数学

1

鴎鵜みね

（最終更新日：2022 年 2 月 20 日）

はじめに

この pdf は，個人的な数学のノートです．とある動画シリーズの制作に付随した補助資料として用いることを目的としています．

内容は，ブルバキを基準とした古典的な数学全般です．できるだけ厳密かつ簡潔に書くように心がけましたが，その分種々の数学的内容の動機や背景，並びに注意などはほとんど記載できていません（本文の流れを遮らない程度の記載はしましたが，通常の書籍などに比べれば遥かに少ないです）．ただし，後々 Appendix などを追加して解説を行うことはあるかもしれません．

本第 1 巻では古典一階述語論理，クラス理論，集合論を扱います．

その他

この pdf の紙面サイズは B5 です．

目次

I

古典一階述語論理

第 1 章

形式体系

1.1

記述

1.1.1　メタ言語と対象言語

紙や板やスクリーンなどに書かれ，あるいは表示されるような何らかを，ここでは**記述**（description）と呼び，また記述に何らかの処理を施すことを**操作**（operation）と呼ぶことにします．また，記述たちが集まり組織化されたものを**記述体系**（description system）と呼ぶことにします．

メタ言語（metalanguage）と呼ばれる記述体系があります．このノートでは，メタ言語内の記述についての規定を「メタ規則」の項目で掲げます．ここでは「メタ規則」を満たすメタ言語として日本語を採用します．

メタ言語を用いて新しい記述体系についての記述を行うとき，メタ言語で記述されるその記述体系を**対象言語**（object language）と呼びます．このノートでは，対象言語について述べる記述を「規則」の項目で掲げます．また，いくつかの「規則」から正しいと分かる事実，ならびにそれがなぜ正しいかの説明を，それぞれ「メタ定理」「メタ証明」の項目で掲げます．

注意 1.1.1

(1) 数学は，いわゆる「数学的記述」と「数学的記述の操作」からなる体系とみなすことができます．このうち前者を担うのが対象言語，後者（のルールの明示）を担うのがメタ言語です．また，対象言語の記述そのものに関するルール（=文法）の明示もメタ言語の役割となりますが，これはすなわち対象言語の構築を意味します．例えるなら，対象言語は積み木，メタ言語は積み木遊びのルールブックです．[*1]

(2) このように，記述体系どうしは「記述（構築）する・される」という階層関係にあります．記述を扱う際は，それがどの階層に属しているのかを明確にする必要があります．特に，メタ言語内の記述と対象言語内の記述は区別しなければなりません．

(3)「規則」「メタ定理」「メタ証明」はいずれもメタ言語で記述されています．一方，「メタ規則」はメタ言語・対象言語のどちらによる記述でもありません．なぜなら，「メタ規則」はメタ言語内の記述について述べた記述であり，メタ言語よりも一つ階層が上だからです．[*2] □

メタ言語に存在するべき記述を示します．[*3]

[*1] というよりも，実際積み木の状態を記述と考えれば，（現実的ではありませんが）文字通り積み木で数学を行えてしまいます．

[*2] ただし，「メタ規則」がどのような記述体系に属しているのかを考える必要はありません．実際に数学を展開する上では，ある特定のメタ言語を用いることになります．従って，もしある記述体系をメタ言語として用いたいとき，（それが「メタ規則」を満たすことを暗に確認したうえで）あたかも「メタ規則」のことは知らないかのようにそれをメタ言語として使ってしまえば，見かけ上は「メタ規則」なしで済ませられるからです．

[*3] このうち ′ は補助的な記述であり，用いなくてもそれほど支障はありません．一方メタ変数については，全く用いずに済ますこともできますが，一般的な議論ができなくなってしまうためここでは考えません．

メタ規則 1.1.2（′，メタ変数）
以下の記述がメタ言語内に存在する．

(1) ′.
(2) **メタ変数**（metavariable）と呼ばれる記述．[*4] □

メタ変数は，何らかの記述の上を動く「変数」のようなものです．メタ変数を含む記述の扱いについて述べます．

メタ規則 1.1.3（メタ変数の書き換え）

(1)「メタ言語内のある記述を，メタ言語・対象言語内の別の記述に**書き換える**（rewrite）」という操作が既知である．[*5]
(2) メタ変数の各々には，メタ言語・対象言語内の特定の記述たちがあらかじめ定まっている．これを**メタ変数の範囲**（range of metavariable）と呼ぶ．
(3) メタ変数は，その範囲内の任意の記述に書き換えることができ，かつ，それ以外の記述に書き換えることはできない．
(4) 記述の中に同じメタ変数があるときは，それらを全て同じ記述に書き換える．
(5) メタ変数を含む記述は，以上の書き換えの結果得られる全ての記述である． □

例 1.1.3.1
メタ変数 $\boldsymbol{M}$ と $\boldsymbol{N}$ の範囲が $!, \mathcal{E}, PPP$[*6]という記述であるとき，記述

$$\boldsymbol{M} \text{ はアレ}$$

とは，記述

$$!\text{はアレ}, \mathcal{E} \text{ はアレ}, PPP \text{ はアレ}$$

である． □

問題 1.1
例 1.1.3.1 と同じ条件のもとで，記述

$$\boldsymbol{M} \text{ と } \boldsymbol{N} \text{ と } \boldsymbol{N}$$

はどのような記述か． □

注意 1.1.4
メタ変数を含む記述はいくつかの記述を一挙に表す「テンプレート」として考えることができますが，同時に，この「テンプレート」をもって実際に全ての記述が列挙されているかのように扱うべきです．特に，メタ変数の範囲となる記述が無数にある（→メタ規則 1.1.19）場合，メタ変数を含む記述は，その書き換え結果として得られる無数の記述を表すことになります． □

以降，「○○の範囲は××である」を「○○は××である」と書きます．[*7]
メタ言語では（主に見やすさのために）しばしば記述の別記を行います．ただし，ある記述に対して既に行っている別記と同じ別記を，他の記述に対して行うことはできません．

[*4] **統語変数**（syntactical variable）とも．
[*5] 「代入する」というのが普通ですが，ここでは対象言語における「代入」（→ [執筆中]）と区別するために「書き換える」と呼んでいます．
[*6] ,（コンマ）は各々の記述の区切りを表すものであり，この主張には含まれていません．以降も同様です．
[*7] 明記していませんが，○○にはメタ変数，××には記述が入ります．

メタ規則 1.1.5（⇒，&）
メタ言語内の記述「ならば」「かつ」をそれぞれ ⇒ と & で表す.[*8] □

別記は望めばいつでも元の記述に戻すことができます．以降，メタ言語においてはこうした別記が独断的に導入されますが，それらがメタ言語内の記述内容や対象言語の構築に影響を与えないようなものであることは認めるものとします.

注意 1.1.6
メタ変数を含む記述と同じように，別記による記述も，実際に本来の記述であるかのように扱うべきです．ところで，メタ言語では直観的に別記という手法を導入しましたが，同様のことを対象言語で行えるかどうかは明らかではありません．私たちは対象言語内の記述そのものに注目しているのに，別記はその記述自体を変化させてしまうからです．しかし，対象言語でも似たことが可能であることが [執筆中] で示されます．そしてこれは「定義」などの話へとつながっていきます. □

1.1.2　数字

私たちの知っているような「数字」がメタ言語内に存在すると，対象言語についての一般的な議論ができて便利です.[*9]
まず「数字」がメタ言語内に存在する場合を考えます．その場合，以下が成り立っている必要があります.

メタ規則 1.1.7（数字）

(1) 以下の記述が存在する.
  (i) **数字**（numeral）と呼ばれる記述.
  (ii) メタ変数 $\boldsymbol{n}, \boldsymbol{m}, \boldsymbol{i}, \boldsymbol{j}, \boldsymbol{k}$ およびこれらに装飾を加えたもの.[*10]これらは数字である.

(2) 以下の記述が既知である.
  (i) 記述が $\boldsymbol{n}$ 個ある.
  (ii) 記述が無数にある.
  (iii) $\boldsymbol{n} < \boldsymbol{m}, \boldsymbol{n} + \boldsymbol{m}$.
  (iv) $\boldsymbol{n} < \boldsymbol{m}$ であるとき，$\boldsymbol{n} - \boldsymbol{m}$. □

注意 1.1.8
数字は [執筆中] で登場する「自然数」とは異なります.「自然数」は対象言語に属していますが，数字はメタ言語に属しています.「数字」は，もっぱら記述を数える（→ 1.1.14）のに用いられる識別可能な「印」たちに過ぎません. □

次に，こうした記述がメタ言語に存在しない場合，もしくはいくつかの記述が既知でない場合に具体的に「数字」を構成する方法を示します.

[*8] これは [執筆中] で登場する → と ∧ とは異なります．⇒, & はメタ言語に属していますが，→, ∧ は対象言語に属しています.
[*9] 実際はこの「数字」も用いずに済ませることができますが（厳密にはそうすべき)，やはり対象言語について一般的に語ることができなくなるためここでは考えません．数字を用いない場合にどのようなデメリットがあるのかについては [執筆中] で具体的に示されます.
[*10] ここで「装飾を加える」とは，数字や ′ を添字として加えることをいいます.

メタ規則 1.1.9（0，|）
記述 $\mathbf{0}, |$ がメタ言語に存在する. □

メタ規則 1.1.10（数字，数字列）
数字（numeral）と呼ばれる記述は以下で決定される.

(1) $\mathbf{0}$ は数字である.
(2) (1) で数字であるとされた記述の右側に | を書き並べた記述は数字である.
(3) (2) で数字であるとされた記述の右側に | を書き並べた記述は数字である.
(4) (3) で数字であるとされた記述の右側に | を書き並べた記述は数字である.
(5) (4) で数字であるとされた記述の右側に | を書き並べた記述は数字である.
(6) $\cdots$

以下同様に続く.
以上で数字であると先に分かった順に数字を左から列挙したものを**数字列**（numeral sequence）と呼ぶ.
□

例 1.1.10.1
数字列は $\mathbf{0}, \mathbf{0}|, \mathbf{0}||, \mathbf{0}|||, \mathbf{0}||||, \mathbf{0}|||||, \cdots$ と続く. □

メタ規則 1.1.11（1 から 9）
以降，以下の別記を行う.

(1) $\mathbf{0}|$ を $\mathbf{1}$ で表す.
(2) $\mathbf{0}||$ を $\mathbf{2}$ で表す.
(3) $\mathbf{0}|||$ を $\mathbf{3}$ で表す.
(4) $\mathbf{0}||||$ を $\mathbf{4}$ で表す.
(5) $\mathbf{0}|||||$ を $\mathbf{5}$ で表す.
(6) $\mathbf{0}||||||$ を $\mathbf{6}$ で表す.
(7) $\mathbf{0}|||||||$ を $\mathbf{7}$ で表す.
(8) $\mathbf{0}||||||||$ を $\mathbf{8}$ で表す.
(9) $\mathbf{0}|||||||||$ を $\mathbf{9}$ で表す.
(10) $\ldots$

以下同様に続く.[*11] □

例 1.1.11.1
数字列は $\mathbf{0}, \mathbf{1}, \mathbf{2}, \mathbf{3}, \mathbf{4}, \mathbf{5}, \ldots$ と続く. □

数字の上を動くメタ変数を導入します.

メタ規則 1.1.12（$\boldsymbol{n}, \boldsymbol{m}, \boldsymbol{i}, \boldsymbol{j}, \boldsymbol{k}$）
メタ変数 $\boldsymbol{n}, \boldsymbol{m}, \boldsymbol{i}, \boldsymbol{j}, \boldsymbol{k}$ およびこれらに装飾を加えたものは数字である. □

注意 1.1.13
このノートでは，数字は太字のアラビア数字で，メタ変数は数字以外の太字で表記します. □

[*11] これ以下の別記は明示していませんが，$\mathbf{0}, \mathbf{1}, \mathbf{2}, \mathbf{3}, \mathbf{4}, \mathbf{5}, \mathbf{6}, \mathbf{7}, \mathbf{8}, \mathbf{9}$ の組み合わせを別記として採用して構いません．特に，暗黙の了解として，私たちの親しんでいるのと同じ要領で $\mathbf{10}, \mathbf{11}, \ldots$ と表され続けると考えることにします．しかし，全く異なる別記法を採用してももちろん問題ありません．各々が視覚的に識別できることだけが重要だからです.

数字が得られたので「数える」という操作を行えるようになりました．

メタ規則 1.1.14（数える）
「メタ言語・対象言語内のあらかじめ指定した記述たちの各々に，$\mathbf{1}$ から始めて数字列の左にある順に数字を割り振る」という操作を，その記述を**数える**（count）という． □

メタ規則 1.1.15（個数，$\boldsymbol{n}$ 個ある）
「メタ言語・対象言語内のあらかじめ指定した記述たちを数えるとき，最後に割り振った数字が $\boldsymbol{n}$ である」ことを，その記述の**個数**（amount）もしくは**回数**（time）が $\boldsymbol{n}$ であるという．あるいは，その記述は **$\boldsymbol{n}$ 個ある**（$\boldsymbol{n}$ description(s)）もしくは **$\boldsymbol{n}$ 回ある**（$\boldsymbol{n}$ time(s)）という． □

例 1.1.15.1
a, b, c, d, e, f, g の個数を求める．メタ規則 1.1.10，メタ規則 1.1.11 より，数字列は

$$\mathbf{0}, \mathbf{1}, \mathbf{2}, \mathbf{3}, \mathbf{4}, \mathbf{5}, \mathbf{6}, \mathbf{7}, \ldots$$

と続く．メタ規則 1.1.14 に従って，

$$\mathbf{1} \text{ を } a \text{ に，} \mathbf{2} \text{ を } b \text{ に，} \mathbf{3} \text{ を } c \text{ に，} \mathbf{4} \text{ を } d \text{ に，} \mathbf{5} \text{ を } e \text{ に，} \mathbf{6} \text{ を } f \text{ に，} \mathbf{7} \text{ を } g \text{ に}$$

割り振ると，最後に割り振った数字は $\mathbf{7}$ である．従って，個数は $\mathbf{7}$ である． □

メタ規則 1.1.16（$<, +, -$）
以降，以下の別記を導入する．

(1)「数字列で $\boldsymbol{n}$ は $\boldsymbol{m}$ より左にある」および「数字列で $\boldsymbol{m}$ は $\boldsymbol{n}$ より右にある」を $\boldsymbol{n} < \boldsymbol{m}$ で表す．
(2)「数字列での $\boldsymbol{n}$ の $\underbrace{\text{右隣の右隣の……右隣の}}_{\boldsymbol{m}\text{ 回}}$ 数字」を $\boldsymbol{n} + \boldsymbol{m}$ で表す．
(3)「数字列での $\boldsymbol{n}$ の $\underbrace{\text{左隣の左隣の……左隣の}}_{\boldsymbol{m}\text{ 回}}$ 数字」を $\boldsymbol{n} - \boldsymbol{m}$ で表す．

□

対象言語内の記述が $\mathbf{0}$ 個ある（何もない）ということを定めます．

規則 1.1.17（空列）
空列（empty string）と呼ばれる記述が対象言語内に存在する． □

規則 1.1.18（0 個ある）
対象言語内の記述の個数が $\mathbf{0}$，もしくは記述が **$\mathbf{0}$ 個ある**（$\mathbf{0}$ description）とは，その記述が空列のみであることをいう． □

記述が無数にあるということも定めます．

メタ規則 1.1.19（無数にある）
メタ言語・対象言語内の記述が**無数にある**（infinitely exist）とは，それを数えるとき，数字列のどれだけ右にある数字までを割り振ってもまだ数字を割り振られていない記述が存在することをいう． □

いわゆる「帰納法」について述べます．この無数の記述を

メタ規則 1.1.20（メタ帰納法）
執筆中 □

注意 1.1.21
メタ帰納法についても，対象言語における「数学的帰納法」と混同しないようにしてください． □

1.2

一般の形式体系

1.2.1　記号と記号列

ここからは対象言語の構築に取り掛かります．同時に，数学そのものを特徴づける枠組みである「形式体系」を最も一般的な形で定めていきます．

メタ規則 1.2.1（形式体系）
形式体系（formal system）$\mathscr{S}$ は以下からなる．

(1) $\mathscr{S}$ の**記号**（symbol）と呼ばれる記述．これは対象言語に属する．
(2) $\mathscr{S}$ の**構成規則**（formation rule）と呼ばれる記述．これはメタ言語に属する．
(3) $\mathscr{S}$ の**公理**（axiom）と呼ばれる記述．これは対象言語に属する．
(4) $\mathscr{S}$ の**推論規則**（inference rule）と呼ばれる記述．これはメタ言語に属する． □

規則 1.2.2（記号列）
$\mathscr{S}$ の**記号列**（string）は以下で決定される．

(1) $\boldsymbol{n}$ 個の $\mathscr{S}$ の記号を左から右に並べてできる記述は記号列である． □

例 1.2.2.1
空列が $\mathscr{S}$ の記号列であることを示す．対象言語内の $\mathbf{0}$ 個の記述とは空列であり，記号は対象言語内の記述であるから，$\mathbf{0}$ 個の記号とは空列である．従って規則 1.2.2 より，空列は $\mathscr{S}$ の記号列である． □

例 1.2.2.2
$\#, =, ?$ が $\mathscr{S}$ の記号であるとき，$\# ==$ は $\mathscr{S}$ の記号列である． □

メタ規則 1.2.3
メタ変数 $\boldsymbol{S}, \boldsymbol{T}, \boldsymbol{U}$ およびこれらに装飾を加えたものは記号列である． □

ある記号列が何らかの記号列に「含まれている」ということを定めます．

規則 1.2.4（同一列）
$\boldsymbol{S_1}$ の**同一列**（identical string）は以下で決定される．

(1) □

第 I 部の参考文献

[1] N. Bourbaki, *Éléments de mathématique: Théorie des ensembles*, Springer, 2006

[2] George Tourlakis, *Lectures in logic and set theory: Volume I: Mathematical logic*, Cambridge University Press, 2003

[3] George Tourlakis, *Mathematical logic*, John Wiley & Sons, 2008

[4] 新井敏康,『数学基礎論　増補版』, 東京大学出版会, 2021

索引

Symbols

か

さ

た

ま