

ABSTRACT

How to write and work electronic mathematical text without using a WYSIWYG software, or without typing LaTeX, or even XML? How the text can be reasonably understandable by humans? This memo aims at proposing a system focusing on mathematical symbolic or semantic rather than typographic rendering. Of course, the system defines several character combinations, but only for their mathematical meanings and for compactness purpose. HTML conversion from written ASCII text, for common rendering, is not envisaged, even if feasible. The challenge is to work mathematical text in a minimalist environment: an ASCII text editor.

This notation system is blessed MathSAN, an acronym for Mathematics with Symbolic Ascii Notation.

This version MathSAN 0.0 does not pretend to cover all domains of maths. It is a speculative system, that has to be experienced, and then evolved if proven valuable in some contexts. The license allows anybody to continue this work.

lucas.borboleta@free.fr 2013-07-14

ARITHMETICAL NOTATIONS

- A1) "+" represents the addition operator.
- A2) "-" represents the subtraction operator.
- A3) "*" represents the multiplication operator.
- A4) "." is a reserved meta character like in "A .c B", and should not be used for representing multiplication operator; see also "set notations".
- A5) "/" represents the division operator.
- A6) "**" represents the power operator. Example: $x^{**2} = x \times x$

CONSTANTS NOTATIONS

- C1) "inf" is referring infinity.
- C2) "pi" is referring 3.14...
- C3) "e" is referring 2.71...

ORDER NOTATIONS

- O1) "<" is the less-than relation.
- O2) ">" is the greater-than relation.
- O3) "<=" is the less-or-equal-than relation.
- O4) ">=" is the greater-or-equal-than relation.

LOGIC NOTATIONS

- L1) "=" represents the is-equal-to relation.

- L2) \neq represents the is-not-equal-to relation.
- L3) "and", "or", "not" have their expected meanings.
- L4) \implies represents if-then or "implies".
- L5) \iff represents is-equivalent-to.
- L6) $:=$ represents is-equal-to-by-definition.
- L7) $:\iff$ represents is-equivalent-to-by-definition.

SET NOTATIONS

- S1) ".A" represents the universal quantifier. Example:
 $\forall x \in \text{NAT}, \forall y \in \text{NAT}, x+y=0 \implies x=y=0$
- S2) ".E" represents the existential quantifier. Example:
 $\exists x \in \text{INT}, \exists y \in \text{INT}, x+y=0$
- S3) ".e" represents the is-element-of relation. Example:
 $a \in \{a, b, c\}$
- S4) "./e" represents the is-not-element-of relation. Formally:
 $x \notin S \iff \text{not}(x \in S)$
- S5) ".c" represents the is-contained-in relation. Formally:
 $A \subset B \iff \{x: x \in A \implies x \in B\}$
- S6) "./c" represents the is-not-contained-in relation. Formally:
 $A \not\subset B \iff \text{not}(A \subset B)$
- S7) ".i" represents the intersection infix operator. Formally:
 $A \cap B := \{x: x \in A \text{ and } x \in B\}$
- S8) ".u" represents the union infix operator. Formally:
 $A \cup B := \{x: x \in A \text{ or } x \in B\}$
- S9) "-" represents the difference operator. Formally:
 $A - B := \{x : x \in A \text{ and } x \notin B\}$
- S10) ".n" represents the relative-not postfix operator. Formally:
 $A.n := \{x \in E : x \notin A\}$ where E is the implicit universe.
 $A \cup A.n = E$
- S11) VOID refers to the empty set. Example:
 $A \cap A.n = \text{VOID}$
- S12) "pset[E]" represents the power-set of a set. Formally:
 $\text{pset}[E] := \{A : A \subset E\}$
 $\text{pset}[\{a,b\}] = \{\text{VOID}, \{a\}, \{b\}, \{a,b\}\}$
- S13) ".x" is the cartesian product infix operator. Formally:
 $A \times B = \{(x,y) : x \in A \text{ and } y \in B\}$
- S14) " $|E|$ " and "card[E]" are representing the cardinal of a set E. Example:
 $|\{a,b,c\}| = \text{card}[\{a,b,c\}] = 3$

NUMBERS NOTATIONS

- N1) NAT is referring to natural integers. Example:
 .A (x,y) .e NAT .x NAT, x+y=0 ==> x=y=0
- N2) INT is referring to positive and negative integers. Example:
 .A x .e INT, .E y .e INT, x+y=0
- N3) RAT is referring to rational numbers. Example:
 .A x .e RAT, x /= 0, .E y .e RAT, x*y=1
- N4) REAL is referring to real numbers. Example:
 .A x .e REAL, x >= 0, .E y .e REAL, y*y=x
- N5) COMP is referring to complex numbers. Example:
 .E x .e COMP, x**2 + 1 = 0

FUNCTION AND MAP NOTATIONS

- F1) "map[sets;variables;conditions]" represents a map. Example:
 abs:= map[REAL --> REAL; x |--> y ; y=x if x>=0 ; y=-x if x<= 0]
- F2) "dom[map]" represents the domain of a map: the starting set of the map.
Formally:
 m := map[dom[m] --> codom[m]; ...]
- F3) "codom[map]" represents the co-domain of a map: the arrival set of the map.
Formally:
 m := map[dom[m] --> codom[m]; ...]
- F4) "fun[sets;variables;conditions]" represents a function, as a partial map "f".
Example:
 inverse := fun[REAL --> REAL; x |--> y ; y = 1/x]
- F5) "defdom[f]" represents the definition-domain of a function "f", a subset of its domain. Example:
 REAL - {0} = defdom[inverse] .c dom[inverse]
- F6) "id[set]" is the identity map. Formally:
 id[E] := map[E --> E ; x |--> x]
- F7) ".o" is the map composing infix operator. Formally:
 (f .o g) (x) = f(g(x))
- F8) ".(n)" or ".n" is the map iterating postfix operator. Formally:
 f.(0) := f
 .A n .e NAT, f.(n+1) := f .o f.(n)
- F9) "(-1)" or ".-1" is the inverse postfix operator. Example:
 square := fun[REAL --> REAL ; x|-->y ; x>=0 ; y=x*x]
 sqrt := fun[REAL --> REAL ; x|-->y ; x>=0 ; x=y*y]
 square.(-1) = sqrt

INDEXING NOTATIONS

- I1) "x@i", like in "{x@i : i .e I}", means "x(i)" where "x" is an indexing map.
Formally:
 x := map[I --> X ; i|--> x(i) = x@i] and dom[x]=I.

I2) "uset[set-term;conditions]" represents the union of set terms under conditions. Example: $\text{uset}[A@i ; i .e I]$

I3) "iset[set-term;conditions]" represents the intersection of set terms under conditions. Example: $\text{iset}[A@i ; i .e I]$

I4) "xset[set-term;conditions]" represents the cartesian product of set terms under conditions. Example: $\text{xset}[A@i ; i .e I]$

I5) "sum[term;conditions]" represents the sum of indexed terms under conditions. Example:

$$\text{sum}[i .e \text{NAT} ; 1 \leq i \leq n] = n*(n+1)/2$$

I6) "prod[term;conditions]" represents the product of indexed terms under conditions. Example:

$$\text{prod}[i : i .e \text{NAT} \text{ and } 1 \leq i \leq n] = n!$$

REAL ANALYSIS NOTATIONS

RA1) "lim[term;conditions]" represents limit of term under conditions. Example: $\text{lim}[1/x ; x \rightarrow \text{inf}] = 0$

RA2) "der[function;variables.iterations]" represents the derivative, optionally iterated, of a function regarding given variables. Examples:

$$\text{der}[x^3*y^2 ; x.2 ; y] = \text{der}[3*x^2*2*y ; x] = 12*x*y$$

$$\text{der}[x^n ; x.(n)] = n!$$

RA3) "int[term;conditions]" represents an integral. Example:

$$\text{int}[x^2*dx ; a \leq x \leq b] = (b^3 - a^3)/3$$

PROBABILITY NOTATIONS

P1) "prob[event]" represents the probability of event.

P2) "prob[random-variable;conditions]" represents the probability that a random variable meets the conditions.

MATRIX NOTATIONS

M1) ".t" is the transpose postfix operator in linear algebra. Formally:

$$M.t@(i,j) = M@(j,i) \text{ where } (i,j) \text{ are indexing the elements of the matrix } M.$$

PHILOSOPHY

MathSAN is not designed for post-processing like conversion into HTML.

MathSAN is designed to be close to notations of mathematical books, but is not obsessed by a complete mimic of its typography:

- Simple mathematical symbols are represented as ASCII words like ".e" or "=>", with "." a prefixing symbol letter for disambiguation, and with surrounding space for clarity. However, postfix symbols, like ".n" in "A.n", are better used without inserting space.

- Complex mathematical notations are represented by a named template with arguments, akin to functional programming language, like "map[...]" in "square := map[REAL-->REAL; x|-->y; y=x**2]".
- Template are identified by reserved names in lower case.
- Notable sets are represented by reserved names in upper case.
- Notable constants are represented by reserved names in lower case.
- Reserved names are often abbreviated for compactness.

DISCUSSION

Why such a notation system? What could be such minimalist environment? Why not just using pen and paper, when the environment is extremely minimalist?

Here are tentative arguments. Here is an example of minimalist environment: working mathematical explanation embedded in C+ source without needing or wanting post-processing, like Doxygen. About pen and paper : even a light electronic environment, like a tablet disconnected from internet, provides copy-paste features that balances the expressiveness of pen-paper environment.

LICENSE

"CC BY-SA 3.0"

This work by Lucas Borboleta (<http://lucas.borboleta.blog.free.fr>) is licensed under Creative Commons Attribution-ShareAlike 3.0 Unported License (<http://creativecommons.org/licenses/by-sa/3.0/>).