

易しくない 算術のお話

科学書刊株式会社:電子版の原稿

「橋梁 & 都市 PROJECT: 2015」(ISSN 1344-7084)

島田 静雄

まえがき

この冊子は、雑誌「橋梁と都市 PROJECT」に連載することを予定して作成した MS-Word 版の原稿から、PDF 形式に変換したもののコピーです。2010 年度から、出版関係は電子出版を模索した努力をする時代に入りました。科学書刊株式会社雑誌は、この動きに対応するため、ハードコピーとしての「橋梁&都市 PROJECT」の発行を休刊とし、電子化にどのように対応するかの研究を始めました。実を言うと、この傾向は 10 年前から予測されていました。筆者は、この先取りとして、三種類の発表形式を試してきました

一つ目は、雑誌の記事としての形式です。「橋梁と都市 PROJECT」のスタイルは B5 版二段組みです。こちらの方は、しばらく休刊になりました。この原稿は MS-Word で A4 版一段組みで作成してきましたが、そのまま体裁のよいレポート形式になるように注意して編集してあります。この形式のままにしたのが、この PDF 版です。

二つ目、この PDF 版をインターネットで公開することです。すこしページ数が多くなりますが、ユーザは、これをダウンロードして印刷して見ることができます。プリンタをお持ちでなければ、原稿ファイルを USB にして持ち込めば、簡易製本までサービスしてくれる街中の印刷屋さんが見つかりようになりました。PDF 版の WEB サイトは、差し当たり下記にしております。

<http://www.nakanihon.co.jp/gijyutsu/Shimada/shimadatop.html>

三つ目は、パソコンの画面でランダムに項目がアクセスするようにリンクを張った WEB 版です。この利用方法を考えて、筆者の原稿は、約 600 字程度のパラグラフ単位分けてあって、インターネットでのアクセス速度が速くなるように、一つのパラグラフがパソコンの一画面に入るようにしてあります。目次と索引とを参照すれば、かなり便利な検索が使えます。WEB サイトは上の PDF 版と同じ個所です。この冊子は、表紙と目次を含めて全 25 ページあります。電子出版を考えると、ページ番号で項目位置を探ることが実用的ではありません。したがって、目次と索引は、章・節・項のパラグラフ番号で検索するように使って下さい。

この冊子は、全部で 24 ページです

目 次

まえがき

0. はじめに

1. 数を言葉で言う

1.1 数とは何か？

- 1.1.1 哲学的な考え方は後回しにする
- 1.1.2 個数のまとまりに単位を付ける
- 1.1.3 アナログ量とデジタル量

1.2 道具を使って数を説明する

- 1.2.1 感覚的には20までの数で対話する
- 1.2.2 数量を確認するときに道具を使う
- 1.2.3 集合名詞の量の数え方は特殊になる
- 1.2.4 数の言い方と書き順に混乱がある

1.3 数のゼロとマイナスとを理解しておく

- 1.3.1 何も無い時の個数を言いたいとき
- 1.3.2 順番を言うときの数が順序数
- 1.3.3 順序数を一意に決められないことがある
- 1.3.4 満と「かぞえ」の使い方がある

1.4 数の表示と単位分けとを学ぶ

- 1.4.1 技術には三つの要素を考える
- 1.4.2 数を別の文字に書いて表す方法
- 1.4.3 アラビア数字を文字とは言わない
- 1.4.4 右書きと左書きと読み

2. 四則演算の常識を確認する

2.1 算術の基礎教育

- 2.1.1 一桁の足し算の教育から始まる
- 2.1.2 レジスタの概念を算盤で理解する
- 2.1.3 補数の概念が使われている
- 2.1.4 掛け算の基礎に九九を覚える
- 2.1.5 掛け算は、九九と足し算を応用する
- 2.1.6 整数を扱う割り算は二種類ある
- 2.1.7 余りを扱う割り算は算術に属する

2.2 引き算と割り算には向きがある

- 2.2.1 引き算は交換律が成立しない
- 2.2.2 座標の考えを使う
- 2.2.3 マイナスの寸法は無い
- 2.2.4 ゼロ(0)で割る割り算を定義しない。

3. 数を表す方法

3.1 意味を持たせて数を書く

- 3.1.1 有限桁数の数字並びを決める
- 3.1.2 数の大小を区別する約束
- 3.1.3 有効数字と有効桁数を理解する
- 3.1.4 数を文字として使うこともある

3.2 固定小数点と浮動小数点

- 3.2.1 実数の実体と表示法の選択
- 3.2.2 電卓は固定小数点表示を使っている
- 3.2.3 パソコン付録の電卓は浮動小数点法を使っている
- 3.2.4 普通の電卓の賢い使い方

3.3 計算の組み立てを文に書く

- 3.3.1 数式の書き方
- 3.3.2 計算手順の説明に使う文
- 3.3.3 代数式は文法規則と関連がある

3.4 丸めの方法と応用

- 3.4.1 数字並びの前にある0を考えない
- 3.4.2 用語の数学的意義を理解しておく
- 3.4.3 お金の計算で使う丸め
- 3.4.4 実数を整数化する丸め
- 3.4.5 覚え易くするために呼び数を使う
- 3.4.6 大小の寸法系列を合理的に決めたい
- 3.4.7 お金の小数単位にセントがよく使われる

4. 論理演算の約束を覚える

4.1 コンピュータでは論理演算が使える

- 4.1.1 数値計算に応用する2値論理学
- 4.1.2 論理演算の用語と記号に混乱がある
- 4.1.3 論理用語と論理記号
- 4.1.4 集合論の考え方も使う

4.2 論理演算の基礎知識

- 4.2.1 初等算術の教育課程にならう覚え方
- 4.2.2 論理演算の具体的な演習例題
- 4.2.3 数式を命題とする考え方
- 4.2.4 論理演算を幾何モデリングに応用

5. 演算の約束も言葉で表す

5.1 数式文字並びはグラフィックスである

- 5.1.1 数式の編集は特殊技術である
- 5.1.2 数式を文章に書いて説明する
- 5.1.3 数式の組版は特殊な技能である
- 5.1.4 計算手順の説明は別に組み立てる

5.2 数式を読み上げて相手に伝える

- 5.2.1 数式は物理的な意義も考える
- 5.2.2 数式を文章に表す
- 5.2.3 算盤の読み上げ算

5.3 歴史の古い幾何と代数

- 5.3.1 パソコンは幾何の計算に向いていない

99. おわりに

索引

英字		幾何	0	順序数	1.3.1
AND	4.1.3	幾何モデリング	4.2.4	初等幾何学	0
basement	1.3.4	帰属	4.1.2	除数	2.1.6
Cent	1.4.2	記号番号	1.3.1	商	2.1.7
Character	1.4.3	記号論理学	4.1.1	乗数	2.1.5
D. Knuth	5.1.2	技術	1.1.1	剰余	2.1.6
Digit	1.2.1	逆順	3.3.3	スラッシュ	2.2.4
Digit、	1.4.3	金銭登録機	1.2.2	推論	4.1.1
divide by	2.1.7	切り捨て	3.4.1	数学	0
divide into	2.1.7	切り上げ	3.4.1	数学	1.4.1
ID	1.3.1	グロス	1.1.2	数学記号	5.1.3
IMP	4.1.3	九九	2.1.4	数式	3.3.1
INT	3.4.4	繰り上がり	2.1.1	数式エディタ	5.1.2
Letter	1.4.3	桁上がり	2.1.1	数値解析	1.4.1
MOD	2.1.7	結論	4.1.1	数値計算法	1.4.1
NOT	4.1.3	言語学	4.1.1	数直線	3.1.2
OR	4.1.3	言文一致	0	整数	1.2.1
Symbol	1.4.3	コロン	2.2.4	セント	3.4.7
Tex	5.1.2	固定小数点法	3.2.1	宣言	5.2.1
Three R's	0	御破算	3.2.2	選言	4.1.2
XOR	4.1.3	交換法則	2.1.5	前提	4.1.1
		交換律	2.1.5	全称記号	4.1.2
		勾配	2.2.4	ソフトウェア	1.4.1
				算盤	1.2.2
				存在記号	4.1.2
		サ～ソ		タ～ト	
ア～オ		座標幾何学	3.1.2	ダース	1.1.2
アナログ	1.1.3	座標変換	3.1.2	対等	4.1.2
アラビア数字	1.2.1	左辺値	3.3.3	代数	0
アンダーフロー	3.2.2	三角関数	5.3.1	代入文	3.3.1
余り	2.1.6	産業革命	0	単位	5.2.1
以下	3.4.2	算術	0	手回し計算機	2.1.2
以上	3.4.2	算数	0	定位点	2.1.2
演算子	3.3.1	算用数字	1.2.1	定義	5.2.1
右辺値	3.3.3	シフト	2.1.5	電卓	1.2.2
オーバーフロー	3.2.2	シフトレジスタ	2.1.5	電動計算機	2.1.2
オブジェクト指向プログラミング	3.2.3	四捨五入	3.4.1	デジタル	1.1.3
		四則演算	2.1.1	等式	3.3.1
お習字	0	次元	5.2.1	同値	4.1.2
		自然数	1.2.1	道具、技法、技能	1.4.1
カ～コ		実学	1.1.1		
可逆の内含、	4.1.2	実数	1.4.2	ナ～ノ	
可算名詞	1.1.2	実数	3.1.1	内含	4.1.2
科学	1.4.2	集合	1.1.2	二値論理学	4.1.1
寒暖計	2.2.2	集合論	4.1.4		
含意	4.1.2	循環小数	3.1.1		
数え唄	2.1.4				
数え年	1.3.4				
キロ・メガ・テラ	… 1.4.3				

ハ～ホ		浮動小数点法	3.2.1	有効桁数	3.1.3
パーセント	1.4.2	復唱	2.1.4	有効数字	3.1.3
ハードウェア	1.4.1	物質名詞	1.1.3	有理数	3.1.1
バイト	1.1.2	分数	1.4.2	呼び数	3.4.5
排他的選言、	4.1.2	文	3.3.1	読み・書き・そろばん	0
排反	4.1.2	ベン図	4.1.4	読み上げ算	1.3.1
話す	0	補数	2.1.2		
ビット、	1.1.2	包摂	4.1.2	ラ～ロ	
ヒューマンインタフェース	1.4.1	翻訳	3.3.3	レジスタ	2.1.2
				レドシヨツセ	1.3.4
		マ～モ		連言	4.1.2
引き算	2.2.1	丸め	3.4.1	連想記憶	2.1.4
否定	4.1.2	未満	3.4.2	論理学	4.1.1
被除数	2.1.6	無理数	3.1.1	論理積	4.1.2
被乗数	2.1.5	命題	4.1.1	論理和	4.1.2
非両立	4.1.2	明治維新	0		
百分率	1.4.2	文字	1.4.3	ワ～	
標準数	3.4.6			ワード	1.1.2
表音文字	1.4.3	ヤユヨ		割り算	2.2.4
評価	3.3.1	ヤード	1.1.3	を超え	3.4.2
フィート	1.1.3	優先順位	3.3.2		
ブール代数	4.1.1				

0. はじめに

日本は、**明治維新**(1868)以降、欧米の学問(科学; science)と技術(technology)に学んで、急速に近代化に成功しました。**学ぶ**とは、**まねぶ**、が転化した言葉です。真似ができるには、下地の教養が必要です。慣用句に「**読み・書き・そろばん**」があります。庶民が社会生活をする上で、最低必要な実用技能の種類を言います。ここで、**そろばん**(算盤)とは、算術技能の比喩的な言い換えです。英語にも同じような言い方があって; Three R's で辞書に載っています。Reading, writing, arithmetic の3語の組みを言い、子供に教育する基礎的な学科のことです。これは、Rの音が3語にあることから、慣用される言い方です。「**読み・書き**」の二つを取り上げるとき、古くは、「漢学のような学問」を修めることとされてきました。ただし、「**書き**」の技能では、**お習字**に、漢字を覚えさせる意義があり、想像以上に教育効果があります。漢字を眼で見て読めても、書けないことも多いのです。英単語を覚えるときも、文字を書く演習をすると、正しいスペルを覚えることに役に立ちます。そろばんの使い方は、学問としての数学よりも、実用技術として覚えることが、前の二つ「**読み・書き**」と異なります。

ここで、不思議なことがあります。それは「**読み・書き**」に加えて、「**話す**」が抜けていることです。現代は、電話、それもテレビ付き通信も実用できるようになりました。しかし、以前は、少し遠くに居る人と、顔を合わせて、相手との対話で情報を交換することが不便でした。文字を介して間接的に伝えなければならなかったのです。その「**書きもの**」は、声に出して読み上げるときの言い方とは、かなりの違いがありました。明治維新以降、日本語は、英語を始めとした外国語の影響を大きく受けるようになりました。言文一致の運動は、今日の口語文を普及させる原動力になりました。敗戦(1945)後、英語の勉強でも、声に出してコミュニケーションをする英会話、つまり、話し方が重要な技能として認められるようになったことを理解しておく必要があります。

イギリスで18世紀の半ばから始まった**産業革命**は、近代的な学問の発展と、その応用である実用技術の進歩に支えられて、近代社会に多くの貢献をもたらしました。とりわけ、学問としての数学とその応用とが、非常に発展しました。小学校での数学教育の科目名は、以前は**算術**と言いましたが、少しハイカラな語感の**算数**に変わりました。中身は四則演算の基礎技能を教えます。中学・高校では学問教育を表に出した教科名の**数学**です。そもそも、数学は、実用的な数値計算に応用することに意義があるのですが、数値計算の演習にあまり時間を掛けないようになっていきます。コンピュータを使って数値計算をさせるプログラミングは、難しい表現で提案された数式を、コンピュータの処理に向くように、加減乗除の組み合わせに変換する技術です。プログラミングは、数値計算の手順を、人と同時にコンピュータも理解できるような言語を使って書きます。ここに、「**読み・書き**」そして、近年では「**話し**」かけて利用することも使われるようになってきました。

数学が扱う課題を世界史的に見ると、**代数**(algebra)と**幾何**(geometry)の歴史は非常に古く、アラビア数学がギリシャ時代に大きく発展したものです。代数は、**数の代り**に文字や記号を使って、数の性質や計算法を研究する学問の意味です。幾何は、英語の綴りから判るように、**測量学**の意義があります。筆者の中学時代までは、代数と幾何とは別の先生が担当する学科でした。ギリシャ時代の幾何学は、数式を使わず、論理的な文章で定理などの説明をします。これを**初等幾何学**として習いました。測量学や天文学は、初等幾何学を応用しています。現代の数学は、専門範囲が広がっています。幾何にも代数学や解析学を応用するようになって、幾何も数学の一分野を占めるようになりました。

数学は、好き嫌いの差が大きい教育科目です。その影響があって、個人の教養の適性を、理科系か文科系か、に大きく分けることもします。しかし、幾らか的が外れた世間の常識である、と筆者は思っています。数学の理解にも、文科的な要素としての「**読み・書き・話し**」の三つの教養が必要です。ここにまとめた「**易しくない算術のお話**」は、この三つの要素を前面に出すようにして、算術と数学の扱う課題を説明しました。数学の歴史が古いこともあって、初等教育である算術が、保守的に扱われていることに、気が付かなくなっています。数学も、「**読み・書き・話す**」技術が大切であることを説明したいため、この報文をまとめました。

1. 数を言葉で言う

1.1 数とは何か？

1.1.1 哲学的な考え方は後回しにする

学問的な立場で数の性質を議論することは、学者の好むところです。数学として学の手を付けると、何か高級な印象を受けます。しかし、現実社会で、数がどのような場面で使われているか、その実態を理解することも大切です。数が扱われている場面は多様です。科学的な方法は、個別の多様性を捨象し、抽象的な性質を残すような、哲学的態度をとります。しかし、科学的な知識を、実用面に応用しようとするのは、価値のある態度です。こちらは、やや軽蔑的に実学と言うこともあります。科学と対応させて技術の用語を使うようになりました。算術は、まさに、技術です。

1.1.2 個数のまとまりに単位を付ける

物を数(かぞ)えるとき、ごく自然に順序をつけて数を言います。最後に言う順番の数が、個数です。個数の多少を区別するときの補助に、数の大小で理解しています。個数の多い全体を言うとき、量(quality)の用語を当てます。英語では、一つ二つと数えられるものを可算名詞とし、複数ある集合をまとめて言うときに複数形にします。個数が多くなって、数え難くなる時、何かの入れ物を補助にして、その入れ物を単位にして数えます。

我々が日常使っている数は、1から9までが数の区別をする基本です。日本語で言う十、百、千…(10、100、1000…)は、個数のまとまりに付ける単位の名前です。これは10進数の数え方です。しかし、6、12、60を数の区切り単位に使うこともあります。時間を区切る単位の「時・分・秒」は、24,60,60 の数単位で区切ります。欧米起源のダースは12個を単位とする言い方です。さらに、12ダース(144個)をグロスとします。これより多くなるときの単位名はありません。欧米の数の数え方は、1から9までと、1から19までを区切る言い方が混じります。桁数の多い数を言うとき、日本では1から9999までを循環的に言い、これより大きい数を4桁ごとに区切り、万・兆・京…の単位名を使います。欧米流では、1から999までの3桁ごとに区切ります。

コンピュータを使うようになって、2進数、8進数、16進数も使うようになりました。コンピュータの記憶領域の寸法を言う数の単位に、ビット、バイトの用語を使います。その上の単位にワードを決めることもあります。1ワードを何バイトで区切るかはコンピュータのメーカーによって違いがありましたので一般に利用できる標準化には進みませんでした。

1.1.3 アナログ量とデジタル量

上の項の説明では、量が多くても、物を個数として区別できる顔があるものと、それができない場合があることの例を説明しました。個数として数えることができる時、この数え方がデジタル的であると言うようになりました。しかし、数えられないほど多くなる集合(例えば穀物)、さらには、水のような液体は、英語では物質名詞とし、それらの入れもの工夫して、それを単位として量を数に換えて表します。英語で言う「a cup of water」がそうです。長さ、面積、体積、重さなどの物理的な性質の大小を区別するときも、物質名詞に準じた数え方をします。このとき、単位を決める道具が物差しと秤です。こちらは人為的な決め方です。曖昧さを持ちますので、アナログ的であると言うことがあります。何か絶対的なもの、または不変的なものを基準にして単位を決めたい、それが科学的な態度である、と言われるようになりました。メートル法を提案した根拠は、地球の子午線の長さを正確に測って、極から赤道までの長さを一万キロと決めたのでした。

長さを測るときの単位も複雑です。英米のフィートポンド法では、1フィートは12インチです。ゴルフの距離を言うときのヤードは、3フィートです。日本で使っていた尺貫法では、長さの単位に、「寸・尺・間・丈・町・里」を使い分けました。これらを使うことは非科学的だと軽蔑するのは間違いです。むしろ、歴史的な経緯があって日常生活と密接につながっていることを理解しておくことが大切です。一時期、日本のゴルフ場で、ホールの長さにヤード表示をやめて、メートル法で言うことをしましたが、評判が悪く、使わなくなりました。正義感を持って、科学的な方法を提案することは、迷惑なこともあるのです。

1.2 道具を使って数を説明する

1.2.1 感覚的には20までの数で対話する

整数の、使い方による区別には、物の個数を数えるときと、順序を区別するときとがあって、これに使う数が**自然数**です。個数の多少を正しく区別できるのは、人が高度の知能を持っているからです。他の動物に、同じような知能があるかどうかは、よく分からないところがあります。人間の子供が赤ん坊から成長していく過程で、数の大小区別があやふやで、「一つ・二つ・いっぱい・いっぱい…」と言うことがあります。社会生活で、物の個数を数えるときの日常的な行動は、指を使います。最も原初的な商業は、物々交換です。互いに言葉が分からない場合でも、個数を確認する方法に指を使います。英語の**Digit**は指のことですが、**アラビア数字**(算用数字)の1から9までの数も言います。両手を使うと、10までの数量は、簡単に数えることができます。相手の指の数も利用すれば、20までの個数の品物の物々交換を簡単に進めることができます。

1.2.2 数量を確認するときに道具を使う

10個以下の少ない数量を互いに確認し合うときは、指で数えます。指も立派な道具だからです。しかし、やや大きな数を商活動の場で確認したいとき、日本では**算盤**(そろばん)がよく使われました。現代では、**電卓**(卓上電子計算機)が普及しました。英語の abacus は、そろばんの訳に当てますが、英米では、子供に数を教える道具です。これらの道具は、その場限りで数を表示して確認する一過性の使い方です。つまり、証拠になる記録が残りません。したがって、数を確認し合うときは、口約束でも、お互いに相手を信頼することが基本です。念のためと、覚えのために「書きもの」を残します。記録が残せる計算道具の代表が**金銭登録機**です。これは、単純な計算機に簡単なプリンタを付けたものです。筆者は、コンピュータが未だ普及していない時代、技術計算に手回しの計算機を多用しました。このとき、途中の計算結果を紙に書いて控えを残し、計算間違いの検査をすることに苦労がありました。そこで、記録の取れる計算機として、金銭登録機を利用することも考えました。しかし、加減算には向きますが、乗除計算にはレジスタの桁数が足りないことが難点でした。

1.2.3 集合名詞の量の数え方は特殊になる

数えられる物でも、個数が多くなると、数の確認があやふやになります。そこで、何かの入れ物を考えるか工夫して、補助に使います。**ダース**は12と同じで、助数詞として使います。物を箱に詰めるとき、3行4列または2行3列にすると収まりが良くなります。日本の習慣では、品物は5個または10個単位でまとめます。欧米人の感覚では、半ダースまたは1ダースに満たない欠陥品と誤解されることがあります。20個をまとめるとき、4行5列の容器がよく使われています。英語を代表として、欧米語を習うと、単数・複数の区別がある普通名詞と、集合名詞と物質名詞の量の表し方を覚え、さらに動詞の使い方に注意しなければなりません。入れ物と単位とを決めて、量を数で理解します。例えば、英語を習い始めると、a cup of water の言い方を教わります。これは、日本語の習慣に無いと思うでしょう。しかし、加算名詞であっても、人は「人(ニン)」、馬や牛は「頭」、虫は「匹」、たたみは「畳」、などの助数詞を付けます。これは単位名の性格があります。外国人が日本語を習うとき、この習慣を覚えるのが厄介ですが、物質名詞と同じ扱い、と割り切っている人もいます。長さ・面積・重さなどは、物質名詞に準じた数え方をします。例えば、農地の大きさは「町、反、畝、坪」などの助数詞を使い分け、小数表記を使いません。このときの計量道具が、物差しと秤です。

1.2.4 数の言い方と書き順に混乱がある

英語を代表として、欧米語では、20までは個別の言い方があります。注意することは、13から19までは、英語は「…teen」と言います。これは、1の位の数を先に、10を後に言う語順ですので、アラビア数字で書く文字順とは逆です。20以上の二桁の数、例えば、53では、「50、3」の順で言います。ところが、ドイツ語では、接続詞を挟んで、逆順に「3と50」の言い方をします。この習慣は、右からの横書きであるアラビア語の習慣に影響されたのではないかと筆者は想像しています。もう一つ、逆順の言い方を挙げます。お金の額を英語風の横向きに書くとき、例えば、「\$34」「¥1500」のように通貨記号を頭に書きます。しかし、それを読むとき、記号の読みを数字の後に言います。数を書いて示す場合、アラビア数字を使うときと、読みをアルファベット文字表記にする区別があります。例えば、1965とアラビア数字で書いてあるとき、数学的に、千九百六十五の意味に取ります。千単位に区切って読む数字であることを区別したいとき、区切り記号を入れて、1,965 のように書くこともします。年号として読むときは、二桁単位で言い、nineteen sixty-five と発声し、またそのようにも書きます。英作文では、文頭を数字で始めない約束です。文頭に数字を書く必要があるときは、必ず文字表記にします。これを数の spell out と言います。漢数字を使う日本語表記では、語順の混乱が起りません。

1.3 数のゼロとマイナスとを理解しておく

1.3.1 何も無い時の個数を言いたいとき

物の個数を数えるときは、和語的には一つ・二つ…のように言います。助数詞を付けて「一個・二個…」、切り紙ならば「一枚・二枚…」、本ならば「一冊・二冊…」のように言います。何も無いときに、ゼロを使う言い方、例えば0個、0枚、0冊などとは言いません。商品などでは、品切れ・売り切れなどと言います。日本語では、数の漢字表記に漢数字(一・二・三・四…十・百・千…)を使います。ゼロには漢字の零を当てることもありますが、普通は何も書きません。漢数字で使う〇は、記号であって、漢字の辞書に載りません。アラビア数字の表記で、0を含む 103 または 2005 を漢数字で書くとき、百三または二千五で済ませます。そろばんの読み上げ算では、「百トんで三、二千トビトビの五」のように言うことで、0の数が幾つ続くかを知らせる方法を使います。英語でも、物の個数を言うときは、数形容詞を名詞の前に付けます。何も無いときは、nothing、no sheet、no book のように、無に相当するnoを数詞の代りに使います。英語ではアラビア数字をdigitに分類するのですが、0を含めないこともあります。つまり、何も無いことに、数字の0を当てるようになったのは、数学史の中でも興味のある話題です。英語では、数字並びを逐語的に言うときは、0{ゼロ}を「オー」と発声します。

1.3.2 順番を言うときの数が順序数

個数を言うときの数と、順番を言うときの数とは、性格が異なっています。個数を表す数は、四則演算に使うことができますが、順序数はそれができません。ただし、順番を変えることはしますが、順序の連続性が壊れないようにします。日本語では、順番は「第一・第二…」のように書きます。英語では「first, second, third」と使い、4以上は「forth, fifth, …」のように接尾辞 th を付けます。20以上、例えば31番目は thirty first とし、一桁目の順序数の表し方を踏襲します。順序数は1から始めます。学校でクラス編成をするとき、クラスごとに生徒に一意の一連番号を割り振ります。これは、1から始める順序数です。0番はありません。同学年に複数のクラスがあるとき、全部の生徒に一意の番号を当てることをしますが、識別用の記号番号(ID; identification number)の使い方です。英字なども混ぜることもします。道案内をするとき、「三番目の信号交差点を右折する」などの言い方をしますが、現在いる位置を暗黙の0として数える順序数で言います。

1.3.3 順序数を一意に決められないことがある

或る範囲の長さや領域を区別したいとき、擬似的な順序数を使います。擬似的というのは、そこで使う整数が上で説明した生徒番号のように一意に付けられないことと、幾何の座標値のような、実数が持つ精度の意義も持たないからです。住所を言うとき、町名を使って「何々一丁目・二丁目…」のように順序数を当てます。より詳しい位置を言いたいときは、下位に細分化した領域を考え、番地番号などを使います。集合住宅では、建物番号、階番号などを形容詞的に加えて、順序数の室番号を付けます。鉄道駅のホームには、上り線・下り線などの線別を区別するとき、順序数で「1番線・2番線…」のように区別します。しかし、線数が1番線の反対側に増設されることもあって、0番線、00番線と付けることがあります。番地名に、例外的に0番地や無番地の言い方を使うことがあります。ただし、負の番号を当てることはありません。人名では「一郎・二郎・三郎…」のように順序数も見られますが、連続した使い方ではないことも少なくありません。ホテルで部屋番号を割り振るとき、4や13を忌み数として、その番号を飛ばすことがあります。実生活の場では、数学的な正確性にはこだわらない使い方をすることも理解しておきます。

1.3.4 満と「かぞえ」の使い方がある

人の年齢を言うとき、生まれた年を1歳とし、正月になると1歳を加える数え方を数え年と言います。12月生まれは、すぐに2歳になります。1月生まれは、まるまる一年経ってから2歳になります。年齢に使う数は順序数ですが、或る幅を持った範囲を一つの数で表しています。そのため、同い年であっても、一年の成長差がありません。満年齢で言うときは、生まれた月日からの経過年齢を、(満)2歳3カ月のように言います。数学的な正確さで言いたいときは、実数を使って2.5歳のように言うことがあります。

一方、階数の多いビルでは、階に順序数を割り付けます。日本では、地表と同じ高さの階を1階とし、そこから上に二階・三階…と順序数を当てる「かぞえ」方式です。ところが、フランスの建物では、日本の一階をレドシヨッセ(道路と同じ高さの階)と言い、二階から上の階番号を1からの順序数で数えます。数学的には、0から始める数え方ですので、エレベータでは日本の一階に当たる階に0を当て、地下の階にマイナス符号を付けた数を当てることもあります。英米では地下室をbasementと言います。これを受けて、日本では地下の階番号は、下向きにB1・B2…と当てます。0階の番号は使いません。

1.4 数の表示と単位分けとを学ぶ

1.4.1 技術には三つの要素を考える

数学(mathematics)は、数を扱う**学問**(science; **科学**)の総称です。その中身を区別する専門分野は、代数・幾何・微積分…などに分けて学習されます。しかし現実には、数値計算に応用することで、始めて身近の問題の解決に使うことができます。こちらは**算術**(arithmetic)、つまり計算の**技術**です。算術と言うと、幾らか初歩的な意味合いに取られることを嫌って、学術的な言い方は、**数値計算法**、**数値解析**などと言います。単に数学と言うと、実用または応用を捨象して、原理の探求に焦点を絞ります。数値計算法は、世俗的な問題に数学原理を応用して計算する方法を指します。このため、大学での学科分類では、数学科(mathematical science)は理学部に属し、数学の**応用**を扱う学科、例えば応用数学科(applied mathematic)は、工学部に属する、などの仕分けをしています。そこで、技術を構成する全体に、三つの要素を考えます。「**道具**、**技法**、**技能**」です。コンピュータに関連する用語で言うと、「ハードウェア、ソフトウェア、ヒューマンインタフェースに当たります。指を使って計算するときは、指が道具です。そろばんは、立派な道具です。計算を手際よく進めるための知識が技法であって、ここに数学も応用されます。道具としてのそろばんを使うには、練習が必要です。これは人(ヒューマン)によって得手・不得手があります。これが、技能であって、ヒューマンインタフェースの用語の背景です。

1.4.2 数を別の文字に書いて表す方法

科学というときに使う漢字の「科」は、物を分類して分けする意義があります。学問としての数学では、数の性質を何種類かに分類します。実用する数は、**有理数**だけ、それも整数と、整数を成分とする**分数**で実数を表し、そして、ゼロと負の概念を加えます。小数点を含む数字並びの**実数**は、小数点の位置を便宜的に決めた有限の桁並びを使い、実質的には、整数扱いをします。小数以下の数を表記したいとき、日本語では単位「割・分・厘・毛…」を使い分けます。例えば、1を基準として 0.35 の比率を言うとき、「三割五分」のように言います。分数を最も普通に使う方法に**百分率**(%)があります。上の 0.35 は 35%としますが、これは 100 を分母にした分数比率の分子を言い、小数点を使わない整数表示にする方法です。記号の「%」は数字の後ろに書き、その順で声に出します。これは、ドル記号などを頭につける使い方とは異なり、単位系を追加する使い方です。百分率を英語で percent (**パーセント**)と言います。cent はラテン系の言語で百のことです。パーセントは、普通、二桁までの整数で表記し、小数を含む実数表記、例えば 35.4%のような使い方を、できれば避けます。

1.4.3 アラビア数字を文字とは言わない

日本語では、「漢字・かな・数字」、この全体をまとめて**文字**と言います。欧米語は、書き言葉に**表音文字**のアルファベットを使いますので、表音文字を **Letter**と言います。数字は **Digit**、コンマなどは記号 **Symbol**と区別し、その全体を **Character**とくります。英語の環境から見たとき、日本語のカナは表音文字とされ、Kana Letterと説明し、漢字はKanji Characterと区別します。日本語の文字表記は、左からの横書きと右からの縦書きも使います。数字には漢数字がありますが、これは縦書きにも横書きにも対応できます。敗戦前までは、書画の表題、絵葉書の説明などに、右からの横書きが使われていました。この表記方法は、アラビア数字や、英語を混ぜて使うときに不便ですので、現在では使わなくなりました。数の書き方と読み方も、言語種類ごとに習慣の違いがあります。日本語では、桁数の多い大きな数を言うとき、「一・十・百・千」までの4桁の表記で区切り、より大きな単位に、4 桁ごとに「万・億・兆、京…」を補助数詞に使います。一方、欧米では、3桁で区切り、日本流に換算すると「千・百万・十億…」が補助単位です。科学技術関係で目にする「キロ・メガ・テラ…」がそうです。

1.4.4 右書きと左書きと読み

日本語の和数字は縦書きが基本です。現代の日本語表記は、英語と混ぜて書くときにも混乱しない左横書きが普通になりました。しかし、右横書きを実用している場面もあります。例えば、トラックやバスの横腹に文字を書くとき、進行方向を向いて右横腹は、右書きも見られます。全体の文字並びを見れば、読み間違いをしませんが、短い文字並びでは、読み方を間違えることもあります。ただし、電話番号や、英字を混ぜて書くときは、左横書きにしています。小説や文芸誌は、縦書きで印刷することが多いのですが、アラビア数字や英字を混ぜると、編集作業に困ります。新聞は縦書きで記事を編集しています。しかし、ラジオ・テレビの番組や、経済記事、また広告などは横書きが使われています。この横書き部分は、写真版などで別に作成したものを貼り込んで編集しています。製本された書物の狭い背表紙に見出し用の表題を書き込むとき、元の表題が横書きであっても、縦書きにすることがあります。数字や英字を含む場合は、上から書き出す右横書きが普通です。しかし、稀に、下からの右横書きを見ることがあります。用紙を縦長の位置で使うとき、綴じの位置を上にし、製本を左開きにすることがあります。縦に長い構造物の図面を扱うときに見られます。高さ方向の寸法文字列の並びは、下から上に測るように書くことになり、横長の紙面では上下逆の表示になってしまいます。

2. 四則演算の常識を確認する

2.1 算術の基礎教育

2.1.1 一桁の足し算の教育から始まる

小学校の低学年では、基本的な四則演算(加減乗除)を教えます。最初は1桁同士の足し算を習います。俗に言う「1足す1は2」で始める算術です。結果が2桁になると、10の桁を加えます。これが**桁上がり**、または**繰り上がり**です。この教育にたつぷりと時間を割かないと、この後に続く複数桁の足し算と、繰り下がりが起こる引き算の理解ができません。なお、引き算の演習では、全体の結果が負の数になる問題を意図的に含ませていません。負の数の概念はかなり込み入っていますので、後の 2.2 節で説明を補います。

2.1.2 レジスタの概念を算盤で理解する

桁数の大きな二つの数の足し算と引き算とを筆算で進めるとき、位取りを上下に揃えて数字を書き、下位(右側)の桁から一桁ずつ順に計算します。繰り上がり・繰り下がりが必要になることを覚えます。算盤を使う計算は、上位の桁から玉の動かし方を覚えます。算盤は、原理的に、10進数の**レジスタ**(register)です。標準的な算盤は 23 桁あります(図 2.1)。この一続きの長さを、途中で幾つかの小範囲のレジスタに分けて、数値計算に使う部分と、数値をメモする場所に使い分けます。小数位置や、単位区切りを合わせる時の目印に、4桁ごとにマーク(定位点)が付いています。

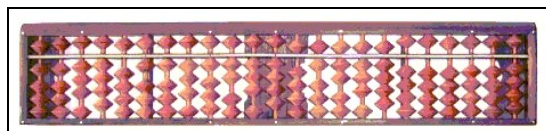


図 2.1 標準的な 23 桁の四つ玉算盤

手回しの機械式計算機(図 2.2)、または電動化した機械式計算機(図 2.3)は、以前、多く使われていました。主な利用者は、銀行などの金融機関でした。科学技術計算にも多く使われていました。これらの機械式計算機は、固定桁数のレジスタが三つあります。標準構造は、10桁×10桁の掛け算の乗数・被乗数が表示できるレジスタ二つと、掛け算の計算結果の表示ができる20桁レジスタです。

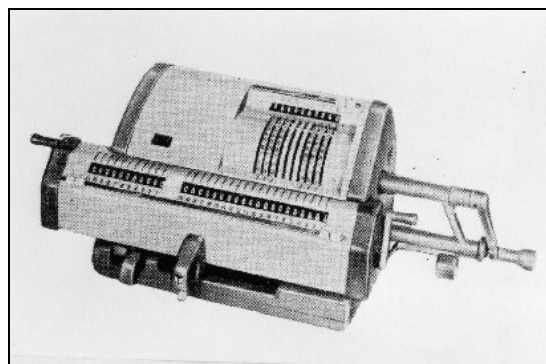


図 2.2 手回しの機械式計算機(タイガー)

機械式の計算機を使って、小さな数から大きな数の引き算をすると、数字列の頭に9999…が詰まります。これが**補数**(compliment)です。算盤を使う計算では、小さな数から大きな数を引くことになるのを避けます。そのようなときは、別の桁位置を使って正負を逆にした引き算をする、と算盤に達者な人が説明してくれました。

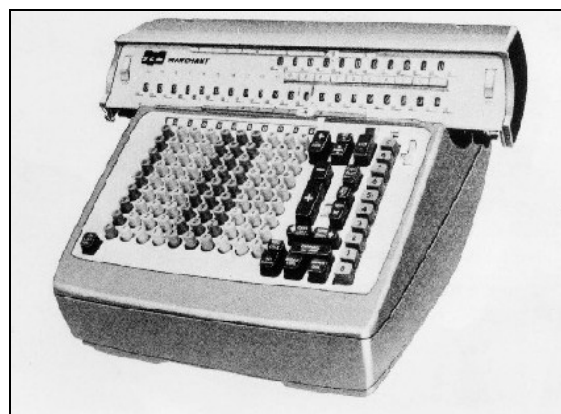


図 2.3 電動計算機(フリーデン)

電卓が普及して、機械式の計算機を眼にすることが無くなりました。標準的な電卓の表示画面は一つです。電卓は、内部に電子式のレジスタを4つ持っていて、 $A \times B = C$ の3つの計算数値を、場面に応じて表示するようになっています。残りの一つは、メモリ用です。操作画面で、Mの字を付けたキーは、メモリの内容进行操作します。

2.1.3 補数の概念が使われている

小学校で引き算を習うとき、同じ桁位置の引き算で、小さな数から大きな数を引くとき、直前の桁から10を借りてくる方法を教わります。これには、意識していませんが、補数の考え方を使っています。例えば $12 - 7$ の計算をしなければならぬとき、引く数7の、10の補数3と、2とを足す計算をして5を求めます。補数の解説は、高校までの数学の教科書では取り上げていません。コンピュータを利用する数値計算を勉強すると、2進数の補数表現の説明があります。しかし、そもそも、補数の概念を習っていなかったことが災いして、この部分についての理解が難しいようです。なお、電卓が簡単に利用できるようになって、小さな数から大きな数を引くと、マイナス符号を付けた、負数の表示がされます。電卓内部のレジスタでは負の数は補数で扱われているのですが、文字表示の処理をして、符号付きの数に変換して表示しています。

2.1.4 掛け算の基礎に九九を覚える

日本では、小学二年生から、掛け算計算の基礎に、「九九」を朗読させて覚えさせます。日本以外では 20 までの数の組み合わせを覚えさせる国もあります。日本語では、一桁の数を一文字のカナで言うことができます。「イチ、サン、ロク、ハチ」はカナ2字の表記です。日本語での発音単位は、英語の音節(シラブル)の区切り方とは違いがあって、1音節と2音節の読みを使い分けます。九九の計算規則は、口調の良い5音節にまとめています。歌にして覚えることもします。声に出すとき、息継ぎの間(ま)も勘定に置いて、6, 8の音節単位に揃えて覚え易くします。子供が計算に九九を応用しているときは、声を出し、自分の耳で聞き直して確かめています。子供は、声を出さないときも、口がもぐもぐと動くのを見ることがあります。大人になると声に出すことをしませんが、頭の中で九九の音節を再現していて、正確さを確認するためのフィードバックを掛けています。理屈で覚えて利用していると、忘れたり、語順を間違えたりしますが、**復唱**は自分の声を聞いて確認する行為ですので、正しい対応につながります。歴史上の重要な事項の年代、電話番号の数字並び、無理数である2や3の平方根の値などを、語呂合わせで覚えることもそうです。逆に、数を決めることにも応用します。他のことと関連を付けた覚え方を応用するのが**連想記憶**です。例えば、言葉遊びに**数え唄**があります。連想記憶は、情報技術の課題としても研究されています。しかし、ここで言いたいことは、声に出すことの有効性です。

2.1.5 掛け算は、九九と足し算を応用する

掛け算の原理は、同じ数を何回も足す処理です。手回しの計算機があると、掛け算が繰り返して足す計算をすることを実感として理解できます。掛け算を代数記号で $A \times B$ と書くとき、掛ける方の数 B を**乗数**(multiplier)、掛けられる数 A を**被乗数**とします。掛け算は、掛ける順番を換えて $B \times A$ としても同じ結果になります。つまり、掛け算規則は**交換律**(commutative law; 交換法則とも言います)が成立します。手計算で複数桁の整数の掛け算をするときは、被乗数を書いて、その下に乗数を書き、一桁の乗数で計算した結果をずらしながら書いて、最後に全体を縦に加算します(図 2.4)。

手回し計算機では、乗数と被乗数とは別のレジスタを使います。被乗数を、定数用レジスタにセットしておいて、ハンドルを回して複数回の足し算で実行させます。その回数を、乗数レジスタで確認します。乗数が二桁以上のときは、計算結果の入るレジスタ全体を左右にずらして足し算の入る位置を変えます。これを**シフトする**と言い、この処理のできるレジスタを**シフトレジスタ**(shift register)と言います。

	1 2 3
×	4 5 6
	7 3 8
	6 1 5
	4 9 2
=	5 6 0 8 8

図 2.4 筆算での掛け算

2.1.6 整数を扱う割り算は二種類ある

整数だけを使う算術の割り算は、きっちりと割り切れる場合と、割り切れなくて余り(あまり: **剰余**)が残る場合があります。小学校で習う筆算の割り算は、掛け算を逆向きにたどる方法を学びます(図 2.5)。中身は、掛け算と引き算の組み合わせです。割り算の式を、 $A \div B$ と書くとき、 A が**被除数**(divident)、 B を**除数**(ordinal number)とし、結果を C としましょう。最初は、九九の掛け算を逆に利用して、割り切れる条件を学びます。それから、長い桁の被除数と除数とを使う割り算を習います。このとき、九九の掛け算と、引き算とを、しっかりと覚えていないと、計算できません。演習の課題は、割り切れる場合から始め、次いで、余りが残る割り算を習います。小学生の年代ではかなり難しい授業ですので、落ちこぼれも出ます。

	答え	4 5 6
1 2 3)	5 6 0 8 8	
-	(×4) 4 9 2	
	6 8 8 8	
-	(×5) 6 1 5	
	7 3 8	
-	(×6) 7 3 8	
	余り	0
	割り切れた	

図 2.5 筆算での割り算

2.1.7 余りを扱う割り算は算術に属する

日本語の環境では、助詞の「で」と「を」とを使って「 A を B で割る」を「 B で A を割る」と言い換えても計算間違いは起きません。しかし、英語では、割り算を言葉で説明する表現が前置詞違いで二種類あります。数学的な意義で割り算を言うとき、「divide A by B 」と言い、 A/B を意味します。もう一つ、「divide B into A 」の言い方も普通に使われています。意味は、「 A を B で割りつける」ことです。割り算の答えは整数(**商**: quotient)で表し、**余り**(remainder)も必要になる計算です。例えば、ロール状の長さの長い紙を、一定長さに切り分けると、最後に半端な長さが残るとき、これが余りです。電卓を使って商と余りの計算をしたいときは、まず普通の計算で代数的な割り算をして、答えの小数以下を切り捨てて商とします。それから、元の数から商 \times 被除数の値を引いて余りを計算します。プログラミング言語では、余り(剰余)だけを求める関数名はMODです。

2.2 引き算と割り算には向きがある

2.2.1 引き算は交換律が成立しない

引き算を代数表記で「 $A-B$ 」と書くとして、書き順を変えた「 $B-A$ 」は、前と同じ計算結果になりません。算術の計算は、結果が正の整数になる場合だけを演習します。負の数を定義しません。小さい数から大きい数を引くことはできない、と約束し、答えが無い、または0とします。この約束は、お金を扱う計算(経理)のときに困ります。そのため、引き算の順序を逆にして計算し、その数を赤字で書くか、三角記号 Δ を数字並びの前に書く表し方をします。補数は使いません。実数も扱う代数的な引き算の約束では、書き順違いの二つの計算結果は、絶対値が同じで、正負の符号が逆に付きす。代数的な数は負の数も使います。そうすると、負の数を引く計算もできます。これは、正の符号に直して加算します。俗に言う「マイナスのマイナスはプラス」です。符号が正の数と負の数とは、全く別世界を表していて、連続的な数の世界ではないのです。その境界が0です。

2.2.2 座標の考えを使う

個数の多い物を一列に並べることを考えます。これを順序数で区別するとき、1の位だけに注目すると、「1, 2, 3, ..., 9, 0」が繰り返して現れます。0からは始めません。身近には、この順序数を眼にすることが多いのですが、マイナス方向にも順序数を割りつける例に、寒暖計の目盛があります。図 2.6 の寒暖計(Wikipedia より)は、摂氏($^{\circ}\text{C}$)と華氏(F:ファーレンハイト)の二種の表示目盛が付いています。日本では華氏目盛を使いませんが、英語圏では、今でも親しみ易い単位として日常的に使っています。華氏の温度目盛は、人が感じることのできる実用的な温度範囲で、判り易い正の整数を割り付け、マイナスの数を使わないように決めたものです。図 2.6 の寒暖計は座標の表示法を使っています。摂氏の表示は、0を境にして逆向きに目盛が振ってあります。マイナス符号(-)が書いてないことに注意します。最近の温度計はデジタル表示が多くなっていて、0度以下をマイナスの記号を表示しています。摂氏の温度系列を使うのは科学的だと思うのですが、必ずしも実用の面で合理的とは言えない面もあります。

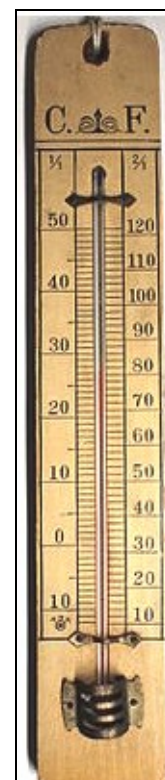


図 2.6 寒暖計

2.2.3 マイナスの寸法は無い

何かの形の寸法を測るとき、物差しを使います。物差しの目盛は、0を起点として一方方向に割り付けます。文房具には、中央を0として左右で逆向きの尺度目盛を付けるものもありますが、一般的ではありません。寸法は、真つすぐに距離を測るだけでなく、曲面に沿わせても測ります。女性の体形を言うとき、バスト・ウエスト・ヒップで測りますが、周長で測ります。図形を図に描いて説明の寸法数値を記入するとき、実物に物差しを当てて寸法を測る作業に使うことを考えて、計算しなくても済むようにします。例えば、外形が円形のパイプであれば、半径ではなく、直径か周長を使うのです。対称な図形では中心から片側の寸法表示で済ませることができるとは思うのですが、実物で、中心位置に物差しの目盛を当てるのが便利になっていなければ、意味がありません。つまり、マイナス符号付き、またはマイナスの概念を含む数を使うことを避けます。

2.2.4 ゼロ(0)で割る割り算を定義しない。

算術の割り算は、引き算の応用です。大きな数を小さな数で割る処理が基本です。引けなくなったときの結果が、割り算の余りです。ただし、0で割る約束をしません。小さな数を大きな数で割る約束もしません。実数を考えた代数的な割り算は、結果を小数で表記できます。日本では、算術の割り算記号に「 \div 」を使い、代数的に書くと、例えば「 $A \div B$ 」と表記します。こちらも「 $B \div A$ 」と同じになりません。代数的には、横線を使う分数で書き、比率を表します。書き表すとき、行数を節約するため、斜め記号(スラッシュ)を使って、例えば、「 A/B 」のように表記する方法も慣用されています。欧米語には、割り算を約束する一般的な記号がありませんので、割り算の意義を表す表記法に苦労があるようです。スラッシュ「 $/$ 」の他にコロン「 $:$ 」も見られます。「 $A:B$ 」と書くと、日本語では「A対B」と読み、比率を意味する、と解釈します。一般的には勾配の表記に多く見られます。このとき、Aは高さ、または縦方向、Bは横方向の寸法として使います。例えば、勾配で言えば、1:5、20%、日本では2割などと書き、割り算の意義があります。壁面のような急な勾配を言うとき、10:1、のように応用することもありますが、直観的に理解できないことが起こります。コンピュータ(電卓も含めます)を使う数値計算では、0で割る場面が現れたときは、「エラーである」と通知してきて、処理を中断します。これは、親切な処理ですが、計算の全体流れの実態を知りたいときに困ることもあります。そのため、0で割る事態のとき、そのコンピュータで実用できる最大数値を仮に代入して、計算処理を継続させるようにすることがあります。

3. 数を表す方法

3.1 意味を持たせて数を書く

3.1.1 有限桁数の数字並びを決める

数学で言う**実数**(real number)は、大きな数から小さな数まで、連続した全体です。幾何学的なモデルは、無限に長い直線座標上に、切れ目なく、連続的につながった点の座標値です。抽象的には、 $-\infty$ から $+\infty$ の範囲です。この座標上には**整数**(integer number)も**無理数**も含まれます。実的には、有限桁数の数字並びで表します。したがって、小数点がある数であっても、小数点の位置は便宜的なものであって、計算に利用する実際は有限桁数の整数で扱います。無限に数が並ぶ**循環小数**、例えば、 $3.3333\dots$ は、 $10/3$ で正確に表すことができます。この性質を見て、昔のギリシャ人は、どのような数字並びの実数であっても、整数を成分とする分数で表すことができる(これが**有理数**の定義)と仮説を立てました。しかし、例えば、正方形の対角線の長さは、2の平方根の数値が要るのですが、分数表示ができないことが証明されましたので、ここのような数を**無理数**と分けしました。そのため、もし正確に無理数を表したければ、無限に数字を書き並べなければなりません。そこで、無理数であっても、実用になる桁数の有理数に切り詰めて使います。これは、正義感を持って数を学問的に扱いたい人には、我慢できないごまかし、または妥協と映るようです。一時期、正義感から「小学校の教育で円周率を3と教えるのは間違いだ」の抗議が起きたことがありました。

3.1.2 数の大小を区別する約束

数の大小を区別する約束は、二通りの方法を使い分けられていることに、気が付いているでしょうか。正の数と負の数を含めた全体を、直線上の座標位置で理解することを一次元の**座標幾何学**と言い、この直線を**数直線**と言います。水平に置いた直線で0の位置を約束しておいて、右側を正、左側を負の数値を表す領域としましょう。数直線上の2点は、相対的に右側の数が、左側の数より大きい、とします。この判断には、原点の位置がどこにあるかには無関係です。ところが、原点からの距離の大きさで判断することもあります。こちらは、数の絶対値での大小比較です。一般社会の習慣では、数直線の向きに、「左右・上下・南北・東西」などの区別を見ます。前の第 2.2.2 項で説明した寒暖計は、2組の直線座標(摂氏と華氏)を縦の直線座標で示しています。原点の位置と尺度とがそれぞれ違います。同じ温度を表す数字が違いますので、数の大小を温度の大小比較に使うことができません。温度の大小を区別したいときは、一つの座標系に揃えます。一方の座標値が、他方の座標値でどうなるかの計算をします。代数的には、関係を表す一次方程式を立てます。幾何学的には、縦横二方向に直線座標を描き、直線のグラフを描きます。これは、一次元**座標変換**を表す方法です。

3.1.3 有効数字と有効桁数を理解する

数をアラビア数字で書いて表すとき、1~9の数字並びの前後にある0を除き、途中の0はそのままとし、残る数字並びが**有効数字**(significant figures)です。大きな整数で、有効数字の後ろに0が続くとき、どこまでの0が意味を持つか、つまり、0を有効数字とする境界が曖昧になります。たとえば、金額の表示で、「千円未満を切り捨て」と注意書きがあっても、下3桁に、形式を整える0が書いてあるときです。そのため、数としての意義を持つ数字並びの桁数を**有効桁数**とします。科学的に書くときは、例えば 1230×10^3 のように、べき乗の数を単位に付けます。実数では、小数点以下にも数字が並ぶとき、有効数字として意味を持たせる範囲の0は残します。例えば、5.00の有効桁数は3です。実数は、有効数字と有効桁数とが分かる表記法を使います。例えば、 1×10^3 や $1.e3$ は有効数字が1桁であり、 1.000×10^3 は4桁であると明示できます。

3.1.4 数を文字として使うこともある

第 1.4.3 項で、英語の環境では、数字(digit)と英字(Letter)とは別体系に分類する、と説明しました。コンピュータプログラミングで使う変数名などは、英字と数字とを混ぜてIDとして使うこともします。名前であることをコンピュータに知らせるため、名前の頭字は英字で始めます。日本では頭字に漢字を書くこともします。例えば、自動車の登録番号があります。数字並びだけでも名前のように使う例に、クレジットカードに見るようなIDがあります。この場合、書きだしの数字に0を使うと、コンピュータ処理では頭にある0を無視して数字並びと解釈することがあります。銀行などの窓口事務で扱う用紙では、ID番号ならば左詰めで記入し、金額などは右詰めで記入します。金額文字の左を空白にしておくと、金額の追加記入の誤りが起きる危険がありますので、¥記号を頭に書くのです。表計算ソフトのMS-EXCELでは、データを書き込む枠(セル)に書式を決めておくことができます。数を入力すると、標準では右詰めで記入されますが、文字型と約束してあれば、左詰めで表示します。なお、データとして入力した数値の表示は、丸め(後の第 3.3 節で説明します)や、指数表示なども使うことができますが、丸めの処理がされた数値は、内部で保存してある値とは異なることがあります。

3.2 固定小数点と浮動小数点

3.2.1 実数の実体と表示法の選択

数そのものを書いて示すとき、有効桁数の数字並びに、余分な0が付くことがあります。そうであると、全体の文字並びが長くなって、読み難く、レポート全体での体裁も悪くなります。全体の文字並びを詰める方法に、**浮動小数点法**(floating point)があり、しない方が**固定小数点法**(fixed point)です。浮動小数点数の表記は、 $A \times B^n$ の形です。Aは、小数点の位置を適当に移動(浮動)させます。実数では数の左に移し、有効桁数が分かる方法も考えます。整数では、小数点位置が最後尾の桁の右に在るとした暗黙の約束を使って、小数点を書きません。浮動小数点数では、A、B、nの3種類の数字列または文字列を使うこととなりますので、コンピュータメモリのビット並びの使い方に工夫が必要になりました。四則演算では、小数点位置を揃えるシフト計算を先行させます。計算結果を保存するときも、決められた書式に合わせる処理が必要になります。

3.2.2 電卓は固定小数点表示を使っている

普通の電卓は、計算に使う桁数の大きさで、7桁、10桁、12桁などの種類があります。電卓を使うとき、算盤と同じように、全部のレジスタをクリアする、**御破算**のキーがあります。画面が0を表示しているときは、データ入力待ちの状態です。数を入力すると、整数が入力されたとみなされ、小数点の位置は右端のままです。数の入力が続けて、表示画面の左端を越え、表示画面の許容桁数以上を入力しようとするエラーになります。これが**オーバーフロー**です。一方、実数を入力するときは、整数の入力の途中で小数点を入力します。さらに続けて数を入力すると、小数点の位置が左に移動して行きます。許容桁数以上は、入力を受け付けませんが、エラーにはなりません。整数部分の大きな実数の掛け算をすると、整数部分の桁数が許容桁数を超えることがあります。このときもオーバーフローのエラーになります。つまり、小数点が画面のどこかに無いとエラーになります。一方、実数の数値が1より小さな数で、小数点以下に0が多く詰まっている数同士の掛け算をして、0の数が増え許容桁数以上になると、エラーの表示にはなりませんが、0だけが残ります。こちらが**アンダーフロー**です。このような数の扱いが、固定小数点法です。

3.2.3 パソコン付録の電卓は浮動小数点法を使っている

パソコンでは、アクセサリ機能として、電卓を擬似的使うことができます。「電卓画面をデザインすること、キーを押して計算機能を行わせること、数を場面に応じて表示させること」、これらのプログラミングは、オブジェクト指向プログラミングの例題としての教育価値があります。ただし、四則演算の算術記号は、全角の(+ - × ÷)ではなく、半角の(+ * - /)になっています。数値計算の機能では、コンピュータ側の高度な計算機能を使いますので、計算できる数値の桁数は32もあります。オーバーフローやアンダーフローを起こす最大最小の数も、コンピュータの機能の方を使っています。さらに言えば、数値を浮動小数点法で計算しています。丸めの機能もありません。したがって、具体的な数値計算法の勉強では、市販の普通の電卓と、パソコンのアクセサリ電卓との、動作と、機能の違いとを理解し、その機能を生かす使い方を納得することが大切です。

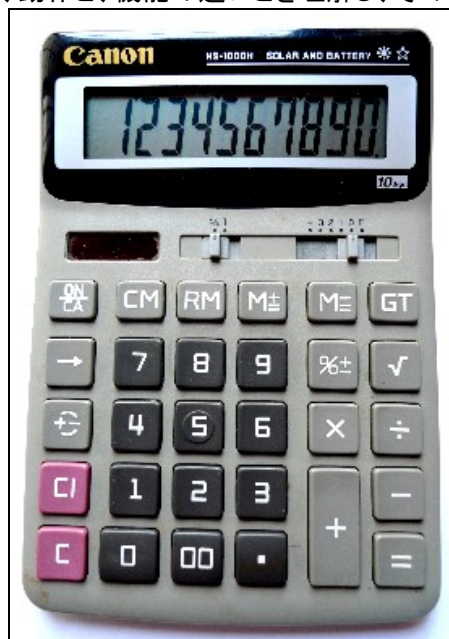


図 3.1 普通の電卓



図 3.2 パソコンのアクセサリにある普通の電卓

3.2.4 普通の電卓の賢い使い方

普通の電卓は、種々の商品があります。お金の計算ならば、7桁あれば、100万円単位までの計算に使うことができます。科学技術計算で、数値の精度を気にするときは、12桁を使います。電卓は、計算結果の「切り上げ・切り捨て・四捨五入」の選択ができて、その処理をする桁位置を選択できます。これが**丸め**(rounding)です。次節で詳しく説明します。電卓では「cut,up,5/4」表示されている切り替えスイッチで選択します。小数点を持つ実数の丸めでは、小数点以下何桁目で丸めるかの桁位置の指定とセットにして使います。なお、筆者は、数学的な関数機能として、ルート(sqrt)計算ができるものを使っています。無い製品もあることに注意します。算術の手順を代数式で書いて示すとき、例えば、掛け算は「 $A \times B$ 」、割り算は「 $C \div D$ 」とし、その順で数値を入力します。記号式の数の呼び方は、 A が**被乗数**、 B が**乗数**、 C が**被除数**、 D が**除数**です。演算記号 $\times \div$ のキーを押して、後から入力した数が定数メモリに残りますので、イコールキー「 $=$ 」を続けて押すと、加減乗除が繰り返して実行されます。例えば「 $2 \times 2 = = = \dots$ 」とキー入力すると、結果の数が、2, 4, 8, 16…の順に表示されます。標準の電卓は、メモ書きに使うような独立したレジスタが一つ付きます。計算の途中結果を紙に書いて残し、後の計算に使うとき、再度数値を入力し直さなければなりません。例えば、分数の計算では、分母を先に計算して用紙に書いておく、などです。この用紙に代わるものがメモリ用レジスタです。商用計算の電卓では、余分にもう一つ、幾つかの計算結果を順に加算して行くGT(grand total)の処理ができるメモリを持っています。メモリ用レジスタを賢く使う方法を工夫すると、紙に書いて記録しておく手間を節約できます。また、この手続きを理解しておく、数値計算の賢いプログラミングを書くときの役に立ちます。

3.3 計算の組み立てを文に書く

3.3.1 数式の書き方

数学で言う**数式**は、文字並びで、計算の組み立てを示した**文(statement)**です。英語では **mathematical expression** と言います。数式は、実践的には数値計算に応用しますが、代数式のまま説明にも使います。文字並びは、「数字(定数など)・数の代りに使う名前を表す文字・記号(演算子や括弧など)」の組み合わせです。算術の加減乗除に使う記号(+ - \times \div)は、代表的な**演算子**(operator)です。数式は、それを**評価**(evaluation)すると、或る確定した値として使うことができます。数値計算は、評価の一つの方法です。一単位としてまとめた数式を、別の名前にして代表させる方法が、例えば「 $Y = A + B$ 」のように表したときの Y です。意味としては、イコール記号「 $=$ 」の左右の数式は同じ値であることを表し、**等式**(equation)と言います。コンピュータのプログラミング言語では、**代入文**(assignment expression)と言い、右辺の式を計算して Y に代入すると約束します。この数式は、三つの部分で構成されています。「 Y 」、「イコール記号 $=$ 」、「数式($A + B$)」です。

3.3.2 計算手順の説明に使う文

電卓を使って数値計算をするときの手順を、文字並びに書くとします。例として下の式で説明します。

$$「A+B \times C - D / E =」 \quad \rightarrow \text{答え} \quad \dots (3.1)$$

具体的に、「 $A \sim E$ 」に「1~5」の数を入れて、電卓のキーをこの順で押すと、答えに1が得られます。この式を、数学的な代数式と解釈するときは、計算手順を間違えないように;

$$「A + (B \times C) - (D / E) =」 \quad \rightarrow \text{答え} \quad \dots (3.2)$$

と括弧()でくっつけて書く約束です。その根拠は演算子に**優先順位**があって、数式の文字並びの、どの部分から先に計算するかを決める規則です。加減乗除だけを考える場合、乗除の演算子「 \times と $/$ 」は同順位、加減の演算子「 $+$ と $-$ 」も同順位です。しかし「 \times と $/$ 」は「 $+$ と $-$ 」よりも順位が高い、と約束します。式(3.1)では、 $(B \times C)$ と (D / E) とは、前もって計算値を求めて、それを仮の変数記号、例えば「 P , Q 」に代入しておきます。最後に、加減算だけを使った式を計算します。ここで、括弧()は、その内側の数式が最高の優先順位を持つ、と約束します。括弧を入れ子状に使うこともあって、最も内側から計算を進めます。括弧の内側の値が確定したときに括弧を外した式に変更します。最後に加減算だけの式が残ります。文字並びに書いて説明すると、三つの式です;

$$「B \times C \rightarrow P; \quad D / E \rightarrow Q; \quad A + P - Q \rightarrow \text{答え} \quad \dots (3.3)$$

例として、上と同じように「 $A \sim E$ 」に「1~5」の数を入れて計算すると、答えに 6.2 が得られます。式(3.3)までは代数式とは言えません。これらは、計算手順の説明をするための便宜的な書き方だからです。

3.3.3 代数式は文法規則と関連がある

数学的な代数式は、例えば、「1足す2は、3です」の言い方は、変数名を約束しておいて、例えば；

$$「C=A+B」 \quad \dots (3.4)$$

の表し方をします。実は、式(3.4)は、英文の「C is equal to A plus B.」を記号化したものです。主語に当たるCを頭書き、述語を後に続けます。コンピュータのプログラミング言語では、式(3.4)の形を書きますが、実際の計算手順は、機械語に**翻訳**(compile)して、先に右辺の(A+B)を計算しておいて、左辺のYに代入する順に組み立て直します。学問としての数学は、式(3.4)の表し方をしますので、文章並び順と実際計算処理の順が逆です。この逆順の約束は、欧米人でも混乱しますので、プログラミング言語Cのマニュアルにも解説があります。**左辺値**、**右辺値**の用語があります。左辺値を表す変数名は基本的に代入で値を決めます。右辺値は、値を参照する目的に使い、値を変更することはしません。日本語では「Cは、A足すBで計算する」とも言えますが、「A足すBはCです」の言い方が普通ですので、書き順と計算順とに混乱を起こしません。

3.4 丸めの方法と応用

3.4.1 数字並びの前にある0を考えない

数を数字並びで書き表すとき、数の精度の指標が**有効桁数**です。整数表記は0を頭書きしません。実数では、絶対値が1より小さな数は、小数点以下に0が詰まることがありますが、この0も有効桁数に含ませません。例えば、無理数の円周率を数値計算に使うとき、3.14 とすると3桁、3.1416 とすると5桁の**有効数字**を持たせると言い、最後の桁に±0.5の誤差がある、と判断して使います。数字並びの後ろに0が続くとき、どこまでの0が数の精度に関係を持つか(数として有効)の判断が必要です。整数では、暗黙の約束では、すべての桁が有効であると判断します。実数では、数の精度を示す必要がある0は残します。例えば、3.0、3.00、3.000は、有効数字の桁数が、2、3、4桁であることを示しています。数値計算では、できるだけ、数の精度が保障できる桁数に整えます。その具体的な処理が**丸め**です。実用する方法は、**切り上げ**(round up)、**切り捨て**(round down)、**四捨五入**(round off)の3通りです。

3.4.2 用語の数学的意義を理解しておく

数の丸め方法を区別して言葉で言うとき、「**以上・以下・未満・を超え**」の用語が出てきます。その意味と使い方を理解しておきます。これらの用語の詳しい説明は「実用文書のまとめ方」としてインターネットに載せてあります。その付録Gに、常用語句の使い方と説明してあります。その要点をまとめておきます。例えば、整数を扱うとき「100以上」では100の値を含み「100,101,102,…」の並びです。「100以下」と言うとき、100の値を含め、「100,99,98…」の意味です。「100を超え」と書くと「101,102,…」の並びであって、100を含みません。「100未満」と書くと、「…97,98,99」で止めます。

3.4.3 お金の計算で使う丸め

お金が絡む計算は、切り上げと切り捨てが実用されています。お小遣いをもらう場面で、1253円が必要な金額であると、100円未満を切り上げて1300円を渡すのがそうです。桁数が多くても、正確な数値を扱っている業務の代表は、銀行などの金融機関です。利子や税率の計算は、元の金額に利率の掛け算処理が入りますが、その計算では1円未満は**切り捨て**の約束です。日常的には、消費税の計算があります。銀行預金の利子の計算も、元金に利率を掛けた利子は、1円未満を切り捨てます。利子との合計を次の元金にします。そうすると、例えば、年利率が3%であるとき、30円だけを預けると、利子は1円未満の0.9円ですので、いつまで預けておいても利子は加算されません。この規則は、想像している以上に厳格な計算法です。経理の計算のとき、最後の桁に1円でも誤差が発見されると、どこで誤差が出たかを徹底的に調べることをします。桁数の多い整数をまとめて説明に使用したいとき、有効桁数を小さくするように、後の桁を0に丸めることをします。例えば、お金の計算では、千円以下を切り捨てる、などの処理をします。そうすると、下三桁はいつも0が並びます。このとき、(×千円)の単位を書いて、0の数を省いた数を書くことがあります。

3.4.4 実数を整数化する丸め

小数点以下に数字が並ぶ実数を、整数に丸めることは、元の実数の或る範囲を、整数で代表させることです。その範囲がどのようになるかを、図 3.1 に示しました。切り上げは、0 の範囲がありません。これは「かぞえ」の方法です。日本では、建物の階数の数え方に応用しています。建物は高さ方向の寸法を持ちますが、床から天井までが階の範囲です。切り捨ては、0 の範囲が2倍です。四捨五入は、マイナスからプラスまで、0 を含めて、実数範囲がすべて同じ長さです。これは、実数の数値計算に応用すると、計算誤差を最小にできる丸めです。コンピュータでは、実数型の数を整数に変換して使いたいときの方法が幾つかあります。実数型の変数を整数型の変数に代入すると、四捨五入で丸められます。切り上げまたは切り捨てにしたいときは、元の実数に0.5 または-0.5 を足してから代入します。整数化にする関数に、MS-EXCELには、INTとして定義されている関数があります。マイナスの実数は、整数部の絶対値が大きくなることに注目して下さい。

実数座標値	...	-3	-2.5	-2	-1.5	-1	-0.5	0	0.5	1	1.5	2	2.5	3	...
座標目盛		:	:	:	:	:	:	:	:	:	:	:	:	:	:
切り上げ	...		-3	-2	-1	0	1	2	3	...					
切り捨て	...	-3	-2	-1	0	0	1	2	3	...					
四捨五入		-3	-2	-1	0	1	2	3							
INT 関数	...	-3	-2	-1	0	1	2	3	...						

図 3.3 実数を丸めた整数値の実数範囲

3.4.5 覚え易くするために呼び数を使う

数学で言う無理数は、正確に表したいならば、小数点以下の桁に無限に数が並ぶのですが、適当な位どりのところで丸めます。つまり、実質的には整数扱いをします。例えば、円周率(π)の表し方があります。常識的には、覚え易い3.14と小数点以下二桁の数を使います。分数表示の22/7も便利です。このような、実用に使い、正確な理論値にこだわらない数値を**呼び数**(nominal number)と言います。設計計算などで正確な計算が要求される場合「桁数が多くなる数、材料寸法に種々の種類がある」などのとき、覚え易い数値に丸めることをします。日本は、アメリカに多くの技術を学びましたので、元の寸法がインチ系列であったものを、ミリ単位の呼び数で使う例があります。例えば、1インチは25.4mmですが、呼び寸法を25mmにします。7/8インチは22mm、3/4インチは19mmとします。元のインチ寸法の材料を測ると、呼び寸法のミリとは違うことがあります。そのため、材料検査をする段階で、「寸法をごまかした」と問題にされたことも起きました。インチ系列とミリ系列の寸法とが混在して、混乱する例に自動車などで使うネジがあります。見かけは同じ寸法に見えても、ネジとして合わない組み合わせができない悲劇が起きます。これを避けるために、国際的な話合いでISO規格が合意され、日本ではその取り決めに尊重して工業標準(JIS)として採用しています。

3.4.6 大小の寸法系列を合理的に決めたい

我々の身の周りには、寸法違いで大小さまざまな品物が使われています。一つの例として、お皿を考えます。食器棚には、小皿から大皿まで、各種の寸法を用意しますが、寸法を選ぶとき、一回り大きい、または小さい、の言い方をします。工学的に判断するときは、等比数列になるような寸法系列で、大小の種類を決めます。上の項で触れた、工業で使う大小違いで多くのネジを必要とするとき、等比数列になるような寸法系列を決めておくと、寸法種類の増加を抑えることができます。意図的に等比数列を決める基準に、JIS Z 8601 **標準数**があります。これは1から10までの尺度を常用対数目盛(0~1)で表示しておいて、それを10, 20, 40等分した座標値の数値を、丸めて使うように提案したものです。10分割のときの数値列は、「1.00, 1.25, 1.60, 2.00, 2.50, 3.15, 4.00, 5.00, 6.30, 8.00, 10.00」です。

3.4.7 お金の小数単位にセントがよく使われる

セントは、ラテン系の言語で100の意味がありますので、セントの言い方は、通貨の下位単位に多くの国で採用されています。アメリカでは1ドルの100分の1です。それ以下の単位名はありません。日本も、戦前までは円の下位単位に銭と厘とがありました。銭とセントとは音が似ています。敗戦直後のインフレで通貨の円の価値が約百分の一に下がり、銭以下の単位を使わなくなりました。数の桁数が増えましたが、小数を使う場面が無くなったことの利点もあります。しかし、お金に下位単位があると、小数点以下二桁までを扱う場面を実数を説明するときに判り易くなります。例えば、職人言葉では、長さをメートル法で言うとき、1m50 cmを1円50銭のように言うこともしました。

4. 論理演算の約束を覚える

4.1 コンピュータでは論理演算が使える

4.1.1 数値計算に応用する2値論理学

論理学(logic)の歴史は、紀元前、ギリシャのアリストテレスにまで遡る歴史の古い文科系の学問です。論理学は、言語学(linguistics)の一分野です。主語・述語の整った文(命題と言います)が正しいか(真 true; 1を当てる)、そうでないか(偽; false; 0 を当てる)、を判断する推論に使用します。推論の最も単純な構造は、「前提」の命題と「結論」の命題の組みです。これに代数的な方法を応用するようになったのは20世紀からの記号論理学です。代数学と同じように、文字や記号を使って文の性質や関係を扱います。或る命題が「真であるか偽であるか」の二つの性質だけで判断に使う論理学を、二値論理学と言います。この性質を数学的に扱うとき、2進数の(0 または 1)を持つ変数を当てます。この変数間の算法を決めた代数学をブール代数と言います。

4.1.2 論理演算の用語と記号に混乱がある

論理学が文科系の学問であった歴史がありますので、論理演算の種類を言う用語は、難しい漢語も使われています(表 4.1)。日本語のワープロ環境では、漢字を使うことができますし、特別なグラフィックス字形の論理記号も使うことに不便はありません。しかし、英語の環境では、論理記号の表し方に苦勞があります。

表 4.1 種々の論理用語と論理記号

論理学用語	論理学の記号 (*1)	集合論の記号	計算機言語 (*2)	英文 (*3)	和文 (*3)
否定 (*1)	$P, \neg P,$	記号に上線を引く	NOT	not	…でない
選言、論理和	$P \vee Q$ (*4)	$P \cup Q$ (*5)	OR	or, and/or	または
排他的選言、排反、非両立	$P \underline{\vee} Q$ (*6)	$P \neq Q$	XOR	exclusive or	どちらか
連言、論理積	$P \wedge Q, P \& Q$	$P \cap Q$	AND	and	かつ
内含、含意	$P \rightarrow Q,$	$P \supset Q$ $Q \subset P$	IMP	if~then, imply	ならば
可逆的内含、対等、同値	$P \equiv Q$	$P \Leftrightarrow Q,$	EQV	equivalence	等しい
全称記号		$\forall x, \Pi x$		all	すべての…
存在記号		$\exists x, \Sigma x$		some	あ(或)る…
帰属		\ni, \in		inclusion	(部分を) 含む
包摂		\supseteq, \subseteq		subsumption	(全部を) 含む

注

- (*1)この列の説明では、論理変数の記号(取り消し線を使う \neg)の方を表記に用いました。式全体を否定にする場合は、式を括弧()で括って、その前に \neg の記号を書きます。
- (*2)BASIC 言語で用いる論理演算子の表し方を例示します。
- (*3)日常言語のなかで使われる表現と、論理学での定義とでは少し異なることがあります。
- (*4)全角英字のVと字形が似ていますので注意して区別して下さい。
- (*5)全角英字のUと字形が似ていますので注意して区別して下さい。
- (*6)選言記号 \vee にアンダーラインを付けた表記です。

4.1.3 論理用語と論理記号

算術は加減乗除の4通りの演算規則があります。二つの論理変数間の論理演算を扱うとき、その組み合わせをすべて数え上げると16通りもあります(図 4.1 参照)。個別に演算名を当てることもしますが、基本とするのは4通りに絞ります。表 4.1 の計算機言語で言うと、「OR, XOR, AND, IMP」です。一つの論理変数に適用する演算に**否定**(NOT)があります。これは、数の計算の符合変換に相当する、と考えることができます。否定の演算は、論理値の0を1に、1を0に変換します。他の演算則に名前を当てることもしますが、上の五つの演算子を組み合わせれば同じ演算ができます。ORとANDは、算術の和と積の演算と性質が似ていることから、「論理和、論理積」の用語が実用されています。日常見る作文の中で眼にする「または」と「と」の使い方に似ています。「PまたはQ」の言い回しは、「PまたはQ, または、その両方」の意義で使います。XORは、「PかQかどちらか」に限定する意義ですので、**排他的選言**の専門用語があります。IMP(**内含**)は、用語も特殊ですし、演算の約束も難しく、また、引き算や割り算と同じように、交換律がありません。EQVは、イコール記号「=」で代用できますが、場面しだいで異なった解釈が起きます。

4.1.4 集合論の考え方も使う

同じ性質を持つ複数の物を、まとめて扱う考えが集合です。これに数学的な方法を応用したものが、狭い意味での集合論(set theory)です。論理演算を説明する図に**ベン図**(図 4.1)があります。白黒写真は、小さな黒い点の集まりです。この個別の点の、或るまとまり(集合)ごとにP, Qの名前を付けて仕分けしておいて、その重なり方の性質を分類してグラフィックスで表したものです。個別の点に2進数の(1, 0)を当てると約束をすることで、数学の顔をもつようになります。コンピュータの一単位のメモリは複数のビット並びですので、個別のビット位置を定義して使えば、相互の区別ができる複数の点の集合を定義できます。このメモリは、データ型として整数の性質をもちます。数値計算で論理演算をさせるときは、論理変数に整数型のデータを使います。

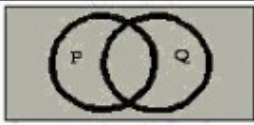
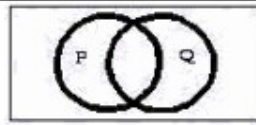
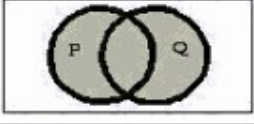
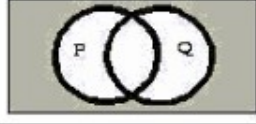
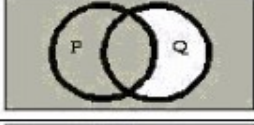
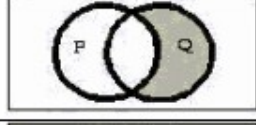








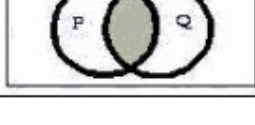

	-15- PとQのトートロジー		-0- PとQの矛盾
	-14- $P \vee Q$ PとQの論理和		-1- ($P \vee Q$) の否定
	-13- $Q \supset P$ 「QならばPである」		-2- ($Q \supset P$) の否定
	-12- 「Pである」		-3- 「Pの否定である」
	-11- $P \supset Q$ 「PならばQである」		-4- ($P \supset Q$) の否定
	-10- 「Qである」		-5- 「Qの否定である」
	-9- $P \equiv Q$ 「PとQは同値である」		-6- 論理的選言、排反 <u>exclusive or</u>
	-8- $P \wedge Q$ PとQの論理積		-7- $P \mid Q$ PとQの非両立

図 4.1 ベン図

4.2 論理演算の基礎知識

4.2.1 初等算術の教育課程にならう覚え方

小学校で教える算術は、「1足す1は2」から始め、順を追って掛け算九九を覚え、割り算に進むことを、第2章で説明しました。論理演算も、数値計算(算術)をして演習できると具体的に理解が進みます。その道具には、パソコンのアクセサリから参照できる電卓を、関数電卓の表示モードで使うのがよいでしょう(図4.2)。市販の関数電卓は、数値計算に使うことを目的としていますので、論理演算の機能がありません。パソコンの関数電卓では、「AND, OR, XOR, NOT」の演算キーがあります。IMPがありませんが、教育利用には十分です。また数の入力でも「2進、8進、10進、16進」の選択ができます。切り替えをすることができて、同じ数を別形式で見ることができます。演算指定のNOTは、10進数以外は補数を計算してくれます。10進数の表示のときは、符合変換として動作します。

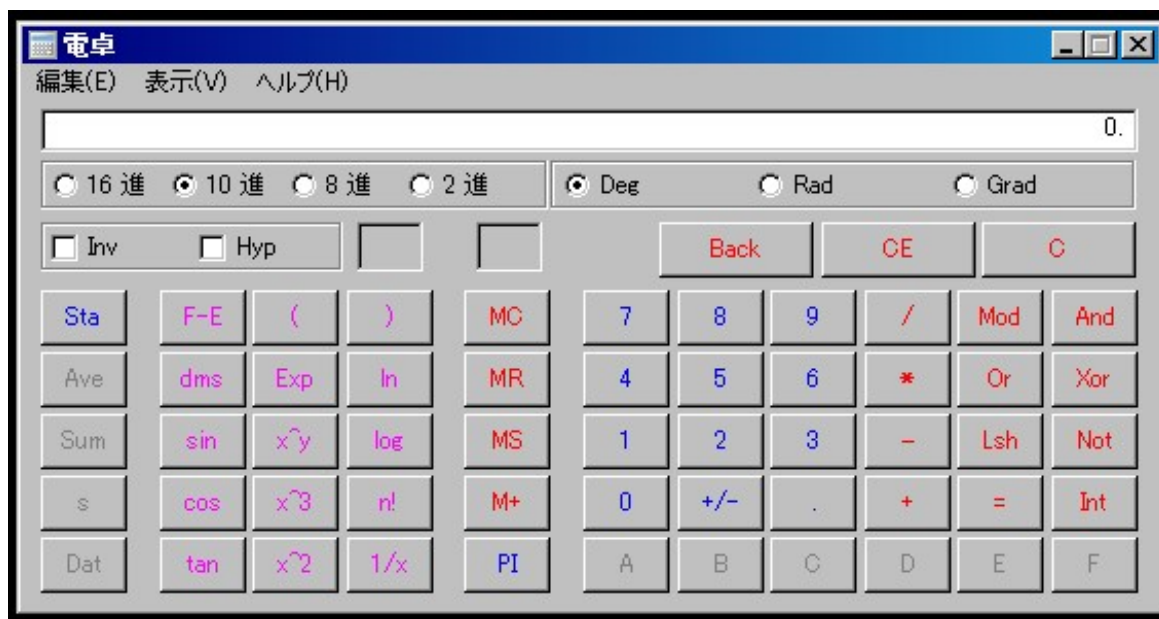


図 4.2 パソコンで利用できる関数電卓の画面

4.2.2 論理演算の具体的な演習例題

- (1) 図 4.2 の関数電卓をパソコンの画面に表示しておきます。論理演算に使うときは、数値の種類として2進、8進、16進のどれかを選びます。そうしておいて、論理値を記憶する整数型の変数のビットサイズを、1, 2, 4, 8バイト(byte, word, dword, qwordと表示されています)のいずれかに決めます。ここでは、2進数とbyte単位を選ぶとして説明します。
- (2) 入力画面をクリアして表示画面が0であることを確認します。そうしておいて、Notのキーを押します。1バイトは8ビット並びですので、結果は1が8個並び、(11111111)が表示されます。これが8ビットサイズの整数型データの、0の補数です。
- (3) 上の状態で、数値の種類を8進数~16進数と換えれば、順に「377, 255, FF」と変わります。10進数を扱う整数型のデータには、C言語では符号付き整数と符号なし整数の区別があります。すべてのビット並びが1であるデータは、符号付き整数では-1とする約束です。パソコンの関数電卓では、論理演算に使うデータ型に、有限ビット数の並びを符号なし整数型として使っていることに注意します。
- (4) 二つの論理変数をP, Qで表すとします。P, Qそれぞれの取りうる論理値は(0,1)の2種類です。或る一つの論理演算子(例えば論理積 \wedge)を使うとしましょう。そうすると、 $P \wedge Q$ の演算の結果が4種類あります。それを公式並みに覚えておく必要があります。これは、論理演算の九九です。この公式をグラフィックスで説明するのが図 4.1 のベン図です。そこで論理代数を応用するとして、Pを(1100)、Qを(1010)のビット並びの集合で一括して表し、パソコンの関数電卓を利用して、論理積の演算結果の種類を求めることができます。その手順を次のように進めます

・全体をクリアします。・キーボードから「1100」と入力します。・電卓の表示画面で「And」のキーを押します。・続けてキーボードから「1010」と入力します。・最後に「=」キーを押すと結果のビット並び「1000」が表示されます。

4.2.3 数式を命題とする考え方

数式を「論理を持った文」として扱うことを、ここでは考えます。例として、単純な数式として、「 $Y=AX+B$ 」と表記されているときの意味を考えることにします。これには、次のような種類の解釈があります。

- (1) Yは、 $AX+B$ で計算できると解釈する。ここで、Xは独立変数、Yは従属変数と言います。
- (2) Yは、 $AX+B$ と同値であると解釈します(等式の意味)。
- (3) $AX+B$ だけを取り出すときは、単純に一次式と言います。
- (4) $Y=AX+B$ が、どのような条件のときに成り立つかを問題にするととき一次方程式と言います。このときのイコール記号は関係演算子です。
- (5) 一次方程式を解くというのはYの値を与えてXの値を求めることを指します。
- (6) 直線のグラフを描くときに利用します(Aは直線の勾配、Bは直線がY軸と交わるときの座標)。
- (7) $AX+B$ を計算してYに代入する(プログラミング言語では代入文です)。
- (8) if()の入れ子に式全体を使うときは、論理値の true、または false を計算します。
- (9) 「 $Y=AX+B$ 」とイコール記号=を挟んで、左右が同値であるかないかの論理値を調べるとします。この全体が論理式であることを区別するとき、同値記号「 \equiv 」に換えることもあります。C言語では等しいを意味する関係演算子は「 $==$ 」と並べて書きます。

4.2.4 論理演算を幾何モデリングに応用

図 4.1 に示したベン図は、論理演算の種類を判り易く説明したものです。二つの論理値の集合P、Qは、円周の外枠で囲っています。この外枠は、別の実体(オブジェクト)であって、論理演算には関係がありません。したがって、論理値の集合が在る実の部分と空の部分の境界を判り易く示すことが目的です。したがって、この境界の両側がどちらも実であるか空であるとき、消去すれば、実の、在る部分の図形が判り易くなります。

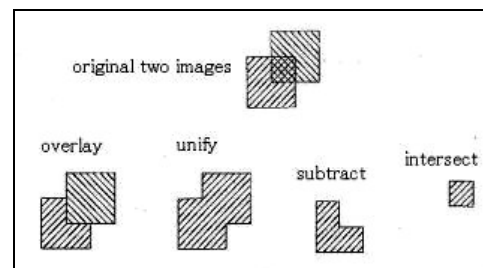


図 4.3 平面図形の干渉処理

図 4.3 は、二つの正方形図形に干渉処理を施して、新しい平面図形を合成する幾何モデリングを例示したものです。干渉処理は、足し算(論理和)、掛け算(論理積)、引き算に分類してあります。このデータ構造は、図形要素の辺(オブジェクト)にトポロジ的な特性を持たせていて、辺が図形の実の部分と空の部分とを分けると定義します。二つの図形の干渉処理をするとき、相手の図形に埋もれる辺と、辺の両側が空になる辺を消去します。

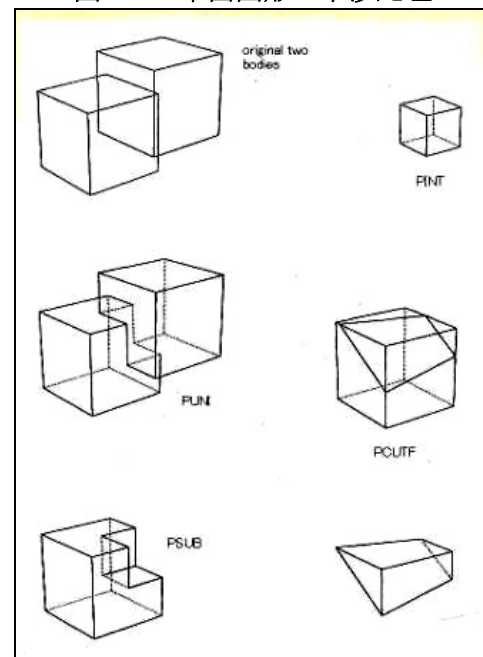


図 4.4 立体図形の干渉処理

図 4.4 は、二つの立方体に干渉処理を行わせた幾何モデリングの例です。立体図形の図形要素は面にトポロジ的な性格を持たせ、面は、立体図形の内側の、実の部分と、表面から外側の空の部分、との境界です、これもオブジェクトの扱いをして、相手側の立体の中に埋もれる場合は消去します。干渉処理の論理式を概念を表 4.2 に示します。

表 4.2 多面体の干渉処理のコマンド(プログラム GEOMAP)

コマンド名	干渉処理の内容	概念式	論理式を使う表現
UNI	二つの幾何モデルの和	$A+B$	$A \text{ or } B$
SUB	モデルAからBと重なる部分を引く	$A-B$	$\neg(A \text{ or } B)$
INT	モデルAとBとの共通部分を求める	$A \times B$	$\neg(\neg(A \text{ or } B))$

5. 演算の約束も言葉で表す

5.1 数式文字並びはグラフィックスである

5.1.1 数式の編集は特殊技術である

科学技術レポートに数式を書いて説明に使うとき、判り易く、また眼で見て整った文字並びに編集するように工夫します。数式に使う文字は、英字だけでなく、ギリシャ文字などもあり、大文字、小文字を使い分け、直立体、斜体、と書き分ける、などをします。数学書、例えば、中学、高校の数学教科書、参考書などで使う文字フォントは、Times New Roman が主に採用されています。数式は、演算子の記号、分数を示す横線、ルート記号の書き方などあって、その全体は二次元的な広がりを持つこともあります。したがって、数式を鉛の活字だけで版組みをしていた時代は、苦勞の多い作業でした(図 5.1)。

二次方程式

$$y = ax^2 + bx + c \quad \dots (1)$$

において、 $y = 0$ のときの解は

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \quad \dots (2)$$

図 5.1 数式はグラフィックスの性格がある

5.1.2 数式を文章に書いて説明する

数式を特殊なグラフィックスであるとする考え方は、コンピュータのOS(オペレーティングシステム)が、オブジェクト指向の扱いをするようになってからです。文字と記号並びだけで表した数式は、声に出して説明することができます。図 5.1-(1)の二次方程式は、その一例です。声の情報は、一次元の情報並びです。しかし図(5.1)-(2)に示した根の公式は、二次元的な並びですので、どのように声に出して言えば、相手に間違いなく伝えられるかを考える必要があります。根の公式は、口調の良い言い方で覚えませんが、式の表し方を思い出す方法の一つであって、厳密に式を再現する目的の言い方ではありません。コンピュータを使って電子的な編集ができるようになって、根の公式のような複雑な式であっても、コンピュータに語りかけて編集処理をする方法が研究されました。この数式編集に取り組んだ最初が、D. Knuth(1938-)であって、そのソフト名が**TeX**です。現在では、パソコンのワードプロセッサに、数式編集のソフト「**数式エディタ**」が組み込まれるようになりました。その中身は、数式をグラフィックスで描きます。ただし、手作業です。

5.1.3 数式の組版は特殊な技能である

図(5.1)が示されたとき、この中の式が、なにを伝えたいかの目的説明が必要です。読者側は、その式をどう解釈するかが問われます。数式を書く側は、変数や定数の定義、などから始めます。変数や定数を表す英字は、一文字を当てます。図(5.1)では英字の説明がありませんが、暗黙の約束で、小文字の $a, b, c \dots$ は定数、 $x, y \dots$ は変数です。記号の意味は、「JIS Z 8201 **数学記号**」にあります。式は、途中で式の合成や分解などの変換があっても、整理した最後の式だけを書くことをします。図(5.1)では、式番号(1)から(2)が得られる過程は省いてあります。この省略ができるのは、数学そのものが非常に論理的な道具であるためです。数式は、一並びの文に書けないこともあります。連立方程式は、幾つかの数式を縦に複数行並べます。マトリックスを表したいとき、同じ種類の変数文字を多く使いますので、数字や他の文字を下付きにして区別し、べき乗は上付き文字で表す、などをします。このような約束を使って、掛け算記号を省きます。割り算は、「 \diagdown 」で書くと文字並びの行数を節約できます。しかし、横線を使った分数形式が標準ですし、見易くなります。括弧も、入れ子の関係が分かるように、種類を変え、また、分数式を含むなどのときは、複数行をまとめるため、高さの高い括弧を書きます。数式を鉛の活字で版組みをするときは、特別な技能を持った植字工が作業しました。したがって、科学技術書を扱う出版社は、独特の存在価値があります。

5.1.4 計算手順の説明は別に組み立てる

図(5.1)示した数式の意味は、数学的には、左辺と右辺とが等しい、と関連づけた**等式**です。また、左辺の値は、右辺の式で計算できる、とも解釈します。左辺と右辺とを入れ替えると、意義的には、左辺で計算した結果を、右辺の変数に代入することと解釈できます。コンピュータのプログラミングでは、図(5.1)-(1)で例示した形式で書いた**代入文**を使います。そのため、実際に数値計算を行わせるときは、右辺の式を先に計算しておいて、左辺の変数に代入します。つまり、計算手順とその実行は、左右逆に行われます。前の第 4.1 節で、電卓を使って数値計算をするとき、キーの押し順が代数式の文字順とは違う、と説明しました。この、逆順に処理させる約束は、プログラミング言語のコンパイラが解読(翻訳)して、実際計算に向くようにオブジェクトコードに組み立て直します。プログラミングで書いた文字並びでは、コンピュータが実行できないからです。

5.2 数式を読み上げて相手に伝える

5.2.1 数式は物理的な意義も考える

代数式を利用して数値計算をさせるとき、変数や定数に数を当てますが、その数は、大きさを持った数値とともに、**次元**(dimension)と**単位**(unit)とが在ります。これらのことを**宣言**(declaration)しておいて、中身の**定義**(definition)を決めます。図(5.1)-(1)に例示した二次方程式は、放物線をグラフに描くことを想定しています。ここで、変数(x, y)は、放物線上の座標です。座標の次元は長さです。次元とは、物理的に言えば、長さ、質量、時間などの組み合わせです。長さの組み合わせには、面積、体積があります。速度は(長さ/時間)の次元です。或る変数を長さとして使うとき、ミクロン、ミリ、センチ、メートル、キロなどの大きさの単位を使い分けます。数式を構成する定数や変数並びは、次元と単位とを整合させる確認が必要です。より技術的な習慣は、なるべく桁数の少ない整数で表し、それに合うような単位系を選択します。図(5.1)に示した定数aは(1/長さ)、bは無次元、cは長さの次元を持ちます。使う数の値が決まって、始めて、数値だけを扱う数値計算、つまり算術を応用することができます。数学が抽象的な学問であると言うことの意義は、次元と単位とを捨象するからです。

5.2.2 数式を文章に表す

数式を文章で説明し、さらに、それを口頭で相手に伝えたいとします。その相手は、人だけでなく、コンピュータも含まれます。数値計算をさせるためのプログラミングを書くことが、文章化の実践的な応用です。元の式の本質を損なわないように新しい数式を提案する、または、式の形を変更することの研究は、計算精度を落とさず、計算時間を短くする目的もあって、数値計算法と言う専門分野が育つことになりました。文章を読み上げる、つまり声に出して伝えても正しく理解してもらえるようにします。文章は次元の情報並びです。図(5.1)の式(2)は、二次元のグラフィックスの性格がありますので、次元並びの文字列で説明するには、工夫が要り、ます。それは、一つの文ではなく、幾つかの文並びの集合に組み立てます。普通の文一単位は、句点で区切ります。プログラミング言語では、種々の書き方の約束があります。C言語では、コロン(;)です。改行や空白(スペース)は無視されますので、物理的には一続きの、切れ目のない文字並びです。これを、眼で見ても判り易くテキスト形式にまとめることが、実用的なプログラム文書の書き方です。ただし、声に出して読み上げるには不向きです。

5.2.3 算盤の読み上げ算

最も原初的な数値計算、それも足し算の連続は、そろばんの「**読み上げ算**」のように、言葉に出して伝えます。電卓を使うときも同じですが、歴史のあるそろばんで説明します。読み上げ算は、読み上げる人と、そろばんを使う人とのコミュニケーションで実行されます。標準的な言い方は「御破算で願いましたは」で始めます。このとき、桁並びをクリアします。計算は、主に整数の足し算の連続で進め、最後に「…では」で止めて結果を書き出します。紙に書いた数値を見ながら操作をする「見取り算」もしますが、耳で聞いて操作をすれば、眼移りをしない分、玉を動かすことに集中できます。また、複数の人が同じ計算をすると、答えの読み合わせをして、結果の正確さを確認することができます。同じ計算を一人で進めるときは、検算で同じ計算を2度することになりますし、結果が同じでなければ、さらに検算しなければなりませんので、時間が余分に掛ります。

5.3 歴史の古い幾何と代数

5.3.1 パソコンは幾何の計算に向いていない

幾何に原理がある代数学の代表は、三角関数です。数値にして幾何学的な性質を理解したいとすると、三つの課題を実用的に解決しなければなりません。第一は、整数で表すことができない多くの問題があって、無理数を実用的な精度にして数値計算に利用しなければなりません。これは、理論にこだわる学者は扱いたくないようです。しかし、幾何に関係する数値計算をコンピュータで処理しようとする、あと二つの課題も理解しておく必要があります。第二の課題は、幾何学を構成する基礎的な変数となる「点、直線、平面」の三つの要素を、どのような数式で定義するか、です。点は、座標系を決めて、座標値で宣言します。線座標、面座標、立体座標から選択します。直線と平面とは、無限の多くの点の集合と考えます。無限の多さは、数値では扱うことができませんし、集合も理解が難しい概念です。第三の課題は、「点、直線、平面」を変数と考えることにしても、これら相互間に代数計算的な算法がないことです。平面幾何学で、二つの直線の掛け算に相当する処理は、交点の計算です。理論的には二つの直線の論理積です。しかし、平行な二直線は、交点がありません。つまり、例外があります。家庭でも使えるようになったコンピュータ(パソコン)は代数計算の装置ですので、幾何学的な性質の計算には向いていません。さらに、幾何の計算に向けたプログラミング言語も商業的に開発されてはいません。科学技術計算では、幾何に絡む計算が多いのです。例えば、テレビの画面で地球規模の天気予報に使うグラフィックスは、有効桁数の多い大量のデータを、高速で処理することが要求されます。ここにスーパーコンピュータの需要があるのです。

99. おわりに

筆者の専門は、橋梁工学です。個人的なことを言うと、敗戦(1945)を経て、戦後の復興事業の一つであった橋梁構造物の、設計から建設までの過程に、多くの実践的な経験を持ったことが幸いでした。その一つが、橋の力学的な構造計算です。橋は、何も無い空間を渡す構造ですが、その上を重量の大きな自動車や鉄道を安全に通行させる目的があります。橋は大きな寸法ですし、費用も高額になりますので、試しに作ってみて、具合が悪ければ作り直す、言わば試行錯誤が許されない厳しさがあります。古典的な力学原理を応用して設計し、架設工事までを計画しなければなりません。設計作業のかなりの部分は、材料寸法を合理的に決定するための数値計算です。戦後すぐはコンピュータを利用する数値計算が利用できない環境でした。精度のよい数値計算をするには、機械式の計算機、さらには電動化した計算機を使いこなす技能が必要でした。パソコンは、現在(2015年の時点)までに大きく進歩、発展してきましたが、数値計算技法の基本には、依然として手計算、碎けて言えば算術、の経験が必要です。単に算術と言うと、小学校の低学年で習う初歩的な四則演算を考えると、コンピュータを巧みに利用する数値計算の技術も、算術です。これを、教育目的を考えて見直しを考えてみました。「初歩的な算術とはいくらか違うよ」と言うことを考えて、「易しくない…」の形容詞を表題に付けました。