数学

1

鴎鵜みね（@km_ine）

（最終更新日：2022 年 3 月 7 日）

はじめに

この pdf は個人的な数学のノートです．同時に，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も目的としています．

内容は，ブルバキを基準とした古典的な数学全般を予定しています．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．[*1]数学を解説するというよりも，むしろ数学が構築されていく様子を確かめるといった方が正確かもしれません．ただし，後々 Appendix などを追加して解説を行う可能性もあります．

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．

[*1] 極めて重要な注意事項に限っては本文の流れを遮らない程度に記載しましたが，通常の概説書などに比べれば遥かに少ないです．本当はそれらも省きたいと思っています．

目次

1

古典一階述語論理

第 1.1 章

形式体系

1.1.1

記述

a. 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**と呼び，また記述に何らかの処理を施す（何もしなくても良い）ことを**操作**と呼ぶことにします．また，いくつかの記述の集まりを**記述体系**と呼ぶことにします．以降，ある記述体系に含まれる記述を，単にその記述体系「内の」記述と呼びます．

あらゆる記述に対し，それが属している記述体系を明確にする必要があります．**つまり，異なる記述体系内の記述は区別しなければなりません．**

ある記述体系内の記述の全てを記述する方法を与えることを，その記述体系を**構築する**と呼びます．もちろん，この「方法」も何らかの記述によって与えられます．ある記述体系内の記述が別の記述体系を構築しているとき，前者を**メタ言語** (metalanguage)，後者を**対象言語** (object language) と呼びます．

例 1.1.1
日本語で書かれた Python の仕様書があるとき，日本語は Python を構築している．メタ言語は日本語であり，対象言語は Python である． □

（まだ）いかなる記述体系の対象言語にもなっていない記述体系を**原始言語**と呼びます．ここでは原始言語として日本語を採用します．このノートの平文も日本語で書かれていますが，原始言語による記述と混同しないようにしてください．また，メタ言語であるが原始言語ではないような記述体系を**非原始言語**と呼びます．

注意 1.1
原始言語の性質から，原始言語には，他のいかなる記述体系内の記述によっても構築されないような記述が含まれています．従って，原始言語内のこうした記述に限っては，私たちがその意味内容を自明かつ正しく理解できる=既知である必要があります（実際にはほとんどの記述が既知とされます．）． □

このノートでは，原始言語が満たすべき条件を「メタ規則」の項目で示します．**以降，断りがない限りは，メタ言語は原始言語であるとします．**つまりこの場合，「対象言語」は原始言語から構築される記述体系を指し，「メタ規則」は「メタ言語が満たすべき条件」と一致することになります．

注意 1.2
実際に個々の数学を展開する上では，特定の原始言語しか用いないので，原始言語に関する一般論である「メタ規則」を参照する必要は本来ありません．また，非原始言語をメタ言語として用いるような議論は第 [執筆中] 部などで行われます． □

また，メタ言語内の記述を「規則」の項目で示します．対象言語について言及する記述や，メタ言語内の記述の別記（後述）や記述の使用法に関する規則などが「規則」として提示されます．さらに，「規則」から正しいと分かる事実を「メタ定理」の項目で示し，それがなぜ正しいかについての説明を「メタ証明」の項目で示します．

原始言語内に存在するべき記述を示します．ここでは，通常の日本語では必ずしも既知ではないような記述に絞ってその存在の明記を行うことにします．

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) **メタ変数**（metavariable）と呼ばれる記述 □

メタ変数は，記述を値として取る「変数」のようなものです．こ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・対象言語内の他の記述に**書き換える**」という操作が既知である．
(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をあらかじめ定めなければならない．これを**メタ変数の範囲**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 1.1.4

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

$$\boldsymbol{M} \text{ は小さい}$$

とは，記述

$$!\text{は小さい}, \mathcal{E} \text{ は小さい}, PPP \text{ は小さい}$$

である． □

問題 1.1

例 1.1.4 と同じ条件のもとで，記述

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

はどのような記述でしょうか． □

注意 1.3

,（コンマ）は各々の記述の区切りを表すものであり，主張には含まれません．以降も同様です． □

以降，「（メタ変数）の範囲は（範囲）である」を単に「（メタ変数）は（範囲）である」と書きます．

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．別記の提示は「（別記）とは（元の記述）である」という形式で行うことにします．

規則 1.1.5（⇒，&）
以下の別記を行う．

(1) ⇒ とは「ならば」である．
(2) & とは「かつ」である．

□

注意 1.4
先に述べておくと，⇒ と & は，[執筆中] で登場する → と ∧ とは異なります．⇒, & はメタ言語内の（既知の）記述ですが，→, ∧ は対象言語内の記述です． □

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，それらが議論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．

b. 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，記述の「個数」についての議論など，対象言語についての一般的な議論ができるようになります．

メタ規則 1.1.6（0，|）
以下の記述がメタ言語内に存在する．

(1) **0**
(2) |

□

規則 1.1.7（数字）
数字と呼ばれる記述は以下で決定される．ここでメタ変数 $\boldsymbol{n}$ は数字である．

(1) **0** は数字である．
(2) $\boldsymbol{n}$| は数字である．

□

注意 1.5
メタ規則 1.1.3 に従って規則 1.1.7 の一連の書き換え操作を行うと次のようになります．

(1) **0** は数字である．
(2) **0**| は数字である．
(3) **0**|| は数字である．
(4) **0**||| は数字である．
(5)（以下省略）

□

数字を定めたので，改めて数字を範囲に持つメタ変数を複数導入します．

メタ規則 1.1.8（$\boldsymbol{n}, \boldsymbol{m}, \boldsymbol{i}, \boldsymbol{j}, \boldsymbol{k}$）
メタ変数 $\boldsymbol{n}, \boldsymbol{m}, \boldsymbol{i}, \boldsymbol{j}, \boldsymbol{k}$ およびそれに装飾を加えたものは数字である． □

ここで，「装飾を加える」とは数字や $'$ を添え字として加えることをいいます．装飾を加えたメタ変数は，元のメタ変数とは異なる記述とされます．

注意 1.6
ここでは太字によってその記述がメタ言語内の記述であることを表すことにします．以降，メタ変数は

数字以外のアルファベットで表記します．□

大小や足し算に相当するものなど，数字に関する基本的な記述を定めていきます．

規則 1.1.9（$<$）
記述 $\boldsymbol{nn'}<$ は以下で決定される．

(1) $\boldsymbol{nn}|<$
(2) $\boldsymbol{nn'}<$ & $\boldsymbol{n'n''}<\Rightarrow \boldsymbol{nn''}<$

$\boldsymbol{nn'}<$ を，「$\boldsymbol{n}$ は $\boldsymbol{n'}$ より**小さい**」または「$\boldsymbol{n'}$ は $\boldsymbol{n}$ より**大きい**」とも書く．□

例 1.1.10
$\mathbf{0}|\mathbf{0}|||<$ □

メタ証明
規則 1.1.9 のメタ変数を具体的な数字に書き換えることで示せます．

(1) $\mathbf{0}|$ は数字である．（規則 1.1.7(1),(2)）
(2) $\mathbf{0}|\mathbf{0}||<$ ((1)，規則 1.1.9(1))
(3) $\mathbf{0}||$ は数字である．((1)，規則 1.1.7(2))
(4) $\mathbf{0}||\mathbf{0}|||<$ ((3)，規則 1.1.9(1))
(5) $\mathbf{0}|||$ は数字である．((3)，規則 1.1.7(2))
(6) $\mathbf{0}|\mathbf{0}||<$ & $\mathbf{0}||\mathbf{0}|||<\Rightarrow \mathbf{0}|\mathbf{0}|||<$ ((1)，(3)，(5)，規則 1.1.9(2))
(7) $\mathbf{0}|\mathbf{0}|||<$ ((2)，(4))

■

注意 1.7
以降，ある記述が数字であることの規則 1.1.7 による確認は省略します．□

規則 1.1.11（$+$）
以下の別記を行う．

(1) $\boldsymbol{n}\mathbf{0}+$ とは $\boldsymbol{n}$ である．
(2) $\boldsymbol{nn'}|+$ とは $\boldsymbol{nn'}+|$ である．□

例 1.1.12
$\mathbf{0}||\mathbf{0}|||+$ と $\mathbf{0}|||||$ は同じ記述である．□

メタ証明
規則 1.1.11(2) を繰り返し参照し，最後に規則 1.1.11(1) を参照すれば示せます．

(1) $\mathbf{0}||\mathbf{0}|||+$ とは $\mathbf{0}||\mathbf{0}||+|$ である．（規則 1.1.11(2)）
(2) $\mathbf{0}||\mathbf{0}||+|$ とは $\mathbf{0}||\mathbf{0}|+||$ である．（規則 1.1.11(2)）
(3) $\mathbf{0}||\mathbf{0}|+||$ とは $\mathbf{0}||\mathbf{0}+|||$ である．（規則 1.1.11(2)）
(4) $\mathbf{0}||\mathbf{0}+|||$ とは $\mathbf{0}|||||$ である．（規則 1.1.11(1)）

■

さて，引き算に相当するものも定めたいところですが，ここでは「小さい数字から大きい数字を引くと $\mathbf{0}$ になる」ような形でうまく定めることにします．そのためには「一つ前の数字」を表すことが必要

となります.

規則 1.1.13（←）
以下の別記を行う.

(1) $\mathbf{0}\leftarrow$ とは $\mathbf{0}$ である.
(2) $\boldsymbol{n}|\leftarrow$ とは $\boldsymbol{n}$ である.

$\boldsymbol{n}\leftarrow$ を $\boldsymbol{n}$ の**前者**（predecessor）ともいう. □

規則 1.1.14（−）
以下の別記を行う.

(1) $\boldsymbol{n}\mathbf{0}-$ とは $\boldsymbol{n}$ である.
(2) $\boldsymbol{n}\boldsymbol{n}'|-$ とは $\boldsymbol{n}\boldsymbol{n}'-\leftarrow$ である.

$\boldsymbol{n}\boldsymbol{n}'-$ を $\boldsymbol{n}$ と $\boldsymbol{n}'$ の**非負差**（nonnegative difference）ともいう. □

例 1.1.15

(1) $\mathbf{0}||||\mathbf{0}||-$ と $\mathbf{0}||$ は同じ記述である.
(2) $\mathbf{0}|\mathbf{0}|||-$ と $\mathbf{0}$ は同じ記述である. □

メタ証明

(1) 規則 1.1.14(2) を繰り返し参照した後規則 1.1.14(1) を参照し，その後規則 1.1.13(2) を繰り返し参照すれば示せます.
    (1) $\mathbf{0}||||\mathbf{0}||-$ とは $\mathbf{0}||||\mathbf{0}|-\leftarrow$ である. （規則 1.1.14(2)）
    (2) $\mathbf{0}||||\mathbf{0}|-\leftarrow$ とは $\mathbf{0}||||\mathbf{0}-\leftarrow\leftarrow$ である. （規則 1.1.14(2)）
    (3) $\mathbf{0}||||\mathbf{0}-\leftarrow\leftarrow$ とは $\mathbf{0}||||\leftarrow\leftarrow$ である. （規則 1.1.14(1)）
    (4) $\mathbf{0}||||\leftarrow\leftarrow$ とは $\mathbf{0}|||\leftarrow$ である. （規則 1.1.13(2)）
    (5) $\mathbf{0}|||\leftarrow$ とは $\mathbf{0}||$ である. （規則 1.1.13(2)）
(2) 途中までは (1) と同様で，その後規則 1.1.13(1) を繰り返し参照すれば示せます.
    (1) $\mathbf{0}|\mathbf{0}|||-$ とは $\mathbf{0}|\mathbf{0}||-\leftarrow$ である. （規則 1.1.14(2)）
    (2) $\mathbf{0}|\mathbf{0}||-\leftarrow$ とは $\mathbf{0}|\mathbf{0}|-\leftarrow\leftarrow$ である. （規則 1.1.14(2)）
    (3) $\mathbf{0}|\mathbf{0}|-\leftarrow\leftarrow$ とは $\mathbf{0}|\mathbf{0}-\leftarrow\leftarrow\leftarrow$ である. （規則 1.1.14(2)）
    (4) $\mathbf{0}|\mathbf{0}-\leftarrow\leftarrow\leftarrow$ とは $\mathbf{0}|\leftarrow\leftarrow\leftarrow$ である. （規則 1.1.14(1)）
    (5) $\mathbf{0}|\leftarrow\leftarrow\leftarrow$ とは $\mathbf{0}\leftarrow\leftarrow$ である. （規則 1.1.13(2)）
    (6) $\mathbf{0}\leftarrow\leftarrow$ とは $\mathbf{0}\leftarrow$ である. （規則 1.1.13(1)）
    (7) $\mathbf{0}\leftarrow$ とは $\mathbf{0}$ である. （規則 1.1.13(1)）

■

さて，このままでは数字の使い勝手があまりよくないので，さしあたり，いわゆる「1」から「9」に相当する数字を別記として定めます.

規則 1.1.16（1 から 9）
以下の別記を行う.

(1) $\mathbf{1}$ とは $\mathbf{0}|$ である.
(2) $\mathbf{2}$ とは $\mathbf{1}|$ である.

(3) $\mathbf{3}$ とは $\mathbf{2}|$ である．
(4) $\mathbf{4}$ とは $\mathbf{3}|$ である．
(5) $\mathbf{5}$ とは $\mathbf{4}|$ である．
(6) $\mathbf{6}$ とは $\mathbf{5}|$ である．
(7) $\mathbf{7}$ とは $\mathbf{6}|$ である．
(8) $\mathbf{8}$ とは $\mathbf{7}|$ である．
(9) $\mathbf{9}$ とは $\mathbf{8}|$ である．

□

注意 1.8

(1) 規則 1.1.16 の別記を元の記述に戻すと次のようになります．これらがすべて数字であることは明らかでしょう．
    (1) $\mathbf{1}$ とは $\mathbf{0}|$ である．
    (2) $\mathbf{2}$ とは $\mathbf{0}||$ である．
    (3) $\mathbf{3}$ とは $\mathbf{0}|||$ である．
    (4) $\mathbf{4}$ とは $\mathbf{0}||||$ である．
    (5)（以下省略）
(2) メタ規則 1.1.16 ではこれ以上の別記は行っていません．通常通り，$\mathbf{0},\mathbf{1},\mathbf{2},\mathbf{3},\mathbf{4},\mathbf{5},\mathbf{6},\mathbf{7},\mathbf{8},\mathbf{9}$ の組み合わせとして（特に私たちの親しんでいるのと同じ要領で $\mathbf{10},\mathbf{11},\ldots$ という風に）表したいところですが，安易にそうすることはできません．例えば $\mathbf{10}$ は記述 $\mathbf{0}||||||||||, \mathbf{0}|\mathbf{0}$ のどちらなのかが不明瞭となってしまいます．
(3) こちらも先に述べておくと，数字は [執筆中] で登場する「自然数」とは異なります．「自然数」は対象言語内の記述ですが，数字はメタ言語内の記述です．数字は，もっぱら記述を数える（→ [執筆中]）のに用いられる識別可能な「印」たちに過ぎません．さらに補足すると，自然数はその「全体」を考えることができますが，**「数字の全体」は考えることができません**．確かに規則 1.1.7 によって数字はいくらでも得られるのですが，その全てを得ることができるとは考えないのです．同様に，$\boldsymbol{n}$ を含む記述について私たちが知りうるのは，あくまでも $\boldsymbol{n}$ が $\mathbf{3}$ や $\mathbf{0}|||||||||||||||$ のときのような個別の場合であって，「全ての数字に対して〜〜である」という代わりに「どんな具体的な数字を与えられても〜〜を示す手続きがある」と言わねばなりません．これは，私たちが「記述」という現実的な（従って有限的な）対象へと一切を還元していることからも納得がいく話です．メタ言語における，このような「有限の対象は認める／「無数に」ある対象はそれらを順に与える手続きによって示されるべきであり，しかもその「全体」を考えることはできない」という姿勢は**有限の立場**（finitism）と呼ばれています．

□

以降，誤解の生じない限り，$\boldsymbol{nm<}, \boldsymbol{nm+}, \boldsymbol{nm-}$ をそれぞれ $\boldsymbol{(n<m)}, \boldsymbol{(n+m)}, \boldsymbol{(n-m)}$ と書き，さらに誤解が生じない限り，$(,)$ を適宜省略します．

問題 1.2
次を示してください．

(1) $\mathbf{3}<\mathbf{6}$
(2) $\boldsymbol{n}+\mathbf{1}$ と $\boldsymbol{n}|$ は同じ記述である．
(3) $\boldsymbol{(n_1+n_2)-n_2}$ と $\boldsymbol{n_1}$ は同じ記述である．

□

数字について一通り見たところで，次は「個数を数える」という操作を明確にしていきましょう．

メタ規則 1.1.17（割り振る）
メタ言語・対象言語内のある記述を，メタ言語内のある記述に**割り振る**という操作が既知である．　□

メタ規則 1.1.18（空列）
以下の記述がメタ言語内に存在する．

(1) $\boldsymbol{\lambda}$

$\boldsymbol{\lambda}$ を**空列**（empty string）ともいう．　□

規則 1.1.19（数える，個数）
メタ言語・対象言語内のある記述たちを**数える**とは，その各々に対する次の操作である．

(1) 始めに，その記述たちに $\boldsymbol{\lambda}$ を追加し，空列を $\mathbf{0}$ に割り振る．
(2) ある記述を $\boldsymbol{n}$ に割り振った直後，まだ数字に割り振っていない記述が存在するならば，その記述を $\boldsymbol{n}|$ に割り振る．
(3) ある記述を $\boldsymbol{n}$ に割り振った直後，全ての記述を数字に割り振っているならば，$\boldsymbol{n}$ を（$\boldsymbol{\lambda}$ を除く）記述たちの**個数**と呼び，操作を終了する．　□

注意 1.9
規則 1.1.19 によれば，そもそも記述がない場合，その個数は $\mathbf{0}$ となります．また空列とは，強いて言えば「見かけ上何も書かれていない状態と区別できない記述」と思うことができますが，ここではあまり関係ありません．　□

例 1.1.20
記述たち a, b, c, ab, ac, bc の個数は $\mathbf{6}$ である．　□

メタ証明
規則 1.1.19 に従えばよいです．まず $\boldsymbol{\lambda}$ を $\mathbf{0}$ に割り振り，次に a を $\mathbf{1}$ に割り振ります．このとき，例えば記述 b はまだ数字に割り振られていないので，これを $\mathbf{1}|$ すなわち $\mathbf{2}$ に割り振ります．同様に，c を $\mathbf{3}$ に，ab を $\mathbf{4}$ に，ac を $\mathbf{5}$ に，bc を $\mathbf{6}$ に割り振ります．もはや数字に割り振られていない記述が存在しないのでここで操作が終了し，個数は $\mathbf{6}$ となります．　■

「無数にある」ということも定めます．

メタ規則 1.1.21（無数にある）
メタ言語・対象言語内のある記述たちが**無数にある**（infinitely exist）とは，それらを数えるとき，いかなる数字 $\boldsymbol{n}$ に対しても，ある記述を $\boldsymbol{n}$ に割り振った直後，まだ数字に割り振られていない記述を必ず挙げられることである．　□

メタ定理 1.1.22（数字の無数性）
数字は無数にある．　□

メタ証明

(1) $\mathbf{0}$ に $\boldsymbol{\lambda}$ を割り振ります．
(2) $\mathbf{1}$ に $\mathbf{1}$ を割り振ります．この直後，数字に割り振られた記述は空列と $\mathbf{1}$ だけであり，$\mathbf{0}$ はまだ数字に割り振られていません．
(3) $\boldsymbol{n}$ に $\boldsymbol{n}$ を割り振った直後に，$\mathbf{0}$ がまだ数字に割り振られていないとします．このとき，$\boldsymbol{n}+\mathbf{1}$ に

$\boldsymbol{n+1}$ を割り振った直後にも $\mathbf{0}$ はまだ数字に割り振られていません．

(4) 数字は無数にあります．（(2)，(3)，メタ規則 1.1.21）

■

注意 1.10

上のメタ証明は「数字に関する**帰納法**（induction）」と呼ばれる部類のものです．この場合は，まず「（$\mathbf{1}$ に関する記述）」を示し，次に「（$\boldsymbol{n}$ に関する記述）$\Rightarrow$（$\boldsymbol{n+1}$ に関する同様の記述）」を示すことでメタ定理を証明しています．実際，$\boldsymbol{n}$ を具体的な数字に書き換えることで，

（$\mathbf{1}$ に関する記述）$\Rightarrow$（$\mathbf{2}$ に関する記述）
（$\mathbf{2}$ に関する記述）$\Rightarrow$（$\mathbf{3}$ に関する記述）
（$\mathbf{3}$ に関する記述）$\Rightarrow$（$\mathbf{4}$ に関する記述）
$\cdots \Rightarrow \cdots$

といった記述を得ることができ，これらと「（$\mathbf{1}$ に関する記述）」から，どのような $\boldsymbol{n}$ に対しても「（$\boldsymbol{n}$ に関する記述）」が成り立つことを示す手続きが得られます（私たちが有限の立場に立っていることに注意しましょう）．この手法にはバリエーションがありますが，以降同様の補足は省略します．　□

第 1 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