

An abstract syntax for a formal language of mathematics

Rob Nederpelt¹ and Fairouz Kamareddine²

¹ Mathematics and Computing Science, Eindhoven Univ. of Technology, P.O.Box 513, 5600 MB Eindhoven, the Netherlands, Tel: +31 40 247 2718, Fax: +31 40 247 5361, r.p.nederpelt@tue.nl

² Department of Computing and Electrical Engineering, Heriot-Watt University, Edinburgh EH14 4AS, Scotland, Tel: +44 131 451 3868, Fax: +44 131 451 8179, fairouz@cee.hw.ac.uk

Abstract. This paper provides an abstract syntax for a formal language of mathematics. We call our language *Weak Type Theory* (abbreviated *WTT*). WTT will be as faithful as possible to the mathematician's language yet will be formal and will not allow ambiguities. WTT can be used as an intermediary between the natural language of the mathematician and the formal language of the logician. As far as we know, this is the first extensive formalization of an abstract syntax of a formal language of mathematics.

1 Abstract syntax for WTT

In this section, we develop a syntax for Weak Type Theory, based on linguistic categories. These categories include nouns and adjectives, which usually are not incorporated in formalizations of mathematics. Constants are treated as 'first-class citizens', as are binders like \sum and \bigcup . With a view to the categories 'nouns' and 'adjectives', we introduce a number of extra binders, to facilitate linguistic constructions. On the sentence level, definitions play a prominent role in WTT. They come in various forms and formats, reflecting the habitual ways in which mathematicians use the definition-mechanism. Like in Type Theory, contexts are an important ingredient of WTT, giving the necessary immediate background for statements and definitions by listing their free variables together with their types. The notion of 'line' is used to express a statement or a definition together with its context. The final entity in the WTT-syntax is the 'book', being a sequence of lines. The book is the formal counterpart of a 'mathematical text' (e.g. a theory).

1.1 Linguistic categories

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

- On the *atomic* level: *variables*, *constants* and *binders*,
- On the *phrase*³ level: *terms*, *sets*, *nouns*⁴ and *adjectives*,
- On the *sentence* level: *statements* and *definitions*,
- On the *discourse*⁵ level: *contexts*, *lines* and *books*.

There is a hierarchy between the four levels: atoms are used to construct phrases; both atoms and phrases are part of sentences; and discourses are built from sentences.

We describe each of these categories below, first variables (see Section 1.3) and constants (Section 1.4), then binders (Section 1.5), phrases (Section 1.6), statements (Section 1.7), definitions (Section 1.8) and finally contexts, lines and books (Sections 1.9, 1.10 and 1.11).

The syntax given in the following sections, establishes *well-formedness* conditions for these categories. We assume that the sets of variables, constants and binders are given beforehand, hence fixed, and that they are mutually disjoint. The syntax decides about the well-formedness of phrases, statements, definitions, and also of contexts, lines and books. Hence, a phrase, . . . , constructed with the syntax, can be considered a *well-formed* phrase, For convenience, however, we leave this implicit in the following sections, hence we suppress the word 'well-formed' in the syntactic description of all categories mentioned.

³ In the Concise Oxford Dictionary, a phrase is 'a group of words forming a conceptual unit, but not a sentence'.

⁴ We use 'noun' for what is known in linguistics as *indefinite noun phrase*, and 'adjective' for *adjective phrase*.

⁵ A discourse is 'a connected series of utterances'.

1.2 The abstract syntax

We use abstract syntax for the description of the various categories. For example, in Section 1.11 we describe the collection of all books, \mathbf{B} , in abstract syntax as: $\mathbf{B} = \emptyset \mid \mathbf{B} \circ \mathbf{l}$.

Hence, a book is either 'the empty book' or a book \mathbf{B} followed by a line \mathbf{l} , with an open circle as separation marker. (By convention, $\emptyset \circ \mathbf{l}$ is written as \mathbf{l} .) Otherwise said, a book is a finite (length ≥ 0) sequence of lines.

In this syntax we make use of binders \mathbf{B} (think of e.g. \sum or \forall), in the following abstract format: $\mathbf{B}_{\mathcal{Z}}(\mathcal{E})$, where the *subscript* \mathcal{Z} is a *declaration* introducing a (bound) variable and its type, e.g. $x \in \mathbb{N}$. (See Section 1.7 for a formal definition of a declaration.) The symbol \mathcal{E} stands for an *expression*, to be specified in Section 1.5.

Examples of formulas with binding symbols in the above format are: $\sum_{x \in \{0,1,\dots,10\}}(x^2)$ and $\forall_{x \in \mathbb{N}}(x \geq 0)$.

The binding symbol for set comprehension, $\{\dots \mid \dots\}$, fits in this format after a slight modification. E.g., write $\{x \in \mathbb{R} \mid x > 5\}$ as $\text{Set}_{x \in \mathbb{R}}(x > 5)$. *For uniformity, our standard for set notation will be the latter one.*

We use the following standard *metasymbols* for the categories mentioned above:

level	category	symbol	representative	level	category	symbol	representative
atomic level	<i>variables</i>	\mathbf{V}	x	sentence level	<i>statements</i>	\mathcal{S}	S
	<i>constants</i>	\mathbf{C}	c		<i>definitions</i>	\mathcal{D}	D
	<i>binders</i>	\mathbf{B}	b				
phrase level	<i>terms</i>	\mathfrak{t}	t	discourse level	<i>contexts</i>	$\mathbf{\Gamma}$	Γ
	<i>sets</i>	σ	s		<i>lines</i>	\mathbf{l}	l
	<i>nouns</i>	ν	n		<i>books</i>	\mathbf{B}	B
	<i>adjectives</i>	α	a				

Other metasymbols used in this paper are:

	category	symbol	representative
	<i>expressions</i>	\mathcal{E}	E
	<i>parameters</i>	\mathcal{P}	P
	<i>typings</i>	\mathcal{T}	T
	<i>declarations</i>	\mathcal{Z}	Z

Notation 11 *In the abstract syntax used in this paper, upper indices and lower indices play different roles. Upper indices are part of the symbol, but lower indices belong to the abstract syntax. For example, with $\mathbf{B}_{\mathcal{Z}}^{\mathfrak{t}}(\mathcal{E})$, we mean all constructs composed of a binder in the set $\mathbf{B}^{\mathfrak{t}}$ (e.g. \lim '), subscripted with a declaration from \mathcal{Z} (e.g. $n \in \mathbb{N}$ ') and followed by an expression in \mathcal{E} (e.g. $\frac{1}{n}$) between parentheses. The superscript \mathfrak{t} attached to \mathbf{B} says that the binders in $\mathbf{B}^{\mathfrak{t}}$ are term-forming. Hence, $\lim_{n \in \mathbb{N}}(\frac{1}{n})$ ' is a term belonging to $\mathbf{B}_{\mathcal{Z}}^{\mathfrak{t}}(\mathcal{E})$.*

1.3 Variables

The set \mathbf{V} of *variables* is given beforehand, infinite, and divided into two disjoint subsets (In abstract syntax $\mathbf{V} = \mathbf{V}^{\mathfrak{t}} \mid \mathbf{V}^{\sigma}$): $(\mathbf{V}^{\mathfrak{t}})$ Variables ranging over *terms*, and (\mathbf{V}^{σ}) Variables ranging over *sets*.

1.4 Constants

Constants play an important role in mathematical language. They are either 'primitive'⁶ or they act as an abbreviation. In the latter case a constant is introduced in the left hand side of a *definition*, being a special kind of sentence (see Section 1.8). Both primitive and defined constants can be *used* after having been introduced. 'Doing' mathematics *without* constants (hence without definitions) is practically unfeasible.

The set of *constants* \mathbf{C} in WTT is given beforehand, is infinite and is disjoint from the set of variables. \mathbf{C} is divided in the following six disjoint subsets (in abstract syntax $\mathbf{C} = \mathbf{C}^{\mathfrak{t}} \mid \mathbf{C}^{\sigma} \mid \mathbf{C}^{\nu} \mid \mathbf{C}^{\alpha} \mid \mathbf{C}^{\rho} \mid \mathbf{C}^{\mathfrak{l}}$):

$(\mathbf{C}^{\mathfrak{t}})$ Constants for *terms*, (\mathbf{C}^{σ}) Constants for *sets*, (\mathbf{C}^{ν}) Constants for *nouns*,
 (\mathbf{C}^{α}) Constants for *adjectives*, (\mathbf{C}^{ρ}) *Relational* constants, $(\mathbf{C}^{\mathfrak{l}})$ *Logical* constants.

⁶ *Primitive* constants are introduced axiomatically, they are not defined in terms of other notions. E.g., the primitive set \mathbb{N} of the natural numbers, the primitive function s ('successor') from \mathbb{N} to \mathbb{N} or the primitive element 0 in \mathbb{N} .

A constant is always followed by a *parameter list*. We denote this as $\mathcal{C}(\vec{\mathcal{P}})$. This list has for each constant a fixed length ≥ 0 , the *arity* of the constant. Parameters \mathcal{P} are either terms, sets or statements: $\mathcal{P} = \mathfrak{t}|\sigma|\mathcal{S}$
(If the parameter list is empty we write c instead of $c(\)$.)

Examples of constants We give examples for each of the six kinds of constants, with parameter lists:

- ($\mathcal{C}^{\mathfrak{t}}$) Constants for *terms*: π , the centre of C , $3 + 6$, the arithmetic mean of 3 and 6, $d(x, y)$, ∇f .
The constants in this example are the terms π , 'the centre', $+$, 'the arithmetic mean', d and ∇ .
The parameter lists are: $(\)$, (C) , $(3, 6)$, $(3, 6)$, (x, y) and (f) , respectively.⁷
- (\mathcal{C}^{σ}) For *sets*: \mathbb{N} , A^c , $V \rightarrow W$, $A \cup B$.
The constants are: \mathbb{N} , c , \rightarrow , \cup . The parameter lists are: $(\)$, (A) , (V, W) , (A, B) .
- (\mathcal{C}^{ν}) For *nouns*: a triangle, an eigenvalue of A , a reflection of V with respect to l , an edge of $\triangle ABC$.
The constants are: 'a triangle', 'an eigenvalue', 'a reflection', 'an edge'.
The parameter lists are: $(\)$, (A) , (V, l) , $(\triangle ABC)$.
- (\mathcal{C}^{α}) For *adjectives*: prime, surjective, Abelian, continuous on $[a, b]$.
The constants are: prime, surjective, Abelian, continuous. The parameter lists are: $(\)$, $(\)$, $(\)$, $([a, b])$.
- (\mathcal{C}^{ρ}) For *relational statements*: ' P lies between Q and R ', ' $5 \geq 3$ '.
The constants are: 'lies between', \geq . The parameter lists are: (P, Q, R) , $(5, 3)$.
- (\mathcal{C}^{ι}) For *logical statements*: $p \wedge q$, $\neg \forall_{x \in \mathbb{N}}(x > 0)$.
The constants are: \wedge , \neg . The parameter lists are: (p, q) , $(\forall_{x \in \mathbb{N}}(x > 0))$.

Note that the parameters in parameter lists are usually either *terms* or *sets*. Only in the case of logical statements the parameters are *statements*.

Special constants A special constant in the category \mathcal{C}^{σ} is \uparrow and a corresponding one in the category \mathcal{C}^{ν} is \downarrow . The unary constant \uparrow 'lifts' a noun to the corresponding set, \downarrow does the opposite. In this manner we can shift between the two type-levels 'noun' and 'set', which are both present and frequently used in CML. Nouns and sets are in a sense interchangeable and one could restrict oneself to only one of these categories, without losing expressive power (as is actually done in the set-theoretic formalization). But, nouns (and adjectives) are present in natural language and are amply used in CML and hence they bring a WTT-text nearer to the intuition of a mathematician since they account for the finer details of a piece of mathematics better than in a purely set-theoretic setting. Examples include:

- (\mathcal{C}^{σ}) (a natural number) $\uparrow = \mathbb{N}$, (a divisor of 6) $\uparrow = \{1, 2, 3, 6\}$,⁸ ($\text{Noun}_{x \in \mathbb{R}}(x > 5)$) $\uparrow = \text{Set}_{x \in \mathbb{R}}(x > 5)$.
- (\mathcal{C}^{ν}) $\mathbb{Z}\downarrow$ is 'an integer', ($\text{Set}_{x \in \mathbb{R}^2}(|x| = 1)$) \downarrow is 'a point on the unit circle'.

1.5 Binders

As a third set given beforehand and infinite, we have the set of *binders*. This set is disjoint from both the set of variables and the set of constants. We divide the set \mathbf{B} of binders into five subcategories, depending on the resulting category of the bound expression $\mathbf{B}_{\mathcal{Z}}(\mathcal{E})$ in which the binder occurs:

- ($\mathbf{B}^{\mathfrak{t}}$) Binders giving *terms*, (\mathbf{B}^{σ}) Binders giving *sets*, (\mathbf{B}^{ν}) Binders giving *nouns*,
- (\mathbf{B}^{α}) Binders giving *adjectives*, ($\mathbf{B}^{\mathcal{S}}$) Binders giving *statements*.

Hence, $\mathbf{B} = \mathbf{B}^{\mathfrak{t}}|\mathbf{B}^{\sigma}|\mathbf{B}^{\nu}|\mathbf{B}^{\alpha}|\mathbf{B}^{\mathcal{S}}$.

In the body \mathcal{E} of bound expressions $\mathbf{B}_{\mathcal{Z}}(\mathcal{E})$ various linguistic categories can occur: $\mathcal{E} = \mathfrak{t}|\sigma|\nu|\mathcal{S}$.

Recall that \mathcal{Z} is a declaration, e.g. $x \in \mathbb{N}$. We give now some examples of bound expressions, specifying the appropriate body category \mathcal{E} . The bound expressions are listed according to the category of the binder:

$$- \mathbf{B}_{\mathcal{Z}}^{\mathfrak{t}}(\mathcal{E}) = \min_{\mathcal{Z}}(\mathfrak{t}) | \sum_{\mathcal{Z}}(\mathfrak{t}) | \lim_{\mathcal{Z}}(\mathfrak{t}) | \lambda_{\mathcal{Z}}(\mathfrak{t}) | \lambda_{\mathcal{Z}}(\sigma) | \iota_{\mathcal{Z}}(\mathcal{S}) | \dots$$

⁷ We often use 'sugared' versions of the combination 'constant followed by parameter list'. For example, instead of 'the centre (C)' we write 'the centre of C ', and instead of ' $+(3, 6)$ ' we write the infix formula ' $3 + 6$ '. *Note: These kinds of sugaring are only for the human reader, they are not part of the syntax of WTT.*

⁸ Here again, we used sugaring. We write, as usual, $\{1, 2, 3, 6\}$ for $\text{Set}_{n \in \mathbb{N}}(n = 1 \vee n = 2 \vee n = 3 \vee n = 6)$. However, the notation with Set is the only 'official' WTT-format.

- $B_Z^\sigma(\mathcal{E}) = \text{Set}_Z(\mathcal{S}) | \bigcup_Z(\sigma) | \iota_Z(\mathcal{S}) | \dots$
- $B_Z^\nu(\mathcal{E}) = \text{Noun}_Z(\mathcal{S}) | \text{Abst}_Z(\mathfrak{t}) | \text{Abst}_Z(\sigma) | \text{Abst}_Z(\nu) | \dots$
- $B_Z^\alpha(\mathcal{E}) = \text{Adj}_Z(\mathcal{S}) | \dots$
- $B_Z^\forall(\mathcal{E}) = \forall_Z(\mathcal{S}) | \dots$

In these lists there are several (more or less) new binding symbols: λ , ι , **Noun**, **Abst** and **Adj**. They are explained below. For the binder **Set**, see Section 1.2.

The λ -binder Church's λ introduces *function abstraction*. The format for an expression bound by the λ -binder is: $\lambda_Z(\mathfrak{t}/\sigma)$. Here $\lambda_Z(\mathfrak{t})$ is a term-valued function and $\lambda_Z(\sigma)$ is a set-valued function. For example:

- (\mathfrak{t}) The term $\lambda_{x \in \mathbb{R}}(x^2)$ denotes the squaring function on the reals.
- (σ) The term $\lambda_{n \in \mathbb{N}} \text{Set}_{k \in \mathbb{N}}(k \leq n)$ sends a natural number n to the set $\{0, 1, \dots, n\}$.

The ι -binder Russell's ι is used for a *definite description*: 'the such and such, such that ...'. The general format for an expression bound with the ι -binder is: $\iota_Z(\mathcal{S})$. The result of the binding of a sentence by means of ι can either be a term or a set (therefore we find $\iota_Z(\mathcal{S})$ both in the $B^{\mathfrak{t}}$ - and in the B^σ -list). For example:

- The term $\iota_{n \in \mathbb{N}}(2 < n < \pi)$ describes natural number 3.
- The set $\iota_U: \text{SET}$ ($3 \in U \wedge |U| = 1$) describes the singleton set $\{3\}$. (The declaration $U : \text{SET}$ expresses that U is a set. See also Section 1.7.)

The Noun-binder Since nouns (to be precise: indefinite noun phrases) are first-class citizens in WTT, they are treated similarly to sets. Consequently, next to set comprehension, we allow *noun comprehension*, i.e. the construction of a noun describing a type in noun-format. For noun comprehension we introduce the binder **Noun**. It is used for an *indefinite description*: 'a such and such, such that ...'. Hence, the general format of a phrase with **Noun**-binder is: $\text{Noun}_Z(\mathcal{S})$, i.e. 'a noun saying of Z that \mathcal{S} '. Examples include:

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

The Abst-binder The **Abst**-binder *abstracts* from a term \mathfrak{t} , a set σ or a noun ν and delivers a noun. It is the formal counterpart of the modifier 'for some ...'. Examples of the three kinds of nouns $\text{Abst}_Z(\mathfrak{t}/\sigma/\nu)$:

- (\mathfrak{t}) $\text{Abst}_{n \in \mathbb{N}}(n^2)$ represents 'a term n^2 for some natural number n ', i.e. 'the square of some natural number'.⁹
- (σ) $\text{Abst}_{n \in \mathbb{N}} \text{Set}_{x \in \mathbb{R}}(x > n)$ represents 'a set $\{x \in \mathbb{R} | x > n\}$ for some natural number n ', i.e. 'an interval of the form (n, ∞) ', with $n \in \mathbb{N}$.
- (ν) $\text{Abst}_{n \in \mathbb{N}} \text{Noun}_{x \in \mathbb{R}}(10n \leq x < 10n + 1)$ 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'.

Remark 12

- The **Abst**-binder is useful and compact. It enables one to put the abstraction quantification on the outside of the expression. However, a noun constructed with the **Abst**-binder can always be rewritten into one without it. We show this by rewriting the examples in a form without **Abst**, viz.:

$$(\mathfrak{t}) \text{Abst}_{n \in \mathbb{N}}(n^2) \rightsquigarrow \text{Noun}_{k \in \mathbb{N}} \exists n \in \mathbb{N}(k = n^2),$$

$$(\sigma) \text{Abst}_{n \in \mathbb{N}} \text{Set}_{x \in \mathbb{R}}(x > n) \rightsquigarrow \text{Noun}_V: \text{SET} \exists n \in \mathbb{N}(V = \text{Set}_{x \in \mathbb{R}}(x > n)),$$

$$(\nu) \text{Abst}_{n \in \mathbb{N}} \text{Noun}_{x \in \mathbb{R}}(10n \leq x < 10n + 1) \rightsquigarrow \text{Noun}_{x \in \mathbb{R}} \exists n \in \mathbb{N}(10n \leq x < 10n + 1).$$

Note that in each of these examples, an 'inside' \exists takes over the role of the 'outside' **Abst**.

- In the third case, the **Abst**-binder, transforming a noun into a noun, closely corresponds to the \bigcup -binder, transforming a set into a set. This can be expressed by the following abstract transformation:

$$(\text{Abst}_Z(\nu))^\dagger = \bigcup_Z((\nu)^\dagger).$$

See the third example: $\bigcup_{n \in \mathbb{N}} \text{Set}_{x \in \mathbb{R}}(10n \leq x < 10n + 1)$ is the set of all real numbers in some interval $[10n, 10n + 1)$. This set can also be written in a form without \bigcup , viz.:

$$\bigcup_{n \in \mathbb{N}} \text{Set}_{x \in \mathbb{R}}(10n \leq x < 10n + 1) \rightsquigarrow \text{Set}_{x \in \mathbb{R}} \exists n \in \mathbb{N}(10n \leq x < 10n + 1).$$

Note again the 'inside' \exists , this time in the place of the 'outside' \bigcup .

⁹ Note the difference between **Abst** and λ : the term $\lambda_{n \in \mathbb{N}}(n^2)$ is the function mapping n to n^2 , whereas $\text{Abst}_{n \in \mathbb{N}}(n^2)$ is an arbitrary element of the set $\{n^2 | n \in \mathbb{N}\}$. Note also that ' $\text{Noun}_{n \in \mathbb{N}}(n^2)$ ' is syntactically incorrect, since n^2 is not a statement, and also absurd: its natural reading 'an n in \mathbb{N} such that n^2 ' makes no sense.

The Adj-binder Adjectives, being first-class citizens as well, can be constructed with the Adj-binder. One can read $\text{Adj}_{\mathcal{Z}}(\mathcal{S})$ as: 'the adjective saying of \mathcal{Z} that \mathcal{S} '. For example:

$\text{Adj}_{n \in \mathbb{N}}(\exists k \in \mathbb{N}(n = k^2 + 1))$ is an adjective saying of a natural number that it is a square plus 1. One could give this adjective a name, say 'squary' and hence say things like '5 is squary' or 'Let m be a squary number'.

1.6 Phrases

Now that variables, constants and binders have been treated, we can give an abstract syntax for the various *phrase* categories. Phrases can be terms, sets, nouns or adjectives:

$$\mathfrak{t} = \mathfrak{C}^{\mathfrak{t}}(\vec{\mathcal{P}})|\mathfrak{B}_{\mathcal{Z}}^{\mathfrak{t}}(\mathcal{E})|\mathfrak{V}^{\mathfrak{t}} \quad \sigma = \mathfrak{C}^{\sigma}(\vec{\mathcal{P}})|\mathfrak{B}_{\mathcal{Z}}^{\sigma}(\mathcal{E})|\mathfrak{V}^{\sigma} \quad \nu = \mathfrak{C}^{\nu}(\vec{\mathcal{P}})|\mathfrak{B}_{\mathcal{Z}}^{\nu}(\mathcal{E})|\alpha\nu \quad \alpha = \mathfrak{C}^{\alpha}(\vec{\mathcal{P}})|\mathfrak{B}_{\mathcal{Z}}^{\alpha}(\mathcal{E})$$

The combination $\alpha\nu$ gives a (new) noun which is a combination of an adjective and a noun. Examples include: 'isosceles triangle', 'convergent series'.

1.7 Statements

Abstract syntax for the category of *statements* is: $\mathcal{S} = \mathfrak{C}^{\rho}(\vec{\mathcal{P}})|\mathfrak{C}^{\iota}(\vec{\mathcal{P}})|\mathfrak{B}_{\mathcal{Z}}^{\mathcal{S}}(\mathcal{E})$

Examples of \mathfrak{C}^{ρ} and \mathfrak{C}^{ι} were given in Section 1.4. As an example of $\mathfrak{B}^{\mathcal{S}}$ take $\forall_{x \in \mathbb{N}}(x \geq 0)$.

Typings and declarations A *typing statement* or *typing*, expresses the relation between something and its type. In WTT we have four kinds of typings, depending on the nature of the type. This type can be:

SET (the type of all sets), a set, a noun or an adjective. Each of these statements relates a *subject* (the left hand side) with its *type* (the right hand side, also called the *predicate*). Abstract syntax for the set \mathcal{T} of typing statements (a subcollection of \mathcal{S}) is: $\mathcal{T} = \sigma : \text{SET} | \mathfrak{t} : \sigma | \mathfrak{t} : \nu | \mathfrak{t} : \alpha$

In words: ' σ is a set', ' \mathfrak{t} is an element of σ ', ' \mathfrak{t} is ν ', ' \mathfrak{t} is α '. Examples of these four cases include:

' $\text{Set}_{n \in \mathbb{N}}(n \leq 2) : \text{SET}$ ', ' $3 \in \mathbb{N}$ ',¹⁰ ' $AB : \text{an edge of } \triangle ABC$ ', ' $\lambda_{x \in \mathbb{R}}(x^2) : \text{differentiable}$ '.

Clearly, \mathcal{T} is a subcollection of $\mathfrak{C}^{\rho}(\vec{\mathcal{P}})$, the set of relational statements, with symbol ':' as special element in \mathfrak{C}^{ρ} . At its turn, a subcollection of the typings is formed by the *declarations*, \mathcal{Z} , where the subject is a variable:¹¹ $\mathcal{Z} = \mathfrak{V}^{\sigma} : \text{SET} | \mathfrak{V}^{\mathfrak{t}} : \sigma | \mathfrak{V}^{\mathfrak{t}} : \nu$.

\mathfrak{V}^{σ} or $\mathfrak{V}^{\mathfrak{t}}$ is the *introduced* or *declared* variable. Subscripts of binders (Section 1.5) can be taken from \mathcal{Z} .

1.8 Definitions

An important category in WTT is the category \mathcal{D} of *definitions*. Definitions introduce a new constant.¹²

We distinguish between *phrase definitions* \mathcal{D}^{ρ} and *statement definitions* $\mathcal{D}^{\mathcal{S}}$: $\mathcal{D} = \mathcal{D}^{\rho} | \mathcal{D}^{\mathcal{S}}$.

Phrase definitions fix a constant representing a phrase. Statement definitions also introduce a constant, but embedded in a statement. In definitions, the newly defined constant is separated from the phrase or statement it represents by the symbol ':='.

Remark 13 *We have decided not to include definitions for binders. The main reason is, that the set of binders used in mathematics is relatively stable, new binders are only seldom necessary. On the other hand, it is not very hard to include binder definitions in the syntax.*¹³

¹⁰ As this example shows, we often replace $\mathfrak{t} : \sigma$ by $\mathfrak{t} \in \sigma$. (This syntactic sugaring is not part of the 'real' syntax.)

¹¹ There are no declarations with an adjective as type.

¹² We only regard non-recursive definitions.

¹³ Abstract syntax for binder definitions could be: $\mathcal{D}^{\mathfrak{B}} = \mathfrak{B}_{\mathcal{Z}}^{\mathfrak{t}/\sigma/\nu/\alpha/\mathcal{S}}(\mathcal{E}) := \mathfrak{t}/\sigma/\nu/\alpha/\mathcal{S}$.

Phrase definitions There are four kinds of phrase definitions:

$$\mathcal{D}^\varphi = \mathcal{C}^{\mathbf{t}}(\vec{V}) := \mathbf{t} \mid \mathcal{C}^\sigma(\vec{V}) := \sigma \mid \mathcal{C}^\nu(\vec{V}) := \nu \mid \mathcal{C}^\alpha(\vec{V}) := \alpha$$

Note that the parameters occurring in the left hand side of a definitions must be *variables*. The reason is of course, that a definition should be as general as possible and hence may 'depend' on a list of variables. Later, when *using* the definition in a certain situation, all these variables must be 'instantiated' according to that situation. Examples of the four kinds of phrase definitions are:

- (\mathbf{t}) *the arithmetic mean* of a and $b := \iota_{z \in \mathbb{R}}(z = \frac{1}{2}(a + b))$,
- (σ) $\mathbb{R}^+ := \text{Set}_{x \in \mathbb{R}}(x > 0)$,
- (ν) *a unit* of G with respect to $\cdot := \text{Noun}_{e \in G}(\forall a \in G(a \cdot e = e \cdot a = a))$,
- (α) *prime* := $\text{Adj}_{n \in \mathbb{N}}(n > 1 \wedge \forall k, l \in \mathbb{N}(n = k \cdot l \Rightarrow k = 1 \vee l = 1))$.

The variable lists in the four examples are: (a, b) , $()$, (G, \cdot) , $()$. These variables must be introduced ('declared') in a context (see Section 1.9). For the first definition, such a context can be e.g. $a : \mathbb{R}, b : \mathbb{R}$. For the third definition the context is: $G : \text{SET}, \cdot : G \rightarrow G$. Both contexts consist of *declarations* only.

However, definitions may also depend on *assumptions*. This is reflected in Section 1.9, where it is stated that a context consists of a list of declarations *and* assumptions. For an example, take the definition of the natural logarithm: $\ln(x) := \iota_{y \in \mathbb{R}}(e^y = x)$.

Here variable x has to be declared in a context, for example: $x : \mathbb{R}, x \geq 0$. This is a declaration: $x : \mathbb{R}$, introducing x of type \mathbb{R} , followed by an assumption: $x > 0$, stating that the introduced x is positive.

In general, definitions are not complete without such a context. That is to say, the 'ground has to be prepared' before the actual definition is stated.

Examples of *instantiations* of the first example definition are: 'the arithmetic mean of 3 and 6', or, for given x : 'the arithmetic mean of x and x^2 '.

Note that the variables in the variable list \vec{V} of a definition are the same as the declared variables in the context, and listed in the same order. (The *assumptions* occurring in the context are not accounted for in the parameter list of a WTT-constant.)¹⁴ E.g., in the first example, the parameter list of 'the arithmetic mean' is ' (a, b) ', which is exactly the same as the list of the declared variables occurring in the context ' $a \in \mathbb{R}, b \in \mathbb{R}$ '.¹⁵

Statement definitions We introduce the following category of statement definitions:

$$\mathcal{D}^S = \mathcal{C}^\rho(\vec{V}) := S \quad \text{The newly defined constant in this sentence definition is } \mathcal{C}^\rho.$$

Remark 14 *We only consider definitions for relational constants, hence we do not include definitions for logical constants, because the latter are hardly ever necessary. However, it is easy to extend the syntax, when desired, in order to allow definitions of new logical constants.*

Example: a is parallel to $b := \neg \exists P: \text{a point}(P \text{ lies on } a \wedge P \text{ lies on } b)$.

Again, we need a context to make the definition self-contained. This context is, for example:

$a : \text{a line}, b : \text{a line}$ (i.e., 'Let a and b be lines').

Note that the notion 'is parallel to' can only be considered as a two-place *relation*, and therefore its definition must be a statement definition. In other cases, things are not so clear and the WTT-user has to make a choice. For example, the definition of ' x is the opposite of y ' can be treated as a statement definition, similarly to the one above: (1) x is the opposite of $y := x + y = 0$, with context consisting of e.g. $x : \mathbb{R}, y : \mathbb{R}$.

But it is also possible to define the same notion in a *phrase* definition (to be precise: a term definition), as follows: (2) *the opposite of $y := \iota_{x \in \mathbb{R}}(x = -y)$* . Now the context is only $y : \mathbb{R}$.

The difference is whether one considers '(is) the opposite of' to be a *relation* or a *constant*. The latter choice allows more freedom, since a phrase definition can always be used as part of a sentence, but not the

¹⁴ In Type Theory, however, variables 'inhabiting' assumptions are added to the variable list.

¹⁵ Since such a parameter list can be reconstructed from the context in which the definition is embedded, these parameter lists (or parts of it) are often omitted.

other way round. For example, if one chooses for (2), then it is possible to instantiate this definition in a phrase: 'the opposite of 5', but also in a sentence: '-5 = the opposite of 5'.

Likewise, it is more flexible to define the phrase (in this case: the noun) 'a unit of (G, \cdot) ' than to define the statement 'e is a unit of (G, \cdot) '. In the first case we can use the noun in a statement, e.g. (for known $a \in G$): ' a : a unit of (G, \cdot) ' (to be read as: ' a is a unit of (G, \cdot) '). In the second case, the sentence definition does not justify the use of the undefined noun 'a unit of ...'.

1.9 Contexts

A *context* Γ is a list of declarations (from \mathcal{Z}) and statements (from \mathcal{S}): $\Gamma = \emptyset \mid \Gamma, \mathcal{Z} \mid \Gamma, \mathcal{S}$

A declaration occurring in a context represents the *introduction of a variable* of a certain (already known) type. A statement S in a context stands for an *assumption*.¹⁶

1.10 Lines

A *line* \mathbf{l} contains either a statement or a definition, relative to a context: $\mathbf{l} = \Gamma \triangleright S \mid \Gamma \triangleright \mathcal{D}$

The symbol \triangleright is a separation marker between the context and the statement or definition (in that context).

1.11 Books

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

¹⁶ According to our syntax, such an assumption –being a statement– can also be a declaration – being a statement with a *variable* as subject. However, typing rules for contexts will ensure that there is no *newly* introduced variable in an assumption, so it is always clear whether a context statement represents the introduction of a new variable, or an assumption.