots$

Examples

$(\mathcal{E} \equiv T)$ The term $\lambda_{x \in \mathbb{R}}(x^2)$ denotes the squaring function on the reals.

$(\mathcal{E} \equiv \mathbb{S})$ The term $\lambda_{n \in \mathbb{N}} \mathtt{Set}_{k \in \mathbb{N}}(k \leq n)$ sends a natural number n to the set $\{0, 1, \ldots, n\}$.

- The term $\iota_{n \in \mathbb{N}}(2 < n < \pi)$ describes natural number 3.
- The set $\iota_{U\colon \mathsf{SET}}\ (3 \in U \wedge |U| = 1)$ describes the singleton set $\{3\}$.

The Noun-binder of WTT

We allow *noun comprehension*, i.e. the construction of a noun.

The binder Noun is used for an *indefinite description*:
a such and such, such that . . . .

$\mathtt{Noun}_{\mathcal{Z}}(P)$, stands for *a noun saying of* $\mathcal{Z}$ *that* P.

Examples:

- The noun $\mathtt{Noun}_{x\in\mathbb{R}}(5 < x < 10)$ is *a real number between 5 and 10*.
- $\mathtt{Noun}_{V:\ \mathsf{SET}}\ (|V| = 2)$ is *a set with two elements*.

The Abst-binder of WTT

The Abst-binder *abstracts* from a term T, a set $\mathbb{S}$ or a noun $\mathcal{N}$ and delivers a noun.

Examples:

($\mathcal{E} \equiv T$) $\texttt{Abst}_{n\in\mathbb{N}}(n^2)$ represents *a term* n^2 *for some natural number* n, i.e. *the square of some natural number*.

($\mathcal{E} \equiv \mathbb{S}$) $\texttt{Abst}_{n\in\mathbb{N}}\texttt{Set}_{x\in\mathbb{R}}(x > n)$ represents *a set* $\{x \in \mathbb{R} | x > n\}$ *for some natural number* n, i.e. *an interval of the form* (n, ∞), with $n \in \mathbb{N}$.

($\mathcal{E} \equiv \mathcal{N}$) $\texttt{Abst}_{n\in\mathbb{N}}\texttt{Noun}_{x\in\mathbb{R}}(10n \leq x < 10n+1)$ represents *a real number in the interval* $[10n, 10n+1)$ *for some* n, i.e. *a non-negative real number which, written in decimal notation, has a zero at the position just before the decimal point*.

The `Adj`-binder of WTT

- Adjectives can be constructed with the `Adj`-binder.
- One can read $\texttt{Adj}_{\mathcal{Z}}(P)$ as: *the adjective saying of* $\mathcal{Z}$ *that* P.
- E.g.: $\texttt{Adj}_{n\in\mathbb{N}}(\exists_{k\in\mathbb{N}}(n = k^2 + 1))$ is an adjective saying of a natural number that it is a square plus 1.
- One could give this adjective a name, say *oversquare* and hence say things like *5 is oversquare* or *Let m be an oversquare number*.

Phrases of WTT

Phrases can be terms, sets, nouns or adjectives:

$$T = \mathtt{C}^T(\overrightarrow{\mathcal{P}})|\mathtt{B}^T_{\mathcal{Z}}(\mathcal{E})|\mathtt{V}^T \qquad \mathbb{S} = \mathtt{C}^S(\overrightarrow{\mathcal{P}})|\mathtt{B}^S_{\mathcal{Z}}(\mathcal{E})|\mathtt{V}^S$$
$$\mathcal{N} = \mathtt{C}^{\mathcal{N}}(\overrightarrow{\mathcal{P}})|\mathtt{B}^{\mathcal{N}}_{\mathcal{Z}}(\mathcal{E})|\mathcal{A}\mathcal{N} \qquad \mathcal{A} = \mathtt{C}^{\mathcal{A}}(\overrightarrow{\mathcal{P}})|\mathtt{B}^{\mathcal{A}}_{\mathcal{Z}}(\mathcal{E}).$$

We already gave examples of $\mathtt{C}^T(\overrightarrow{\mathcal{P}})$, $\mathtt{C}^S(\overrightarrow{\mathcal{P}})$, $\mathtt{C}^N(\overrightarrow{\mathcal{P}})$ and $\mathtt{C}^A(\overrightarrow{\mathcal{P}})$ and of $\mathtt{B}^T_{\mathcal{Z}}(\mathcal{E})$, $\mathtt{B}^S_{\mathcal{Z}}(\mathcal{E})$, $\mathtt{B}^N_{\mathcal{Z}}(\mathcal{E})$ and $\mathtt{B}^A_{\mathcal{Z}}(\mathcal{E})$.

The combination $\mathcal{A}\mathcal{N}$ gives a (new) noun which is a combination of an adjective and a noun. E.g.: isosceles triangle, *convergent series*.

Statements of WTT

Abstract syntax for the category of *statements* is: $P \ = \ \mathtt{C}^P(\vec{\mathcal{P}})|\mathtt{B}^P_{\mathcal{Z}}(\mathcal{E})|\mathtt{V}^P.$

Examples of $\mathtt{C}^P(\vec{\mathcal{P}})$ and of $\mathtt{B}^P_{\mathcal{Z}}(\mathcal{E})$ (with the $\forall$-binder for $\mathtt{B}^P$) were already given.

The abstract syntax for the set $\mathbf{T}$ of typing statements ($\mathbf{T} \subseteq P$) is:
$\mathbf{T} \ = \ \mathbb{S} : \ \textsf{SET} \ |\mathcal{S} : \ \textsf{STAT} \ |T : \mathbb{S}|T : \mathcal{N}|T : \mathcal{A}.$

Examples of these cases include: $\mathtt{Set}_{n\in\mathbb{N}}(n \leq 2) : \ \textit{SET}$, $p \wedge q : \ \textit{STAT}$, $3 \in \mathbb{N}$,[1]
AB *: an edge of* $\triangle ABC$, $\lambda_{x\in\mathbb{R}}(x^2)$ *: differentiable*.

[1] As this example shows, we often replace $t : s$ by $t \in s$, with abuse of notation.

Definitions of WTT

- The category $\mathcal{D} = \mathcal{D}^{\varphi}|\mathcal{D}^{P}$ of *definitions* introduces new constants.
- We distinguish between *phrase definitions* $\mathcal{D}^{\varphi}$ and *statement definitions* $\mathcal{D}^{P}$.
- Phrase definitions fix a constant representing a phrase.
- Statement definitions introduce a constant embedded in a statement.
- In definitions, the defined constant is separated from the phrase or statement it represents by the symbol ":=".

Phrase definitions of WTT

We take $\mathcal{D}^{\varphi} \quad = \quad \mathtt{C}^{T}(\vec{V}) := T \mid \mathtt{C}^{S}(\vec{V}) := \mathbb{S} \mid \mathtt{C}^{N}(\vec{V}) := \mathcal{N} \mid \mathtt{C}^{\mathcal{A}}(\vec{\mathcal{V}}) := \mathcal{A}$

Examples of phrase definitions are:

$(\mathtt{C} \equiv \mathtt{C}^{T})$ *the arithmetic mean* of a and $b := \iota_{z \in \mathbb{R}}(z = \frac{1}{2}(a+b))$,

$(\mathtt{C} \equiv \mathtt{C}^{\mathbb{S}})$ $\mathbb{R}^{+} := \mathtt{Set}_{x \in \mathbb{R}}(x > 0)$,

$(\mathtt{C} \equiv \mathtt{C}^{\mathcal{N}})$ *a unit* of G with respect to $\cdot := \mathtt{Noun}_{e \in G}(\forall_{a \in G}(a \cdot e = e \cdot a = a))$

$(\mathtt{C} \equiv \mathtt{C}^{\mathcal{A}})$ *prime* $:= \mathtt{Adj}_{n \in \mathbb{N}}(n > 1 \wedge \forall_{k,l \in \mathbb{N}}(n = k \cdot l \Rightarrow k = 1 \vee l = 1))$.

The variable lists in the four examples are: (a, b), $(\ )$, $(G, \cdot)$, $(\ )$. These variables must be introduced (*declared*) in a context.

For the first definition, such a context can be e.g. $a : \mathbb{R}, b : \mathbb{R}$.

For the third definition the context is: $G :$ SET , $\cdot : G \to G$.

Statement definitions of WTT

$\mathcal{D}^P \;=\; \mathtt{C}^P(\vec{V}) := P$ is the category of statement definitions *defining constant* $\mathtt{C}^P$.

For example in a context like: *Let* a *and* b *be lines*:

$(\mathtt{C} \equiv \mathtt{C}^P)$ a *is parallel to* b $:=$ $\neg\exists_{P\text{: a point}}(P \text{ lies on } a \wedge P \text{ lies on } b)$.

Contexts of WTT

A *context* Γ is a list of declarations $\mathcal{Z}$ and statements P:

$$\mathbb{\Gamma} \quad = \quad \emptyset \mid \mathbb{\Gamma}, \mathcal{Z} \mid \mathbb{\Gamma}, P.$$

A declaration in a context represents the *introduction of a variable* of a known type.

A statement in a context stands for an *assumption*.

Lines of WTT

A *line* l contains either a statement or a definition, relative to a context:

$$\mathbf{l} \;=\; \mathbb{\Gamma} \triangleright P \;\big|\; \mathbb{\Gamma} \triangleright \mathcal{D}.$$

The symbol $\triangleright$ is a separation marker between the context and the statement or definition.

Here are two examples of lines:

A statement line: $\quad x : \mathbb{N}, y : \mathbb{N}, x < y \;\triangleright\; x^2 < y^2$,

A definition line: $\quad x : \mathbb{R}, x > 0 \;\triangleright\; ln(x) := \iota_{y \in \mathbb{R}}(e^y = x)$.

Books of WTT

A *book* B is a list of lines: $\mathbf{B} \;=\; \emptyset \mid \mathbf{B} \circ \mathbf{l}.$

A simple example of a book consisting of two lines is the following:

$x : \mathbb{R}, x > 0 \;\triangleright\; ln(x) := \iota_{y \in \mathbb{R}}(e^y = x) \;\circ$

$\emptyset \;\triangleright\; ln(e^3) = 3\ .$

MathLang's Grammatical categories

They extend those of WTT with blocks and flags.

`T` terms

`S` sets

`N` nouns

`A` adjectives

`P` statements

`D` definitions

`Z` declarations

Γ contexts with flags

`L` lines

`K` blocks

`B` books

The Grammatical Categories of MathLang

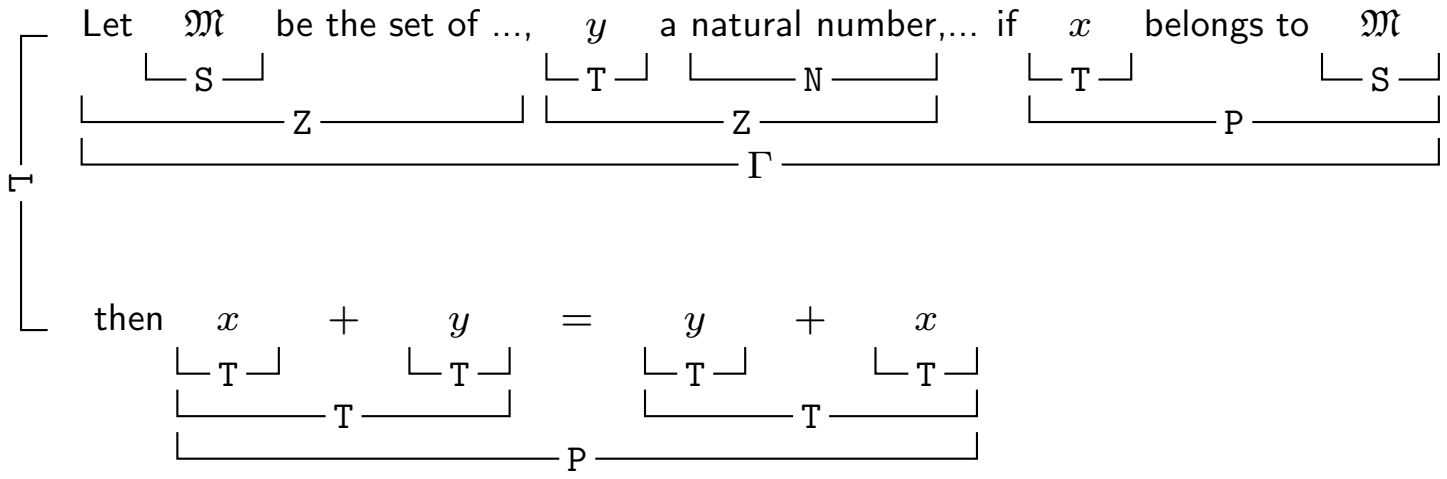


Figure 4: A mathematical line and its grammatical categories

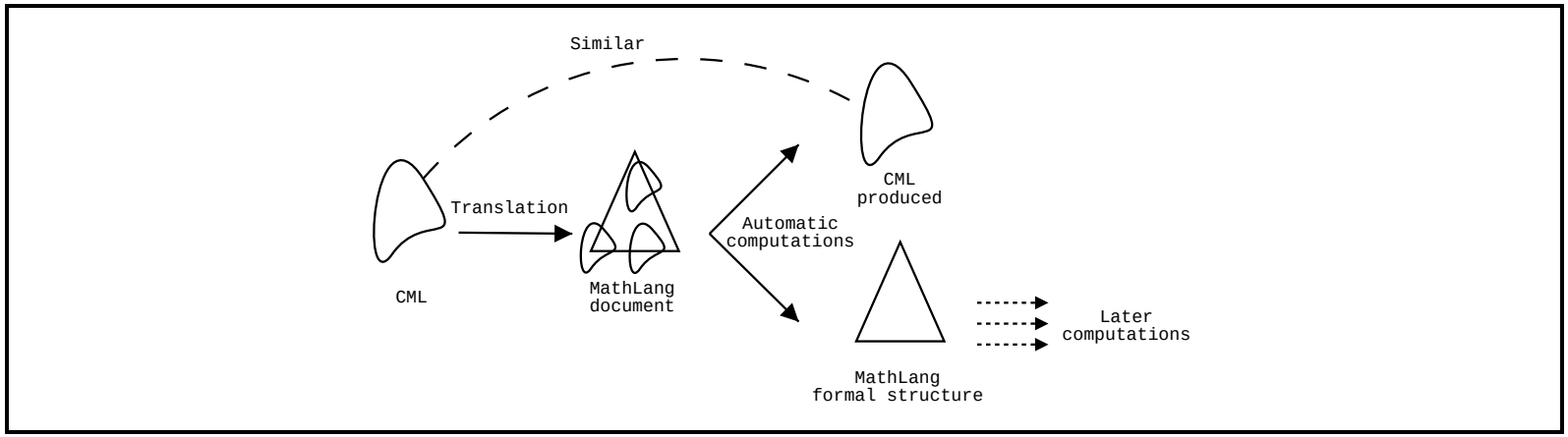


Figure 5: Translation process of MathLang

Derivation rules of WTT

(1) B is a weakly well-typed book: $\vdash\ B :: \mathbf{B}.$

(2) Γ is a weakly well-typed context relative to book B: $B\ \vdash\ \Gamma :: \mathbb{\Gamma}.$

(3) t is a weakly well-typed term, etc., relative to book B and context Γ:

$$\begin{array}{lll} B;\Gamma\ \vdash\ t :: T, & B;\Gamma\ \vdash\ s :: S, & B;\Gamma\ \vdash\ n :: N, \\ B;\Gamma\ \vdash\ a :: A, & B;\Gamma\ \vdash\ p :: P, & B;\Gamma\ \vdash\ d :: D \end{array}$$

$OK(B;\Gamma).$ stands for: $\vdash\ B :: \mathbf{B}$, *and* $B\ \vdash\ \Gamma :: \mathbb{\Gamma}$

A preface for a book B could look like:

constant name	weak type	constant name	weak type
$\mathbb{R}$	S	$\cup$	$S \times S \to S$
$\surd$	$T \to T$	$\geq$	$T \times T \to P$
$+$	$T \times T \to T$	$\wedge$	$P \times P \to P$

- $\mathbb{R}$ has no parameters and is a set.
- $\surd$ is a constant with one parameter, a term, delivering a term.
- $\geq$ is a constant with two parameters, terms, delivering a statement.
- $\texttt{prefcons}(B) = \{\mathbb{R}, \surd, +, \cup, \geq, \wedge\}$.

- $\texttt{dvar}(\emptyset) = \emptyset \qquad \texttt{dvar}(\Gamma', x : W) = \texttt{dvar}(\Gamma'), x \qquad \texttt{dvar}(\Gamma', P) = \texttt{dvar}(\Gamma')$

$$\frac{OK(B;\Gamma), \quad x \in \mathtt{V}^{\mathtt{T/S/P}}, \quad x \in \texttt{dvar}(\Gamma)}{B;\Gamma \ \vdash \ x :: T/S/P} \ (var)$$

$$\frac{B;\Gamma \ \vdash \ n :: N, \quad B;\Gamma \ \vdash \ a :: A}{B;\Gamma \ \vdash \ an :: N} \ (adj{-}noun)$$

$$\frac{}{\vdash \ \emptyset :: \mathbf{B}} \ (emp{-}book)$$

$$\frac{B;\Gamma \ \vdash \ p :: P}{\vdash \ B \ \circ \ \Gamma \triangleright p :: \mathbf{B}} \qquad \frac{B;\Gamma \ \vdash \ d :: D}{\vdash \ B \ \circ \ \Gamma \triangleright d :: \mathbf{B}} \ (book{-}ext)$$

Example in WTT

Definition 2. A *Fermat-sum* is a natural number which is the sum of two squares of natural numbers.

Lemma 3. The product of a square and a Fermat-sum is a Fermat sum.

Figure 6: *Fermat-sum* example: original text

A WTT-translation could be the following small WTT-book B of 2 lines:

$$a\ Fermat\text{-}sum\ := \mathtt{Noun}_{n\in\mathbb{N}}\exists_{k\in\mathbb{N}}\exists_{l\in\mathbb{N}}(n = k^2 + l^2)$$

$$\forall_{u:\ a\ square}\forall_{v:\ a\ Fermat\text{-}sum}(uv :\ \ a\ Fermat\text{-}sum)$$

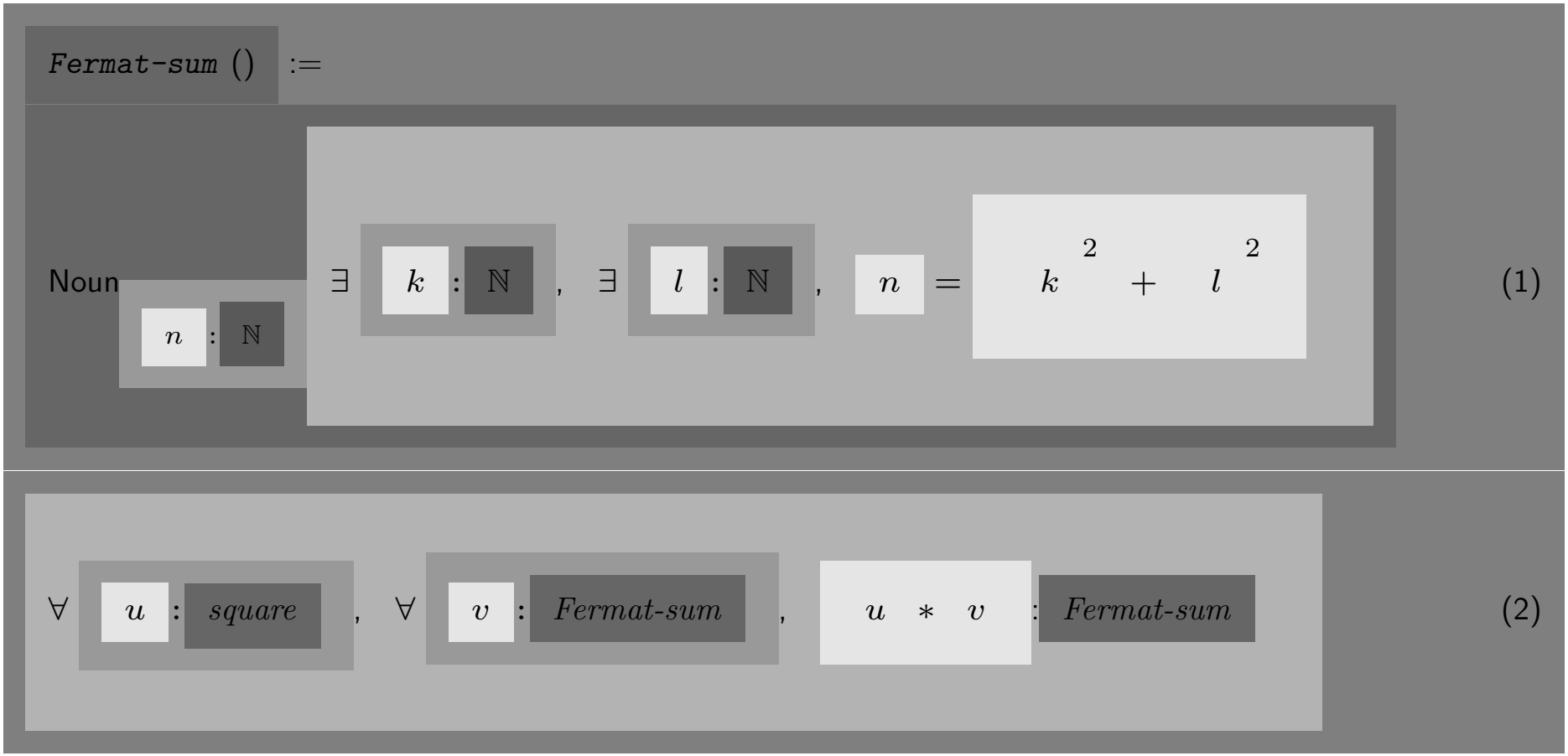


Figure 7: *Fermat-sum* example: symbolic structural view of MathLang

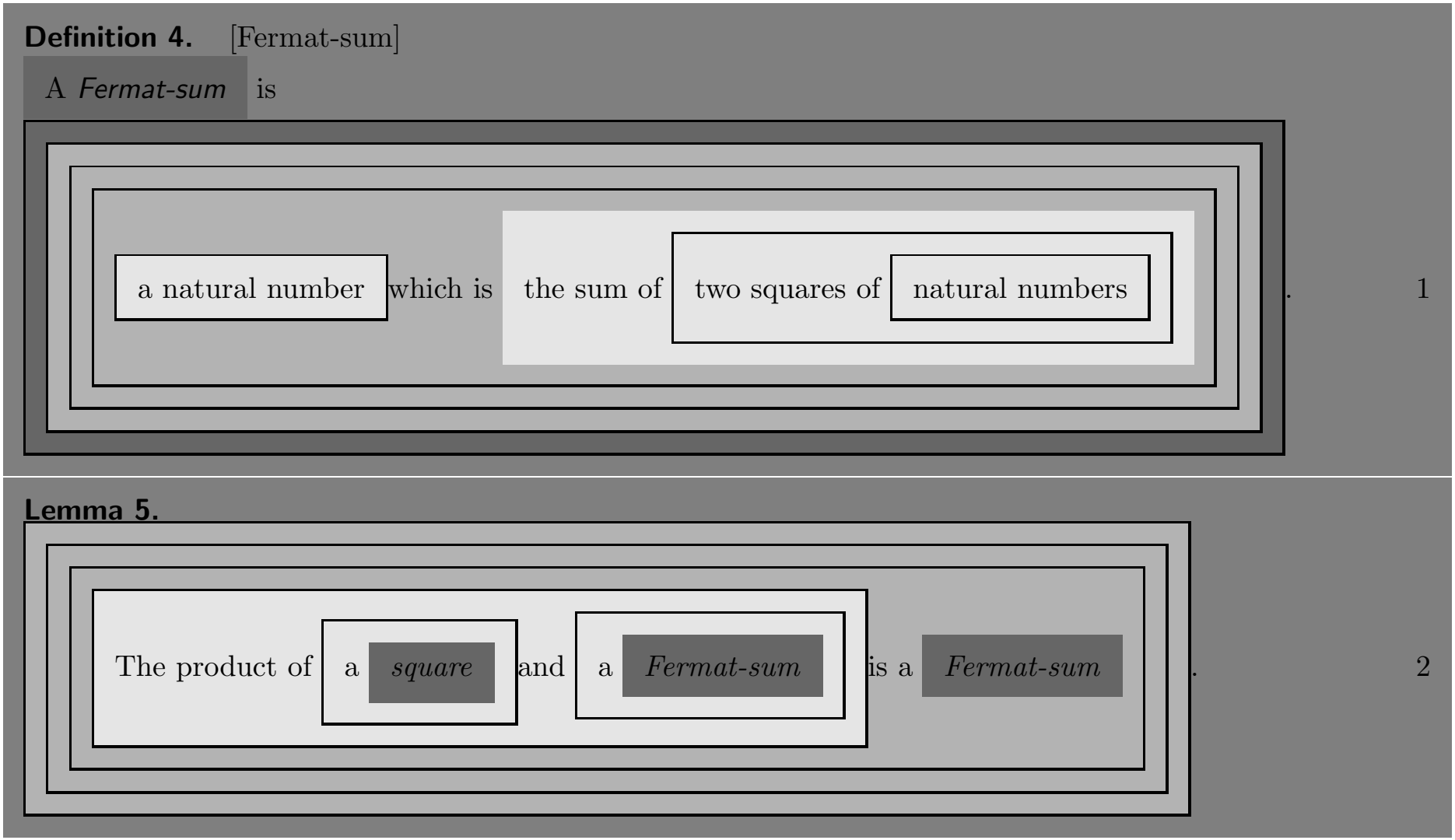


Figure 8: *Fermat-sum* example: CML view of MathLang

Another MathLang example

T Terms | S Sets | N Nouns | P Statements | Z Declarations | Γ Context

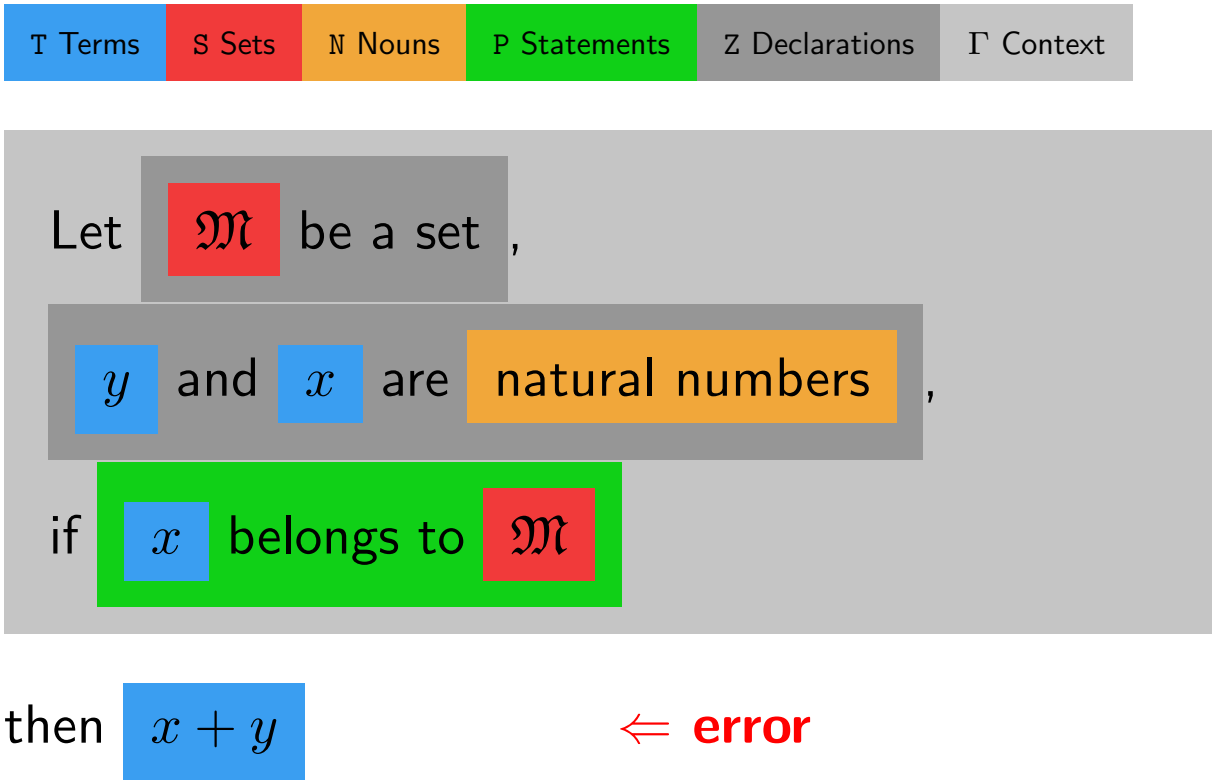
Let $\mathfrak{M}$ be a set ,

y and x are natural numbers ,

if x belongs to $\mathfrak{M}$

then $x + y = y + x$

MathLang Checking



Comparaison with other work

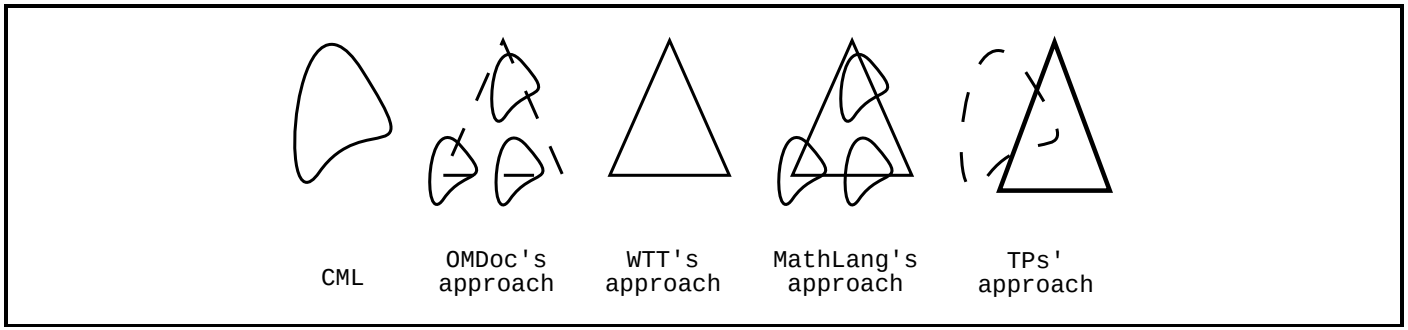


Figure 9: Approaches

Comparaison with other work

- No theorem provers provide an independent language for describing mathematical content in such a manner that the *goals 1..4* are true.

- *Existing mathematical vernaculars are not ready for immediate use*, and if accessible for a mathematical user, then with great difficulty.

- *Galina* usable as a *language of commands* for Coq. Not meant as *a first step in formalization* as MathLang is.

- The built-in version of the *mathematical vernacular* of *ΩMEGA* is meant to give the user on request a mathematics-like computer *view* of an already checked proof. It has the same drawbacks as Galina.

- The basic languages of *Mizar* and Isar preserve the mathematical content and have proven to be suited for expressing large corpora of mathematical content. Their syntax is, however, rather complicated and requires much of an ordinary user to become acquainted with it.

- In the *Theorema project* computer algebra systems, the provers are designed to imitate the proof style humans employ in their proving attempts. The proofs can be produced in human-readable style. However, this is done by *post-processing* a formal proof in natural language.

- The typed functional programming language *GF* defines languages such as fragments of natural languages, programming languages and formal calculi. GF is based on Martin-Löf's type theory.

- We do not at all assume/prefer one type theory instead of another.
- The formalisation of a language of mathematics should separate the questions:
  - *which type theory is necessary for which part of mathematics*
  - *which language should mathematics be written in*.
- Mathematicians don't usually know or work with type theories.
- Mathematicians usually *do* mathematics (manipulations, calculations, etc), but are not interested in general in reasoning *about* mathematics.

References

[1] C. Burali-Forti. Una questione sui numeri transfiniti. *Rendiconti del Circolo Matematico di Palermo*, 11:154–164, 1897. English translation in [16], pages 104–112.

[2] G. Cantor. Beiträge zur Begründung der transfiniten Mengenlehre (Erster Artikel). *Mathematische Annalen*, 46:481–512, 1895.

[3] G. Cantor. Beiträge zur Begründung der transfiniten Mengenlehre (Zweiter Artikel). *Mathematische Annalen*, 49:207–246, 1897.

[4] A.-L. Cauchy. *Cours d'Analyse de l'Ecole Royale Polytechnique*. Debure, Paris, 1821. Also as *Œuvres Complètes* (2), volume III, Gauthier-Villars, Paris, 1897.

[5] A. Church. A formulation of the simple theory of types. *The Journal of Symbolic Logic*, 5:56–68, 1940.

[6] D.T. van Daalen. *The Language Theory of Automath*. PhD thesis, Eindhoven University of Technology, 1980.

[7] R. Dedekind. *Stetigkeit und irrationale Zahlen*. Vieweg & Sohn, Braunschweig, 1872.

[8] D.R. Dowty. *Introduction to Montague Semantics*. Kluwer Academic Publishers, 1980.

[9] G. Frege. *Begriffsschrift, eine der arithmetischen nachgebildete Formelsprache des reinen Denkens*. Nebert, Halle, 1879. Also in [16], pages 1–82.

[10] G. Frege. *Grundlagen der Arithmetik, eine logisch-mathematische Untersuchung über den Begriff der Zahl*. , Breslau, 1884.

[11] G. Frege. *Grundgesetze der Arithmetik, begriffschriftlich abgeleitet*, volume I. Pohle, Jena, 1892. Reprinted 1962 (Olms, Hildesheim).

[12] G. Frege. Über Sinn und Bedeutung. *Zeitschrift für Philosophie und philosophische Kritik,* new series, 100:25–50, 1892. English translation in [**?**], pages 157–177.

[13] G. Frege. Ueber die Begriffschrift des Herrn Peano und meine eigene. *Berichte über die Verhandlungen der Königlich Sächsischen Gesellschaft der Wissenschaften zu Leipzig, Mathematisch-physikalische Klasse 48*, pages 361–378, 1896. English translation in [**?**], pages 234–248.

[14] G. Frege. Letter to Russell. English translation in [16], pages 127–128, 1902.

[15] G. Frege. *Grundgesetze der Arithmetik, begriffschriftlich abgeleitet*, volume II. Pohle, Jena, 1903. Reprinted 1962 (Olms, Hildesheim).

[Hea56] Heath. *The 13 Books of Euclid's Elements*. Dover, 1956.

[16] Heijenoort, J. v. (ed.): 1967, *From Frege to Gödel: A Source Book in Mathematical Logic, 1879–1931*. Cambridge, Massachusetts: Harvard University Press.

[17] D. Hilbert and W. Ackermann. *Grundzüge der Theoretischen Logik*. Die Grundlehren der Mathematischen Wissenschaften in Einzeldarstellungen, Band XXVII. Springer Verlag, Berlin, first edition, 1928.

[18] Kamareddine, F., L. Laan, and R. Nederpelt: 2002, 'Types in logic and mathematics before 1940'. *Bulletin of Symbolic Logic* **8**(2), 185–245.

[19] Kamareddine, F., and R. Nederpelt: 2004, A refinement of de Bruijn's formal language of mathematics. Journal of *Logic, Language and Information*. Kluwer Academic Publishers.

[20] Kamareddine, F., Maarek, M., and Wells, J.B.: 2004, MathLang: An experience driven language of mathematics, Electronic Notes in Theoretical Computer Science 93C, pages 123-145. Elsevier.

[21] F. Kamaredine, M Maarek, and J.B. Wells. Flexible encoding of mathematics on the computer. 2004.

[22] F. Kamareddine, T. Laan, and R. Nederpelt. Revisiting the notion of function. *Logic and Algebraic programming*, 54:65–107, 2003.

[Lan30] Edmund Landau. *Grundlagen der Analysis*. Chelsea, 1930.

[Lan51] Edmund Landau. *Foundations of Analysis*. Chelsea, 1951. Translation of [Lan30] by F. Steinhardt.

[23] MacLane, S.: 1972, *Categories for the Working Mathematician*. Springer.

[24] G. Peano. *Arithmetices principia, nova methodo exposita*. Bocca, Turin, 1889. English translation in [16], pages 83–97.

[25] G. Peano. *Formulaire de Mathématique*. Bocca, Turin, 1894–1908. 5 successive versions; the final edition issued as *Formulario Mathematico*.

[26] F.P. Ramsey. The foundations of mathematics. *Proceedings of the London Mathematical Society,* 2nd series, 25:338–384, 1926.

[27] J.B. Rosser. Highlights of the history of the lambda-calculus. *Annals of the History of Computing*, 6(4):337–349, 1984.

[28] B. Russell. Letter to Frege. English translation in [16], pages 124–125, 1902.

[29] B. Russell. *The Principles of Mathematics*. Allen & Unwin, London, 1903.

[30] B. Russell. Mathematical logic as based on the theory of types. *American Journal of Mathematics*, 30:222–262, 1908. Also in [16], pages 150–182.

[31] M. Schönfinkel. Über die Bausteine der mathematischen Logik. *Mathematische Annalen*, 92:305–316, 1924. Also in [16], pages 355–366.

[32] Whitehead, A. and B. Russell: 1910^1, 1927^2, *Principia Mathematica*, Vol. I, II, III. Cambridge University Press.

[33] Zermelo, E.: 1908, 'Untersuchungen über die Grundlagen der Mengenlehre'. *Math. Annalen* **65**, 261–281.