The Language of Mathematics

—when one alphabet just isn't enough—

Julian J. Schlöder
Institute for Logic, Language & Computation
University of Amsterdam

May 9th, 2014

Introduction

Everybody's Problem

- For all sets A, $\emptyset \subseteq A$.
  - The empty set is contained in every set.
  - The empty set is in every set.
- $\emptyset \notin \emptyset$.
  - The empty set is not an element in every set.
  - The empty set is **not** in every set.

The Language of Mathematics?

- What is Mathematics?
- Slightly tautological: Mathematics is what Mathematicians do.
- The Language of Mathematics is the language Mathematicians use when doing Mathematics.

The Language of Mathematics?

- What is Mathematics?
- Slightly tautological: Mathematics is what Mathematicians do.
- The Language of Mathematics is the language Mathematicians use when doing Mathematics.

There are issues with this...

> $a||b \quad \forall a \wedge b \in X.$

M. Cramer, *Proof-checking mathematical texts in controlled natural language*, PhD thesis, 2013.

M. Ganesalingam, *The Language of Mathematics*, Springer, 2013.

The Language of Mathematics

And this language is:

- Highly context-dependent, depending on the addressee (layman, student, colleague. . .).
- In essence the attempt to convince an imagined reader that a formal proof of a given proposition exists (resp. that the proposition is true).

The Language of Mathematics

And this language is:

- Highly context-dependent, depending on the addressee (layman, student, colleague. . .).
- In essence the attempt to convince an imagined reader that a formal proof of a given proposition exists (resp. that the proposition is true).
- There is a weird dilemma; with the axioms, definitions and the propositions all the information is there, but one could also write down the complete formal proof.
- So the writer provides enough information for an imagined reader to come to the conclusion that the proposition is provable on his own. In particular, the writer tries to anticipate the difficulties the reader might have.

...Compared to Natural Language

The most fundamental difference mathematical language exhibits compared with natural language is the treatment of information content:

- In natural language, statements add information, i.e., restrict context.
- In mathematical language, statements must be inferable from the already available information.

. . . Compared to Natural Language

The most fundamental difference mathematical language exhibits compared with natural language is the treatment of information content:

- In natural language, statements add information, i.e., restrict context.
- In mathematical language, statements must be inferable from the already available information.
- Thus the crucial property of a mathematical statement is its attentive content.
- Every step in a proof does not add new information, but it draws the attention of the reader to the steps in a imagined formal proof the writer deems crucial.

Example

Theorem

There are infinitely many prime numbers.

Proof.

Let n be any natural number. Consider $k = n! + 1$. Let p be a prime that divides k. If $p \leq n$, then p divides $n!$, so p does not divide k. Contradiction. □

Example

Theorem

There are infinitely many prime numbers, i.e., for each natural number n there is a prime $p > n$.

Proof.

Let n be any natural number. Consider $k = n! + 1$. Let p be a prime that divides k. If $p \leq n$, then p divides $n!$, so p does not divide k. Contradiction. □

Example

Theorem

There are infinitely many prime numbers, i.e., for each natural number n there is a prime $p > n$.

Proof.

Let n be any natural number. Consider $k = n! + 1$. Let p be a prime that divides k. If $p \leq n$, then p divides $n!$, so p does not divide k, because otherwise p would divide 1. Contradiction. □

Example

Theorem

There are infinitely many prime numbers, i.e., for each natural number n there is a prime $p > n$.

Proof.

Let n be any natural number. Consider $k = n! + 1$. Let p be a prime that divides k. If $p \leq n$, then p divides $n!$, so p does not divide k, because otherwise p would divide 1, and primes are larger than 1. Contradiction. □

Example

Theorem

There are infinitely many prime numbers, i.e., for each natural number n there is a prime $p > n$.

Proof.

Let n be any natural number. Consider $k = n! + 1$. Let p be a prime that divides k, by the Fundamental Theorem of Arithmetic. If $p \leq n$, then p divides $n!$, so p does not divide k, because otherwise p would divide 1, and primes are larger than 1. Contradiction. □

Example

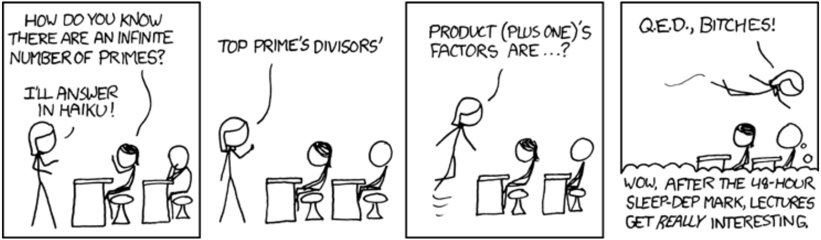
Theorem

There are infinitely many prime numbers, i.e., for each natural number n there is a prime $p > n$.

Proof.

Let n be any natural number. Consider $k = n! + 1$. Then $k \geq 2$. Let p be a prime that divides k, by the Fundamental Theorem of Arithmetic. If $p \leq n$, then p divides $n!$, so p does not divide k, because otherwise p would divide 1, and primes are larger than 1. Contradiction. □

Example



`http://xkcd.com/622/`

Overview

Notational Types

- infix, $n + m$

Notational Types

- infix, $n + m$
- suffix, $n!$

Notational Types

- infix, $n + m$
- suffix, $n!$
- prefix, $\sin x$

Notational Types

- infix, $n + m$
- suffix, $n!$
- prefix, $\sin x$
- n-ary classical, $f(x)$ $<(a, b)$ $T(a, b, c)$

Notational Types

- infix, $n + m$
- suffix, $n!$
- prefix, $\sin x$
- n-ary classical, $f(x)$ $< (a, b)$ $T(a, b, c)$
- circumfix, $[a, b]$ $|A|$ $||v||$

Notational Types

- infix, $n + m$
- suffix, $n!$
- prefix, $\sin x$
- n-ary classical, $f(x)$ $<(a, b)$ $T(a, b, c)$
- circumfix, $[a, b]$ $|A|$ $||v||$
- positional-symbol, $\overline{A}$ $\underset{\sim}{f}$ id_X ${}_A\oplus$ π^* ${}^n\sqrt{\ }$ $\frac{a}{b}$

Notational Types

- infix, $n + m$
- suffix, $n!$
- prefix, $\sin x$
- n-ary classical, $f(x)$ $< (a, b)$ $T(a, b, c)$
- circumfix, $[a, b]$ $|A|$ $||v||$
- positional-symbol, $\overline{A}$ $\underset{\sim}{f}$ id_X ${}_A\oplus$ π^* ${}^n\sqrt{}$ $\frac{a}{b}$
- positional-implicit, ab a^b ${}^\kappa\lambda$ f_k $T_\alpha{}^\beta{}_\gamma{}^\delta$

Notational Types

- infix, $n + m$
- suffix, $n!$
- prefix, $\sin x$
- n-ary classical, $f(x)$ $<(a, b)$ $T(a, b, c)$
- circumfix, $[a, b]$ $|A|$ $||v||$
- positional-symbol, $\overline{A}$ $\underset{\sim}{f}$ id_X ${}_A\oplus$ π^* ${}^n\sqrt{\ }$ $\frac{a}{b}$
- positional-implicit, ab a^b ${}^\kappa\lambda$ f_k $T_\alpha{}^\beta{}_\gamma{}^\delta$
- mixed, $\kappa \to \lambda_\nu^\mu$ $[E : F]$ $\binom{n}{k}$ $\int_y^z f\ dx$ $\log_a b$

Notational Types

- infix, $n + m$
- suffix, $n!$
- prefix, $\sin x$
- n-ary classical, $f(x)$ $<(a, b)$ $T(a, b, c)$
- circumfix, $[a, b]$ $|A|$ $||v||$
- positional-symbol, $\overline{A}$ $\underset{\sim}{f}$ id_X ${}_A\oplus$ π^* ${}^n\sqrt{\ }$ $\frac{a}{b}$
- positional-implicit, ab a^b ${}^\kappa\lambda$ f_k $T_\alpha{}^\beta{}_\gamma{}^\delta$
- mixed, $\kappa \to \lambda^\mu_\nu$ $[E : F]$ $\binom{n}{k}$ $\int_y^z f\ dx$ $\log_a b$
  - complex types of simple notations, e.g., $\log$ has type `[implicit-right-below,prefix]`.

Structural Ambiguity

> Define $x - y$ as $x + (-y)$

$-$ is used both as a 2-ary and a unary function symbol.

Structural Ambiguity

Define $x - y$ as $x + (-y)$

$-$ is used both as a 2-ary and a unary function symbol.

ρ generates the splitting field of some polynomial over F_0.

- generation over F_0
- the splitting field over F_0
- a polynomial over F_0

Structural Ambiguity

> Define $x - y$ as $x + (-y)$

$-$ is used both as a 2-ary and a unary function symbol.

> ρ generates the splitting field of some polynomial over F_0.

- generation over F_0
- the splitting field over F_0
- a polynomial over F_0

What does this formula mean:

$$a(b + c)$$

Structural Ambiguity

> Define $x - y$ as $x + (-y)$

$-$ is used both as a 2-ary and a unary function symbol.

> ρ generates the splitting field of some polynomial over F_0.

- generation over F_0
- the splitting field over F_0
- a polynomial over F_0

What does this formula mean:

$$a(b + c)$$

And this?

$$f(x + y)$$

Lexical Ambiguities

- nice {name, extender}.
- proper {subset, map, morphism, forcing}.
- almost all numbers are not rational

Lexical Ambiguities

- nice {name, extender}.
- proper {subset, map, morphism, forcing}.
- almost all numbers are not rational

Though there are exceptions...

Definition

A *mouse* is an iterable premouse.

Lexical Ambiguities

- nice {name, extender}.
- proper {subset, map, morphism, forcing}.
- almost all numbers are not rational

Though there are exceptions...

Definition

A *mouse* is an iterable premouse.

This is not restricted to words, but also happens with symbols:

- π can be the number, or the prime counting function;
- $\aleph$ can be the function, or the size of the continuum.

This can be disambiguated with a typed lexicon.

Formal and Informal Language

It is widely believed that one can state any mathematical result purely in first-order logic. For example the Power Set Axiom:

$$\forall x \exists y \forall z : z \in y \leftrightarrow (\forall a : a \in z \rightarrow a \in x).$$

Formal and Informal Language

It is widely believed that one can state any mathematical result purely in first-order logic. For example the Power Set Axiom:

> $\forall x \exists y \forall z : z \in y \leftrightarrow (\forall a : a \in z \to a \in x)$.

But we can state the Power Set Axiom semi-formally:

> Say that a is a subset of b iff $\forall z : z \in a \to z \in b$.
> Then define the powerset of a, $\mathcal{P}(a)$, to be the set of all subsets of a.

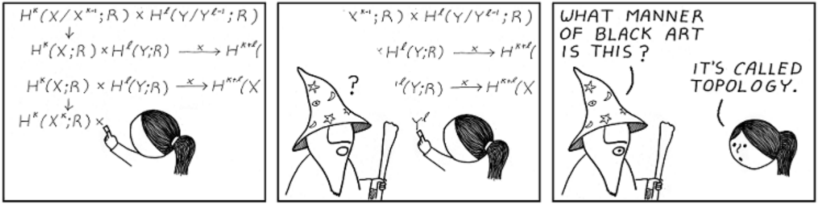
> $\forall x \exists y : y = \mathcal{P}(x)$.
> For each set there is its powerset.

This formulation required the expansion of the lexicon through informal language use.

Implicit Definition

> For each line L there is a point p_L such that p lies in L.

This defines a function from the space of lines to the space of points.



http://abstrusegoose.com/253

Plurals

A typical problem in dealing with plurals is that one might talk about a collective property or a collection of things with a distributive property.

- 12 and 25 are coprime. $\rightsquigarrow$ collective property.
- 2 and 3 are prime. $\rightsquigarrow$ distributive property.

Plurals

A typical problem in dealing with plurals is that one might talk about a collective property or a collection of things with a distributive property.

- 12 and 25 are coprime. $\rightsquigarrow$ collective property.
- 2 and 3 are prime. $\rightsquigarrow$ distributive property.
- A, B and C are (pairwise) disjoint. $\rightsquigarrow$ distributively, all pairs have a collective property.

Plurals

A typical problem in dealing with plurals is that one might talk about a collective property or a collection of things with a distributive property.

- 12 and 25 are coprime. $\rightsquigarrow$ collective property.
- 2 and 3 are prime. $\rightsquigarrow$ distributive property.
- A, B and C are (pairwise) disjoint. $\rightsquigarrow$ distributively, all pairs have a collective property.
- φ and ψ are inconsistent. $\rightsquigarrow$ ambiguity.
  - Is $\{\varphi, \psi\}$ inconsistent, or is φ inconsistent and ψ inconsistent?
- φ and ψ imply χ. $\rightsquigarrow$ ambiguity.
  - $\{\varphi, \psi\} \vdash \chi$ or $\{\varphi\} \vdash \chi$ and $\{\varphi\} \vdash \chi$?

Determiners

Intuitively the selects an unique object, and a selects a possibly not unique object.

- The empty set.
- Let V be a vector space.

Determiners

Intuitively the selects an unique object, and a selects a possibly not unique object.

- The empty set.
- Let V be a vector space.

a can also work as universal quantification:

> Then V is a vector space. A vector space has a base,
> so let $b_1, \cdots, b_n$ be a base of V.

Determiners

Intuitively the selects an unique object, and a selects a possibly not unique object.

- The empty set.
- Let V be a vector space.

a can also work as universal quantification:

> Then V is a vector space. A vector space has a base,
> so let $b_1, \cdots, b_n$ be a base of V.

And the can also be an anaphora:

> Suppose there are such a field and vector space.
> Let B be a base of the vector space.

Quantifiers

In Mathematics, there should be no quantifier scope ambiguity.

- There is a δ for each ε...
  - *for each* outscopes *there is* here.

Quantifiers

In Mathematics, there should be no quantifier scope ambiguity.

- There is a δ for each ε...
  - *for each* outscopes *there is* here.

Furthermore, *some* and *every/all* cannot be treated symmetrically, as *some* has existential import.

> Then $V = U \cap H$ for some $U \in \mathcal{I}$. Then $U \cap H = i^{-1}(U)$.
> Then $V = U \cap H$ for all $U \in \mathcal{I}$. # Then $U \cap H = i^{-1}(U)$.

Quantifiers

In Mathematics, there should be no quantifier scope ambiguity.

- There is a δ for each ε...
  - *for each* outscopes *there is* here.

Furthermore, *some* and *every/all* cannot be treated symmetrically, as *some* has existential import.

> Then $V = U \cap H$ for some $U \in \mathcal{I}$. Then $U \cap H = i^{-1}(U)$.
> Then $V = U \cap H$ for all $U \in \mathcal{I}$. # Then $U \cap H = i^{-1}(U)$.

And then, some people are just reckless:

> $\neg A(x) \forall x \in X \Leftrightarrow \exists x \in X : \neg A(x)$.

Meta-Language

Frequently, mathematicians quantify over sentences in the language:

> One of the following statements is false.

> Exactly one of these cases holds.

> Thus we are in Case 2.

Meta-Language

Frequently, mathematicians quantify over sentences in the language:

> One of the following statements is false.

> Exactly one of these cases holds.

> Thus we are in Case 2.

Sometimes, properties of variables are restricted and/or may be lifted:

> Suppose that $n > 0$. Then ...
> Now suppose that $n \leq 0$. Then ...

Presuppositions (1)

> Let n be the smallest element of A.

This presupposes that A indeed does have a smallest element. Contrary to conversational language, presuppositions in mathematics do not add information, but are assumed to be inferred from the context. If the presupposition can't be met, we have a logical mistake.

Presuppositions (1)

> Let n be the smallest element of A.

This presupposes that A indeed does have a smallest element. Contrary to conversational language, presuppositions in mathematics do not add information, but are assumed to be inferred from the context. If the presupposition can't be met, we have a logical mistake.

> If A has a smallest element, let n be the smallest element of A.
> Let A be a well-founded set and let n be the smallest element of A.

Presuppositions (1)

> Let n be the smallest element of A.

This presupposes that A indeed does have a smallest element. Contrary to conversational language, presuppositions in mathematics do not add information, but are assumed to be inferred from the context. If the presupposition can't be met, we have a logical mistake.

> If A has a smallest element, let n be the smallest element of A.
> Let A be a well-founded set and let n be the smallest element of A.

> $\#$ If A is a set of reals, let n be the smallest element of A.
> If A is a set of naturals, let n be the smallest element of A.

Cramer, Kühlwein, Schröder. *Presupposition Projection and Accommodation in Math. Texts*, KONVENS, 2010.

Presuppositions (2)

If a presupposition can't be met, it can be accommodated.

> Define (a function) $\min A$ to be the smallest element of A.

This presupposes that all A in the domain of min have a smallest element. If this can not be (directly) inferred from context, we can locally accommodate the presupposition, i.e., restrict the domain of $\min$ to the sets A that have a minimal element.

Presuppositions (2)

If a presupposition can't be met, it can be accommodated.

> Define (a function) $\min A$ to be the smallest element of A.

This presupposes that all A in the domain of min have a smallest element. If this can not be (directly) inferred from context, we can locally accommodate the presupposition, i.e., restrict the domain of $\min$ to the sets A that have a minimal element.

> Divide both sides of the equation by x.

This presupposes that x is never 0. Accommodating this is the source of many mathematical errors; even among trained Mathematicians.

Cramer, Kühlwein, Schröder. *Presupposition Projection and Accommodation in Math. Texts*, KONVENS, 2010.

Disambiguation

Typing

We can enforce manual typing (Mizar does this):

> # Find f with $f(x + y) > x \cdot y$.

> Find a function f with $f(x + y) > x \cdot y$.

> Find a real f with $f(x + y) > x \cdot y$.

Typing

We can enforce manual typing (Mizar does this):

> \# Find f with $f(x + y) > x \cdot y$.

> Find a function f with $f(x + y) > x \cdot y$.

> Find a real f with $f(x + y) > x \cdot y$.

However, this is very tedious in actual applications and quite unnatural. An excerpt from a Mizar's definition of logics:

> let A be alphabet; let p,q be formula of A;
> func p '->' q -> formula of A equals [...];

> let A1, A2 be alphabet, p be formula of A1, q be formula of A2;
> \# consider r = p '->' q;

In these frameworks one necessarily needs Typecasts.

Context

Alternatively, one can decide to read potentially ambigous statements with the excpectation that they can be disambiguated from context.

> Let $f \subseteq \mathbb{R}^2$ be a functional relation
> such that for all x, y, $f(x + y) > x \cdot y$.

An automated theorem prover can infer that f is used as a function.

Context

Alternatively, one can decide to read potentially ambigous statements with the excpectation that they can be disambiguated from context.

> Let $f \subseteq \mathbb{R}^2$ be a functional relation
> such that for all x, y, $f(x + y) > x \cdot y$.

An automated theorem prover can infer that f is used as a function.

So we make the general assumption that mathematical text in fact is non-ambigous and see if we can meet this assumption.

This strategy was implemented in the Naproche Project, but was deemed too computationally intensive for practical application.

J. Schlöder, *Internship Report*, Naproche Project, 2010.

M. Cramer, *Proof-checking mathematical texts in controlled natural language*, PhD thesis, 2013.

Consistency

Reversing this, one can also infer that one reading is inconsistent.

> Define f such that for all $x, y \in \mathbb{R}$ $f(x + y) > x \cdot y$.

In this case one can infer that $f(x, y)$ is not used multiplicatively:
For there is no number f s.t. for all x and y, $f \cdot (x + y) > x \cdot y$.

Consistency

Reversing this, one can also infer that one reading is inconsistent.

> Define f such that for all $x, y \in \mathbb{R}$ $f(x + y) > x \cdot y$.

In this case one can infer that $f(x, y)$ is not used multiplicatively: For there is no number f s.t. for all x and y, $f \cdot (x + y) > x \cdot y$.

Sometimes this is our only hope—when typing does not help us.

Recall the subset-element problem. Both *(contained) in* and *(element) in* are relations between sets.

But we can observe that in both cases one of the two possible readings is inconsistent.

So if confronted with two ambigous readings of a sentence, we can check if one of them is inconsistent and discard this reading.

Mizar

Mizar

- Has non-ambigous syntax based on Pascal.
- Is statically typed, to avoid ambiguities.
- Requires manual premise selection.
- Proof-checking is local to each statement.
- Supports schemata to give second order logic capabilities.
- Its logic is axiomatized as Tarski-Grothendieck Set Theory.
- Every step in a proof must be explicated.
- Is currently the largest collection of formalized knowledge; most important results according to the MML:
  - Fundamental theorems of algebra and arithmetic (Milewski; Kornilowicz, Rudnicki).
  - Jordan Curve theorem (Kornilowicz et al.).
  - Levy Reflection theorem (Bancerek).
  - Gödel Completeness theorem (Koepke, Braselmann, S.).

Mizar Project, www.mizar.org

Mizar

```
reserve n,p for Nat;
theorem Euclid: ex p st p is prime & p > n proof
set k = n! + 1;
n! > 0 by NEWTON:23;
then n! >= 0 + 1 by NAT_1:38;
then k >= 1 + 1 by REAL_1:55;
then consider p such that
A1: p is prime & p divides k by INT_2:48;
A2: p <> 0 & p > 1 by A1,INT_2:def 5;
take p;
thus p is prime by A1;
assume p <= n;
then p divides n! by A2,NAT_LAT:16;
then p divides 1 by A1,NAT_1:57;
hence contradiction by A2,NAT_1:54; end;
```

F. Wenzel & F. Wiedijk, *A comparison of Mizar and Isar*, Journal of Automated Reasoning, 2002.

Naproche

Naproche

- Implements a controlled natural language inspired by the language in mathematical textbooks.
- Supports implicit function definition.
- Also uses typing, but has, e.g., also quantifier scope disambiguation and lexical disambiguation.
- Computes presuppositions and possibly accomodates them.
- Selects premises automatically.
- Proof-checking is contextual (proofs are analyzed via DRT).
- The fundamental logic is a weak fragment of second order logic with identity.
- Is currently unable to sustain large knowledge bases, but:
  - *Grundlagen der Analysis* by E. Landau (Cramer)
  - Fragments of Set Theory (Cramer, Kühlwein, S.).
  - Number Theory by M. Carl (ongoing project).

Naproche Project, www.naproche.net

Naproche

Lemma 1: For all m, n, $m + n - m = n$.
Lemma 2: No prime p divides 1.
Lemma 3: If n divides k and m, then n divides $k - m$.
Lemma 4: For every n, for every k, k divides $n!$ or $k > n$.
Lemma 5: For every n, $n = 1$ or some prime p divides n.

Theorem: For every n, there is a prime p such that $p > n$.
Proof:
Fix n. Then $n! + 1$ is a natural number and $n! + 1 \neq 1$.
So there is a prime p such that p divides $n! + 1$.
Assume for a contradiction that it is not the case that $p > n$.
Hence p divides $n!$.
Then p divides 1. Contradiction.
Qed.

Formalization by Marcos Cramer, University of Luxembourg.

Fin

Thank you!