

人の言を通す  
通信ひとわり

達哉ん

2018年6月執筆開始

## はじめに

気づけば、C言語の教科書を書いてから5年、問題集を書いてから2年が経とうとしている。"弟子"として1から教えた若人も手を離れ、弟子も他の知己も含め、時折尋ねに来た時に応ずるばかりとなった。今は、新たに学んだ日本茶でも淹れてほっと一息つきながら、最近はじめた落語に興じている日々である。

変わったのは何もプライベートばかりではない。時はすべてを連れて行くもので、仕事も住まいも知人も大きく変わった。それは、弟子や知己からの質問も例外ではない。ネットワークやシステムに関する質問が増えたのだ。仕事もまた然り。面白いことには、質問も仕事も、ネットワークは抽象化した理論をお話することが主で、システムは個々の具象を扱うことが多いというのが、共通していることである。

個々のシステムについては、各々をマニュアル的に記すことで満足な仕上がりが達せられる。DNS,LDAP,DHCP,POP,SMTPなど各々のサービスの設定は言うに及ばず、ルータの設定コマンドなども、マニュアルで対応すれば良いところであろう。だが、それらの設定の意味を理解することまでは、マニュアルに求められまい。殊に、汎論の範疇と呼ぶべき部分までマニュアルに記載しているようでは、あまりに冗長となってしまう。また、ネットワーク一般の理論もマニュアルに記すべきものではない。そこで、これらネットワークの汎論を統べる役目の一冊を用意したいと考えた。周辺の方への説明の際などに便利でもあろうし、無論、自身の記憶を掘り起こす良い復習の契機にもなろう…こうして、この本を著すこととした。

新たに記すときに悩むのは、その性格付けである。コンピュータネットワークの世界には、大部の聖典もあればよく纏まった入門書もある。同じ方向性で書いたとしても、陳腐なものにしかかなりはしない。いかなる性格付けにするか、その悩みに鍵をくれたのは幸いに私を慕ってくれる優秀で賢明なる若き友であった。彼らの根問の中で、私は文字に象られている通りの「人に言を通す」という通信こそが大切であると考えたのである。口伝から始まり現代ではインターネットで多くの通信が行われる。その中で、私は非常に多くの通信方法を経験し、また利用している。趣味で演る落語は口伝が基本だ。生で見る落語は、肉声か、あるいはマイクを通す程度である。一方、レコードやCD、DVDで落語を見ることもある。手紙を丁寧に書いて送るのは実に楽しいもので、時折私書が郵便箱にあるとそれだけでとても嬉しい。一方、友人とのやり取りは電子メールやSNS、チャットアプリなどが主力となっている。これだけ多く「人の言を伝える」ことに接する現代、まずは口伝、それから手紙と俯瞰し、そこからアナログ通信・デジタル通信へと発展させていくのはどうだろうか、と考えた。

といっても、演劇の声の出し方や手紙の書きかたなどを丁寧に書くものではない。落語の稽古風景や郵便制度の話を詳しく書くのは土俵が異なるだろう。そうではなく、各分野の経験を元に、「人に言を通す」ことを掘り出して話す、そんなテキストを執筆することとした。歴史に通信を学び、そこから電気通信はどうなっていたのか、どのような必要性で技術が生

まれたのか、経験から考察を加えて執筆していこうと思った。自身の学んだ電気通信を、経験に従って解釈しなおし、書き直す。地味であり、新規性も乏しいかも知れない。だが、何より話好きな自分の教え方と書くテキストに一貫性を入れるなら、言を通す、それを置いて他になかろう。オーソドックスかもしれないが、様々な分野の経験が何がしかのシナジーをおこし、説明に彩りを加えてくれればと思う。

本書の導入として、我々人類が物事を伝えるために用いた手段には如何なるものがあるのか、それは現代でどのように生きてあるいは滅びたのかという章を設けた。電気通信と程遠く見えるこれらの伝達には、しかし電気通信の要件を定めるような人類のアイデアが詰まっている。そのアイデアの最も元となる部分には「記録」があり、第I部ではその記録技術や信号処理技術についての最小限を記した。ここで記録された情報をまずは電氣的に伝えることを考えるのが第II部である。ラジオ・電話・FAXと言った現代の「デジタル信号」からすると一代前の原理であるが、これを知ることはTCP/IP世界にも関わる低レイヤー技術の理解に十分に役立つと考える。第III部からはいよいよ本丸、コンピュータ・ネットワークへと論を進める。階層化モデルに従った理解と代表的なプロトコルを、各々第III部・第IV部として本書の大部分を用いて解説する。付章的であるが、第V部として、冗長化・セキュリティ・トラブルシューティングなど、構築や管理といったより実践的な情報の基盤となる、最初の一步を記述した。

各章の記述においては、多少の数学を要求する箇所もあるが、高校程度の数学が取り扱えれば読めることを基本とした。このため、前提知識の解説を随時【補足】と記して用意した。一方、厳密な取り扱いが本論にとっては不要であったり、あるいは難解である場合、その部分についてはまず厳密な取り扱いをせずに論を展開し、後の節で詳細を補うこととした。この詳述部分には【補遺】と記し、読み飛ばしても大局に影響しないよう心がけた。また、汎論に対する事例の紹介という意味付けで、各章で出来る限り演習問題を出題し、巻末にその解答を記述した。汎論の解説ではあっても、手ずから具体例を解釈することは理解の深化に繋がると信ずる。

最後に、この本・これまでの本や、著者と付き合ってくれたこれまでの友人たち、付き合いこととなるこれからの読者たちにもお礼を申し上げたい。南天竺には赤梅檀という立派な木があり、その周には難莚草という草が蔓延ると聞く。この2つの植物は”有無相持ち”、赤梅檀の降ろす露が難莚草に双無き水となり、難莚草の盛衰は赤梅檀に栄養をもたらすという。友人と自身、読者と著者…携わる人々がそんな関係であることを祈り、この本を捧げる。

桂米朝師の「伝」に畏敬を払いながら  
達哉ん

# 目次

第0章	「伝える」歴史	1
0.1	声による伝達	1
0.1.1	人間の発話	1
0.1.2	発話の制約	1
0.1.3	口伝という方法	2
0.2	筆写による記録	3
0.2.1	文字と図画	4
0.2.2	写本から複写へ	5
0.2.3	筆記媒体が捨てたもの	5
0.3	遠距離の高速通信	6
0.3.1	持ち運ぶことの難点	6
0.3.2	視覚による通信	6
0.3.3	視覚の通信が捨てたもの	8
0.4	通信の秘匿	8
0.4.1	未開封証明	9
0.4.2	暗号化	9
0.5	記録・通信媒体の増加	10
0.5.1	増加した媒体	10
0.5.2	そして電気通信へ	11
第I部	情報の記録	12
第1章	デジタルデータの記録	14
1.1	二元状態とデジタルデータの特徴	14
1.1.1	記録の条件と二元状態	14
1.1.2	デジタルデータの特徴	15
1.1.3	【補足】2進数・記数法について	15

1.2	情報の符号化 .....	17
1.2.1	整数の表現 .....	17
1.2.2	文字の表現 .....	19
1.2.3	モールス符号 .....	19
1.2.4	小数の表現 .....	19
1.3	符号化効率の改善と圧縮 .....	22
1.3.1	符号化効率の改善 .....	22
1.3.2	可逆圧縮・不可逆圧縮 .....	23
1.3.3	エントロピー符号 .....	24
1.3.4	辞書式 (ユニバーサル) 符号 .....	27
1.3.5	差分 (デルタ) 圧縮 .....	29
第 2 章	アナログデータの記録 .....	31
第 II 部	電気伝送の方法 .....	32
第 3 章	電気信号の同期 .....	34
第 4 章	変調 .....	35
第 5 章	伝送品質の担保 .....	36
第 6 章	伝送路の方式 .....	37
第 III 部	コンピュータ・ネットワークの基礎 .....	38
第 7 章	通信プロトコル .....	40
第 8 章	ネットワークの構成 .....	41
第 9 章	ネットワークアーキテクチャ .....	42
第 10 章	ネットワーク機器 .....	43
第 11 章	Layer1 : 物理層 .....	44
第 12 章	Layer2 : データリンク層 .....	45

第 13 章	Layer3：ネットワーク層 (1)IP とアドレス	46
第 14 章	Layer3：ネットワーク層 (2) ルーティング	47
第 15 章	Layer4：トランスポート層	48
第 16 章	Layer5～7：上位 3 層	49
第 IV 部	ネットワーク・プロトコル	50
第 17 章	ネットワークモデル・再論	52
第 18 章	telnet	53
第 19 章	DNS	54
第 20 章	DHCP	55
第 21 章	FTP	56
第 22 章	HTTP	57
第 23 章	メールに纏わるプロトコル	58
第 24 章	認証に纏わるプロトコル	59
第 V 部	安定通信網の構築と運用	60
第 25 章	冗長化	62
第 26 章	セキュリティ	63
第 27 章	トラブルシューティング手法の基礎	64

## 第 0 章

# 「伝える」歴史

現代で通信という言葉聞いた時、連想される多くはインターネットを始めとした様々な電気通信だろう。だが、電気通信ができる以前より、人は様々な意思疎通…通信を行ってきた。本章では、電気通信に至るまでの人の意思疎通の営みを見ていく。これらの営みの中で、人間は通信に必要とされる要件を見出してきた。通信とは何か、何が必要なのか。まずは歴史からその要件を学び取っていきこう。

## 0.1 声による伝達

動物には種によって様々な対話方法がある。蝙蝠なら超音波を使い、鳥ならば鳴き声で対話を行う。動物の鳴き声のモノマネで知られる江戸家小猫氏によれば、人間に最も近いゴリラなどは 10 種類の鳴き声により様々な対話を行うという。例えば、我々がゴリラの鳴き声と聞く「ウホウホ」という鳴き声は、特に喜ばしい時、若いゴリラが使うことが多い表現なのだそうだ。我々人間にその差異はなかなかわかりづらいが、しかしながら種ごとの鳴き声による伝達は、我々人間の発話による伝達に同じことなのだという。

### 0.1.1 人間の発話

自らの声帯により何らかの音を出し、これに意味を持たせて伝達する。我々人間が知る多くの動物の鳴き声にはそれほど多くの種類の伝達はなく、概ね自らの感情の伝達が中心となっているようではあるが、人間の発話による伝達もそれと同様である。生まれてすぐの赤子は、泣くという動作を伴った発話により自身が不快であることを伝える。成長し、感情が分化する(図 0.1)につれ、何が不快なのかなどもその泣き方に入ってくるようで、親はその泣き方で状況を察することも少なくない。やがて、言葉を覚え話すようになると、それらの言葉の意味を経験から学ぶようになる。家族から広がるコミュニティの中で、共通した意味の言葉を使うようになり、人との伝達・コミュニケーションを始めていくのである。

コミュニティの違いによる差異こそあるものの、我々人間の”鳴き声”は言語を為し、その共通認識により対話、伝達が始まったのである。この言語は感情にとどまらず、具象抽象の垣根を超えて叙事や論証、または歌や言葉遊びと言った様々な知的活動の礎となった。動物の鳴き声による対話と対応した人間の会話は、最も原始的な伝達の方法であるものの、様々な技術が発達した今なおコミュニケーションの基礎を為している。通信の始まりは、このような”発話による伝達”であったといえよう。

### 0.1.2 発話の制約

しかし、単純な発声による発話では、声が届く範囲にしかその意味が届かない。この時の届く範囲とは、空間的な範囲もあれば時間的な範囲でもある。落語会が行われていた会場に、

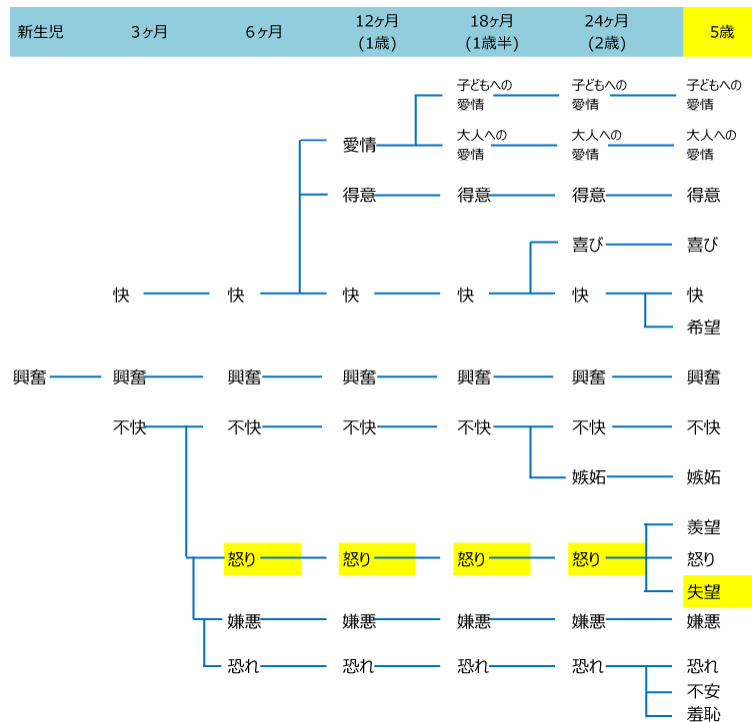


図 0.1: ブリッジスの情緒分化図 日本病児保育学会ブログ記事より引用

終わった後に入っても落語を聞くことはできないし、また、同じ時間に遠く離れた場所においても落語を聞くことはできない。肉声の落語を聞くことができるのは、その時その場所で演者と同じ空気を共有していた者に限られる。逆に、その空気を共有している者の中では、選択的に伝えたり聞いたりするというのも難しい。高座の上で演者が下座に何らかの指示を出す時、声を使うとすれば客席の幾人かにはどうしても聞こえてしまう。同じ会場で鼾をかいて眠る客がいたとして、これだけを聞かないというのもなかなか無理がある。

先の鼾の例のような雑音があった場合などは特にそうだが、発話によるコミュニケーションでは聞き取れない・追いつかないと言った問題も発生する。マンツーマンのゆっくりした対話であれば言い直してもらえばすむだけの話であるが、先の落語のような例では、そういうわけにも行かない。音自体が聞こえなかった、音は聞こえたけれども判別がつかなかったと言ったレベルから、言語的な問題…イントネーションの差異、語彙にない単語の使用、そもそも言語が異なる、勘違い…と言った知識や思考の問題まで、発話の時には伝達の誤りが起こりうる。

発話による伝達は、原始的であるがゆえに手軽ではあれど制約や問題も内包している。通信・伝達技術というのは、これらの制約や問題を如何に打破するのかという人間の飽くなき挑戦の成果であるとも言える。

### 0.1.3 口伝という方法

発話の制約を最も単純に外したのは口伝である。現在でも民話や民謡など、様々なものが口伝により伝えられているが、これは人間の記憶を介することによって時間や空間の制約を緩和したものといえよう。最も身近なところでは、伝言がそうであろう。もっとも、伝言ゲー



ムに見られる通り、伝達の誤りという問題は解決されているとはいいがたい。「JPCZ による擾乱の影響で山陰地方を中心に荒天となる」などと専門の用語が入った言を伝言したところで、途中に知らぬ人が入れば伝言がうまく行く可能性は低いだろう。また、伝言には意識的・無意識的な取捨選択もあり、伝達の誤りという観点ではむしろ増えているかもしれない。

しかしながら、人間以外の資源が必要ないこの口伝という方法は、最も初期から存在すると同時に、現代まで続いている伝達方法である。口伝の国内における最も古い例としては、稗田阿礼が挙げられる。古事記には彼(彼女?)の誦により伝えられたものが筆録されている。その後の歴史にも、説法や講釈と言った例はほぼ口伝であったようだ。一方、現代で行われている口伝の例としては、先にも持ちだした落語を例に挙げたい。昭和の爆笑王、桂枝雀師は著書「枝雀とヨメはんと七人の弟子」の中で次のように書いておられる。

”一番最初は「ちょっとやるから聞いてなさい」と言って、一字一句口移しで教えることから出発いたします。やらせてみて、ちょっと違うとイントネーションから直す。それが段々お稽古やっていくうちに、五分なら五分、流れを教えるようになる。”

これは、平成のはじめ頃に書かれた文であるし、枝雀師は99年にお隠れになったから、いま他の噺家が同様のやり方をしていると言い切ることはできない。だが、筆者が、この枝雀師の三番弟子、文之助師に”つる”を教わった時は、ほとんどこの通りの教え方であった。ただ、カルチャースクールでのことであるから、録音し、自分で原稿に起こし、その上で直してもらいながらという指導であった。とはいえ、これも口伝には違いない。多くの伝達法がある中で、わざわざこの原始的な口伝という方法で伝承をするのはなぜだろうか。

それは、口伝によってしか伝えられない、空気や情感、発声、身振り手振りの細かな違いなどを伝えられるというところであろう。現代の技術をもってすれば、双方向で似たような指導を遠隔地間で行うことも可能であるように見える。しかし、それでも微妙な空気感や息の間といったものは伝えきれない。落語という芸能は高座の落語家と客席が一体となって作り上げる空間が生命であるから、なるほど、その部分を伝えるためには口伝を取るよりほかないのであろう。また、落語は時代により変化していくものであるから、口伝という形態での変化は逆に追い風となるのであろう。これは、落語の伝承が「ネタの原稿を伝える」のみでないことを示しているといえる。

現代の視点からすれば口伝は古臭く、非効率な手法に見えることも多い。だが、それを意図して取り込んでいる芸に学べる通り、口伝でこそ伝わることもある。通信・伝達技術の多くは、重要と考えるににくいものや伝えづらいものを取捨選択しているのである。逆に、その捨てられたものを拾う価値があるシーンや、そこまでの効率を要求しないシーンがあるからこそ、一見古臭く見える通信・伝達の方法も現代に息づいているのである。

## **0.2** 筆写による記録

---

口伝は人間の記憶を媒介にして時間や空間の制約を打ち破った。しかしながら、当然全ての人間が稗田阿礼のような記憶力を持つはずもないし、伝達も記憶も個人の状況に依存することとなる。これを打破するのは日本へと輸入された文字、そして筆記媒体であった。

### 0.2.1 文字と図画

文字とそれを記録する媒体は、ヒエログリフやパピルスと言った著名な例があるとおり古代エジプト文明の時代にまで遡る。楔形文字や漢字と言った様々な文字が青銅器時代に生まれたのである。日本へ文字が入ってくるのは弥生時代とされ、それからの後、8世紀頃には先の稗田阿礼の言を元にした古事記(図0.2)や風土記、日本書紀といった書物が発行された。

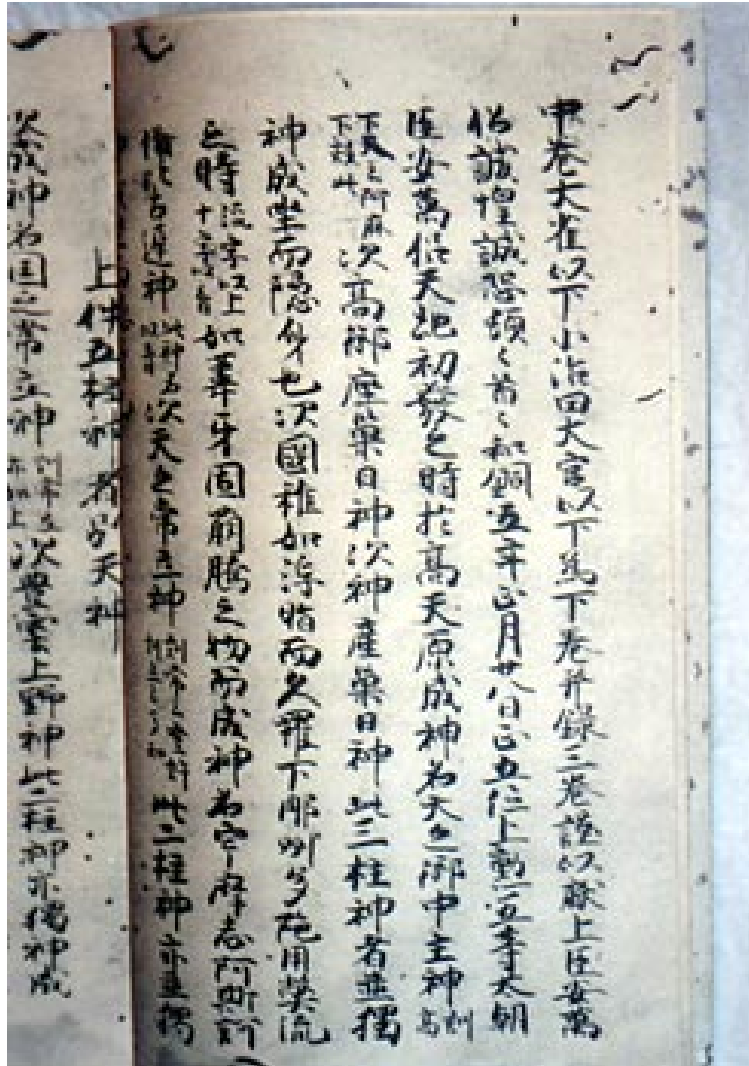


図 0.2: 真福寺収蔵の国宝・『古事記』 Wikipedia より引用

文字の輸入に伴い、離れた場所に文字を書いた媒体を送るという形式での伝送も行われるようになっていった。それらの書物の中には何百年という時を経てなお現代で読むことができるものも多い。これは、文字とその記録媒体が時間と空間の制約を打ち破ったということである。著名な数学者の書簡や日記が見つかることもあれば、伝わるつもり無く詠まれた”この世をば 我が世とぞ思う 望月の 欠けたることも なしと思えば”が現代にまで伝わっている(道長本人は記録に残さなかったが、その場に居合わせた人が日記に記したために現代まで伝わっている)など、文字は言語によって表現可能なものの時間的・空間的制約を打ち破ったのである。汚損や焼失といったリスクこそあるものの、口伝や発話の段階からすれば大きく進歩したと言えるのは明らかであろう。

また、これらの媒体は図画の伝達も可能とした。これは、発話や口伝で伝えられなかったものを伝えられるという利点をもたらした。紙媒体は、文字を媒介に発話を記録すると共に、図画として描けるものも記録する媒体なのである。

### 0.2.2 写本から複写へ

---

主として紙媒体に記述された文字による通信は、複写されて更に伝えられることにより、口伝やうわさ話と同様に広く伝わる手段となった。写本はその最も原始的な例である。紀元前3世紀、アレクサンドリア図書館では既に組織的な写本が行われていたとされる。別には、ギルガメシュ叙事詩もその頃の写本を最初とし、原本こそわからぬものの現代まで継がれている。日本においても多くの写本がある。特に源氏物語などは、既に写本が遠距離の伝達の役割を担っていたことが、菅原孝標女が更級日記に記述した例などからわかる。

先に述べたとおり、筆記にまつわる媒体には確かに汚損や焼失というリスクがある。これ以外にも、数十年・数百年という時の流れの中で退色するようなものもあれば(現代に流通しているボールペンインクの多くや、万年筆のインクなどは必ずしも長期保存に適するものばかりではない)、言語や文字が減びて内容を理解できないというものもある。しかし、複写による流通は、これらのリスク解決に対し、“冗長性”という解決策を与えた。つまり、写本の1冊が残れば伝わる緒は紡ぐことができ、翻訳により別言語へと訳されていけば減びた言語を用いずとも内容を理解することができる。

現代においてバックアップと呼ばれる手法も、原理としては複写と同等である。その観点から言えば、複写は多方向への同時期の通信・流通を可能にしたのみならず、その安全性において起源的なものといえる。紙と図画・文字による伝達は、複写によって通信の時間的・空間的制約を打ち破るとともにその冗長性などを(意識的かどうかは不明ではあるが)確保したのである。

無論、人間の手による複写は手間であるし、また誤りもある。そのため、短時間で多くの複写を行う印刷技術が発達した。8世紀には既に木版印刷の技術が生まれていたようで、その後も活版(凸版)印刷、銅版(凹版)印刷、平版印刷と多くの印刷技術が追隨していくこととなる。広い意味では、印鑑などもその一例とも言え、承認の意を当人しか原版を持たない複写によって表しているとも考えられる。このように、複写・印刷の技術は、通信の幅を大きく広げた、その代表的なものと言える。

### 0.2.3 筆記媒体が捨てたもの

---

一方、筆記媒体・印刷媒体は視覚によって通信できないものを取捨選択している。音や匂い、手触りや味といったものを、筆記媒体・印刷媒体のみで完全に送ることは出来ない。勿論、これらが必ずしも必要だとは限らない。ただ、筆記媒体は、図画と言語によって伝えられるもののみを選択し、結果として他を捨てることで、通信に様々なメリットを生み出したのである。

具体的な例を見てみよう。先に、ゴリラには10種の鳴き声があると書いた。これらの鳴き声を、我々は正確に伝わるように書き分けられるだろうか。あるいは、その各々の声のシ

チュエーションを解説するとき、十二分な語彙・表現があると言い切れるだろうか。おそらく、この間に Yes と答えられる人は少数であろう。では、この鳴き声の差異が現実に必要なような例はあるだろうか？その道の専門家でもない限り、これを文字で正確に伝えなければ困るというケースは考えにくい。十分な語彙・表現と図画を以って伝達するとき、筆記媒体が捨てた要素は確かに存在するのであるが、一方でその要素が必要なケースでなければ、筆記媒体による通信によりその利便性のみを享受できるのである。

## 0.3 遠距離の高速通信

---

筆記媒体によって破られた制約は多いが、しかし、筆記媒体にも難点があった。通じるのに時間がかかるという点である。

### 0.3.1 持ち運ぶことの難点

---

郵便碁という遊びを考える。現代において、遠く離れた場所にいる二人が囲碁を打ちたいとすれば、インターネット上のゲームサイトにアクセスするなどすれば、ほぼリアルタイムで遊ぶことができる。しかし、それがなかった明治の頃、ハガキに一手ずつ手を書いて送る“郵便碁”が生まれた。現代の郵便制度を持ってしても、1局に1年から2年はかかる。通常は一度に4つの手を送り、4局同時に進めたそうであるが、それでも非常に多くの時間がかかる。日本郵便碁愛好会に所属されているようなじっくり楽しみたい方はともかくとして、遠方の友人と会話がてら軽く打ちたいという時に気軽に楽しめるとは言いづらい。

郵便制度の発達の前には伝書鳩や飛脚により同様の伝達が行われていた。しかし、時間がかかるという点はいずれも変わらないようである。明石飛脚という噺では足に自身のある男が大阪から明石までの約15里(現代で言うと約59km相当)の道のりを一日かけて片道だったという。無論、交通手段が未発達であったという時代背景もあろうが、現代でも郵便は大抵遅い手段として知られる<sup>1</sup>。

### 0.3.2 視覚による通信

---

マラトンの戦いの例がもっとも古いものとして知られるが、戦況を知らせるような伝令は内容こそ多くなくとも速度が要求される。そこで、視覚的に伝えられるものに何らかの意味を定めておき、遠距離でも素早く通じるようにした。狼煙や手旗信号(図0.3)がそれである。現代でも、野球でベンチから監督が出すサインはこれに当たるだろう。また、筆記媒体も介していることとなるが、アドバルーンもその一例と言える。

これら視覚による通信は視線を確保することが要求されるが、通常人間が見える範囲においてはタイムラグなしに通信を行うことができる。また、中継も比較的容易い。手旗信号や筆記媒体など、細部を見る必要がある手段だと距離こそ短くなるものの、伝達にかかる速度自体は高速で、音では届かない距離でも伝達できることには変わらない。一方、その意味付けの少なさから、多くの情報を伝えるには不向きである。

---

<sup>1</sup>これは非常に特殊な例であるが、10TB容量のHDD400台を詰めた箱を2日かけて車などで運べば、185Gbpsという驚異的な速度になる。読み書きの時間は無視しているし、それだけ大容量のデータを運ぶ必要もないだろうが、データ量が著しく多い場合は郵便が最速になることも考えられる例としてここに記した。

# ■手旗信号によるカタカナ

※① は白旗をあげる。

123			123			123		
ア			カ			サ		
イ			キ			シ		
ウ			ク			ス		
エ			ケ			セ		
オ			コ			ソ		
123			123			123		
タ			ナ			ハ		
チ			ニ			ヒ		
ツ			ヌ			フ		
テ			ネ			ヘ		
ト			ノ			ホ		
12		12		12		12		
マ			ヤ			ル		
ミ			ユ			レ		
ム			ヨ			ロ		
メ			ラ		長音		半濁音	
モ			リ		濁(だく)音			

図 0.3: カタカナの手旗信号 海上自衛隊・海の手帳より引用

### 0.3.3 視覚の通信が捨てたもの

---

視覚による通信は、速度を重視したがために、伝えられるものが非常に限定的なものとなった。手旗信号や狼煙であれば、前もって決められた意味付けの範疇に属するものしか伝えられないのである。また、狼煙は多くの場合一方向的でもあった。逆に、双方向的にも可能な手旗信号は、音やものに比べ受信者が気づきづらいか、あるいは受信者を常にさく必要があった。

インターネット技術の標準化を推進する任意団体 **IETF** (Internet Engineering Task Force) は、その標準技術仕様等を RFC (Request For Comments) という形で公開しているが、この中の RFC4824 では、手旗信号を用いた IP データグラムの転送を記している。これはジョーク RFC として出された仕様ではあるが、手旗信号について（電気通信を元に）実に愉快的考察がなされている。以下、例を挙げてみよう。

- 送信・受信に各 1 名以上ずつの、2 名以上の組によりひとつのインターフェイス (送受信設備：この場合は人) がなされる。
- 通信速度は送受信者の経験に依存する。
- なりすまし防止の為の自己紹介が推奨される。
- タイムアウト、通信が時間切れとなる感覚はデフォルトで 60 秒で、手を休めることができる程度になっている。
- 転送量が多く長時間の通信となる場合は代替のインターフェイスを用意する必要がある。
- 時間とともに転送速度 (帯域幅) が変わり、概ね減少していく傾向にある。
- リンク内に別のインターフェイスがあった場合受信できるため、セキュリティ的に安全ではない。
- インターフェイスが送受信したデータを記憶する必要があり、生存時間が定まっていない。
- インターフェイスを通過したデータであっても、その記憶に残っていることがある。

これらの点は、現代の電気通信からすればその多くが特異なものであり、実用上はほぼデメリットと捉えられるものであろう。換言すれば、これほどまでに多くのものを犠牲にしても、何よりも速度が必要になる時、初めて手旗信号など視覚の通信に頼る必要が出てくるのである。

## 0.4 通信の秘匿

---

先に書いた藤原道長の歌は、本人の意図によらず後世へと伝わったものである。この歌であれば（元々が酒席等の話であったようなので）笑い話で済むが、意図しないところに伝わるのは望ましくないことも多い。対面の方法であれば、本人であることを認識するか、ある

いは初対面であれば本人確認をするなどの方法で当人同士とし、他に誰もいない場所で(音漏れがなく、盗聴器なども仕掛けられていない前提であるが)話すなどの方法がある。だが、遠距離の場合、送受信者双方の目から離れる時間が必ず存在する。その時に、秘密を守るため、やはり様々な手法が用いられてきた。

#### 0.4.1 未開封証明

中世のヨーロッパで主に用いられていた方法が未開封証明、封蝋(シーリングワックス、図0.4)である。蝋を垂らし、多くの場合は家紋などを象った、当人オリジナルの封印を施して郵送した。洋形封筒の場合、このシーリングワックスを破損することなく開封して再度元のとおりに戻すのは至難の業である。したがって、全きシーリングワックスの未開封の書状が届くことは、途中で第三者に読まれあるいは改ざんされていないことの証明となった。



図 0.4: 筆者の封蝋の例

だが、これはあくまでも手紙が届いた場合の話にしか過ぎない。その内容を読むことを目的とする場合、書状を手に入れてしまえば内容は平文であるため、易々と読むことができる。本来の受信者に渡す必要さえなければ、そのまま内々に破棄してしまうことで受信者には何もわからない。これは、現代の郵便制度でも同様であり、心ない郵便局員が一般郵便を持ち帰り、内容を読んだり届けなかったりしたという事件もあった<sup>2</sup>。

#### 0.4.2 暗号化

そこで、途中で奪われたり盗聴されたりしてもその内容を把握できないよう、暗号が用いられるようになった。

古来の暗号として特に有名なのはシーザー暗号である。古代ローマの軍人、ガイウス・ユリウス・カエサル(英語読みでジュリアス・シーザーとなるため、これにちなんでシーザー暗号と呼ぶ)は自身が秘匿としたい文を送るとき、それぞれの文字を”シフト”して送信したという。例えば、aをcに、dをfに、xをzに、yをaに、zをbにと、それぞれ(サイク

<sup>2</sup>少なくとも日本においてはこれが事件として報道される程度に珍しいことであるので、郵便が信用ならないと言っているのではない。寧ろ日本の郵便送達率は9割を超え、世界に冠たる信頼の郵便制度と自負すべきところである。

リックに見て) 2文字後のアルファベットへと変えて送信したのである。この規則を知らない(また気づかない)者には、内容を把握することは出来ない。当時の教育水準や、カエサルに敵対していた者の教養程度からして、これは効果的な手法であったと思われる。

一方、暗号化がうまく行かなかった例もある。世界大戦中の日本軍は、早口の薩摩弁で伝令を行うことにより、当地を故郷とする者にしか内容がわからないよう(即席の、ある種の暗号として)秘匿していたという。ただ、これは敵軍に捉えられた捕虜に読ませることにより、本来読まれるべきでない敵軍にも内容が伝わってしまったようである。

このように、奪われないこと・途中で第三者が読まないことを必ずしも期待できない状況においては、難読・容易に内容を把握させないようにすることのほうが重要となっていた。今日の電気信号においても、暗号化は非常に重要な役割を果たしている。

## 0.5 記録・通信媒体の増加

---

ここまでの例を見てもわかる通り、記録媒体や通信媒体は時代とともに増加してきた。個別にその例を全て論ずることは出来ないが、いくつかを取り上げてみよう。

### 0.5.1 増加した媒体

---

先までの例の後に増えた媒体は、特に光を用いたものが多い。

写真(ここで言うのはアナログ写真・銀塩写真)は、受光によって化学変化を起こす物質を用い、受光状態を保存する媒体である。実際には、受光の記録とその定着に当たる現像の工程を経ることとなるが、絵よりも手軽かつ多くの場合精密にその光景を記録した。これを連続的に撮影したものが初期の動画である。だが、この方法では絵のみしか送ることが出来ない。そこで、その内容は別の台本にするなどし、活弁士と呼ばれる人々が画に合わせて内容を語っていたという。これが今日で言う映画の原形、活動写真(無声映画)である(同じものを絵に変えた紙芝居はそれより以前からあった。)

レコードは盤面に音=振動に応じた溝を掘ることにより録音を可能にした。これが音をそのまま伝達できる最初の記憶媒体であった。これもまた先の写真と同じく、その状況—ここでは音—をそのままに記録するという手法であった。このアナログな録音方法は、本稿を書いている2018年現在、再び売れ行き100万枚に振り返り、およそ30年ぶりに生産が再開される会社も出てくるなど、見直しが進んできている。その主たる論調は、ここまで学んできた事実「通信は何かを取捨選択して他の制約を破ってきた」という中での”取捨選択”の部分捨てをすべきでないというものである。デジタルの音とアナログの音は違う。同様にデジタルの写真とアナログの写真は違う。実用の観点からは捨てられた部分だったかもしれないが、それを価値として見出すことによって、これらアナログの手法は今日でも利用されていると言える。

記録媒体を上げてきたが、通信媒体で言うとモールス信号が挙げられる。これは2つの長さの音あるいは光によって構成され、いくつかの音によって文字を伝達する手法である。ここまでであれば手旗信号と変わらないように思えるが、光源を用いたモールス信号は手旗信



号の使えない闇夜でも利用できるし、人にかかる負担もそれほどのものではなかったようである。

このように、記録媒体や通信媒体が発達し、多くのやり取りができるようになったのが、概ね 1800 年台から 1900 年台であった。その頃、人々はこれまでの火や水に加え、電気の利用を知る。多くのものが電気にとって代わられる中で、通信もまたその波に乗った。この同じ時代は電気通信の幕開けでもあった。

### 0.5.2 そして電気通信へ

---

次章からいよいよ本題の電気通信であるが、歴史に学んだことを振り返っておこう。

口伝えから始まった通信は、以下のような様々な要求の度合いに応じて伝達事項を取捨選択し、種々の発展を見せていった。

- 共通認識による正確な伝達 (使う言葉の認識や狼煙の意味付け)
- 時間や空間を超えた伝達 (様々な媒体による記録)
- 通信の速度への要求 (光を用いた通信)
- 同時の複数の通信 (印刷技術や郵便碁の多重化など)
- 伝達に気づくということ (手旗信号で捨てたもの)
- 秘匿性 (封蝋やシーザー暗号)
- 冗長性 (記憶に頼らないということ、印刷技術)

電気を用いた通信では、これらの考えや要件をどのようにに取り込み、実現しているのだろうか。

電気通信の技術も、ここまでに学んだ通信と同様多岐にわたるものである。だが、通信とは字に書いたとおり、人の言を通じさせることであり、電気を用いたとしてもこの基本に変わりはない。個々の技術の理解の根底に、これら「伝える」要件を見出すことで、技術が活きるのだと信じる。電気通信は現代に発展した技術であり、あまり枯れているようには見えないかもしれないが、人類の経験と歴史に基づいた大いなる叡智でもある。このことを常に意識し、これから電気通信のお話にお付き合い願う次第である。

### 演習問題

---

- 1 歴史的なシーザー暗号では、アルファベットを 3 文字後ろにずらして (a,d,x,y,z をそれぞれ d,g,a,b,c に変えるなど) 暗号化していた。1024 文字以下 1 行の半角文字列が入力されるとき、それをシーザー暗号化・復号化するプログラムを作成せよ。
- 2 糸電話もひとつの通信の方法である。糸電話の通信にはどのような特徴があり、何を取捨選択したのか、本章の他の方法を見た時と同様に考察してみよ。

# 第I部

## 情報の記録



## 第 1 章

# デジタルデータの記録

電氣を用いない通信から見てきた通り、通信と記録は密接な関係にある。また、記録の際には記録すべき要素の取捨選択が行われる。人間が直接運ぶにせよ光や音を使うにせよ電氣通信にせよ、記録されたものを受け渡しするのが通信であるということに立ち戻れば、記録と通信は表裏一体の関係にあることは明らかであろう。

本章では、離散的な「デジタルデータ」をいかにして表現・記録するかという点について学んでいく。

## 1.1 二元状態とデジタルデータの特徴

伝言ゲームでは、なかなか情報が正確に伝わらない。この例を見てもわかる通り、通信は記録と伝達手段(伝送路)の双方において正確・確実であり、途中で内容が変化しないことが求められる。

### 1.1.1 記録の条件と二元状態

記録、すなわち表現において正確性や確実性を担保するためには、次のような条件が必要となろう。

- いくつかの状態に明確に分かれることあるいは定量的であること。
- 内容によらず表現方法が統一されていること。
- 伝送や処理に適した、単純な形態であること。

この条件を満たす最も端的な表現として、二元状態：明確な二つの状態でそれ以外の状態のないものがあり、実際のデータ伝送において広く用いられている。通信において使われる二元状態は、2進数(0/1)のほか、論理(T/F)、極性(+/-)、電流の有無、スイッチのon/offなどが代表的だが、センサの通信などにも目を向ければ動静(動きの有無)、音声(声の有無)なども通信に使われている二元状態といえよう。この二元状態が通信に使われる理由としては、以下のようなものが挙げられる。

#### 二元状態の通信における利点

- はっきりと状態の区別がつき、検出も容易で、誤りも生じにくい
- 多くの電氣的状態に一致し、情報の伝送や記憶に有効である
- 数字の0/1に対応させて2進数表現することで四則・ビット演算を適用できる

### 1.1.2 デジタルデータの特徴

デジタルとはよく聞く言葉であるが、この訳語「離散量」あるいは「計数」はそれほど馴染みがない言葉かもしれない。デジタルというのは幾つかの、飛び飛びの状態を取る量という意味であり、対してアナログは連続的に値が定まる。例えば電車の線路が始点からどれだけの距離にあるかはアナログであるが、現在居る駅という情報はデジタルと言える。針が連続に動くアナログ時計は（現実に取り出れるか、その精度はどうかというのは別にして）10.232秒など小数点以下の秒数も表しているが、デジタル時計では数字として表示される桁数以上の情報は捨てられている。

このように、デジタルデータは離散的な有限個の状態に最初からデータが分かれているため、各々に2進数値を対応させることで二元状態の羅列によりデータを表現できる。また、状態が有限個であり、2進数値に対応付けるときに誤差が生じることがない。この例としては、ある文字を音声と書き文字の双方で伝える場合を考えてみよう。文字を音声で伝える際、例えば”し”と”ひ”がごく近い発音だとどちらの文字かひどく伝わりづらいことがある（実際に、江戸言葉や上方言葉ではこれらが混同した発音あるいは入れ替わることがある）。このため、「執事」と「羊」や「しなびた」と「ひなびた」など、”し”と”ひ”しか違いのない2つの言葉は時として文脈などから推定せざるを得ないこととなる。だが、書いている文字であればどちらを指しているのかは一目瞭然であり誤解の恐れはない。ここで挙げたのは極端な例であるが、デジタルデータは最初から状態が明確に分かれていることがアドバンテージとなっているということである。

### 1.1.3 【補足】2進数・記数法について

本節は前提知識の補足である。  
既知の読者におかれては飛ばして次節を読みたい。

先の解説で2進数を用いたが、記数法について慣れない読者に向け、補足解説を記す。

我々が物を数える際には手の10本の指を用いる。ここから物を数える位取りとして、指がいっぱいになったら位を1増やすということで10進法が生まれた。10進法とは、10(十、ten) 毎に位がひとつ上がるような数え方であり、我々が普段使っている数の表記法である。10進法により表された数のことを10進数と呼ぶ。

一方、先の二元状態は2つの指を持つといえる。そこで、2毎に位がひとつ上がり、0,1,10,11,...と数える2進法を使う。

これらのような、数の表記にあたりどのようにして位をとるかの方法を位取り記数法 (positional notation) あるいは単に記数法と呼ぶ。また、その位取りの基準となる数 (10進法なら10、2進法なら2) を基数と呼ぶ。

## 記数法の一般論

先までの例と同様に、指の本数から考えよう。指が  $n (\geq 2)$  本しかないと仮定する。このときは、 $n$  本がいっぱいになったら次の位に移る。たとえば、指が 3 本しかなければ、3 になったら次の位に移るようにして数えることができる。

一般の自然数  $n$  に対して、 $n$  進法とは、 $n$  を基数にする方法である。つまり、数えていて、 $n$  に達する毎に上の位に移るという事である。

例えば、123 という数字を考えよう。これは、123 個の 1 であるが、同時に 12 個の 10 と 3 個の 1 である。更に分ければ、1 個の 100 と 2 個の 10 と 3 個の 1 と分かれる。つまり、 $10^k$  の束がいくつあるかを数え、それが 10 個に達したら  $10^{k+1}$  を 1 増やす。これによって、各位は  $10^{k+1}$  の束にならない余りとなる。

上記に従って位取りをしてみよう。10 進法では、小数点の直上を  $10^0$  とし、その上の桁を  $10^1, 10^2, \dots$ 、小数点以下を  $10^{-1}, 10^{-2}, \dots$  としている。この 10 を  $n$  に変えたのが  $n$  進法である。たとえば、2 進法における 11111 は  $2^4 \times 1 + \dots + 2^0 \times 1 = 31$  である。

$n$  進法で使われる数字は、 $n$  種類ある。2 進法なら 0 と 1 だし、10 進法なら 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 であり、16 進法の場合は 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, a, b, c, d, e, f である<sup>1</sup>。数字だけでは何進法であるかわからない場合もあるため、これを明確にするときには、数字の右下に  $33_{(n)}$  のように記す。たとえば、 $101.1_{(2)} = 5.5_{(10)}$  である。

## 記数法の変換

$n$  進法から 10 進法への変換はどのように行うだろうか。これは  $n$  進法の仕組みを考えてみればわかる。

$n^k$  の位が  $a_k$  であるような数字  $A$  の 10 進法での値は定義から次のように求められる。

$n$  進法から 10 進法への変換

$n^k$  の位が  $a_k$  であるような数字  $A_{(n)}$  は、10 進法において

$$A_{(10)} = \sum_k a_k \times n^k$$

と計算できる。

逆に、10 進法から  $n$  進法に変える場合は、どうすればよいだろうか。

1 つには、単に「取り尽くす」方法がある。つまり、 $n^k$  のうち、元の数を超えない最大のものを見つけ、その数で元数を割った商を  $n^k$  の位にし、余りについて  $n^{k-1}$  で割り…と繰り返すのである。特に、小数点がある場合はこの方法を利用すると楽である。

<sup>1</sup> アルファベットは大文字を用いることもある。

だが、この計算はやや面倒に感じることもあろう。そこで、主に整数向けであるが、先に書いた位取りの考えを用いる方法がある。

$n^k$  の位が  $a_k$  であるという事は、 $n^k$  で割った後に、小数点以下を切り捨て、 $n$  に対する剰余を取ると  $a_k$  になる、という事である。例えば、10 進法で 100 の位の数字を出したいとすれば、100 で割って 10 による余りを取れば良い。これと同じ性質が一般の  $n$  進法にも成立する。この性質を利用し、次のような手順を踏んで記数法を変換できる。

#### 10 進法から $n$ 進法への変換

1. 変換したい 10 進数を  $n$  で割り、その余りを別におく。
2. これを、商が 0 になるまで繰り返す。
3. 別においた余りを逆順に並べると  $n$  進数での表記になる。

いくつかの数を変換してみればわかるが、10 進整数は他の記数法でも整数である。一方、ある記数法で有限小数であるからと言って他の記数法で有限小数であるとは限らない。例えば、 $1/3$  は、10 進法では無限小数だが、3 進法であれば  $0.1_{(3)}$  とシンプルに表せる。

## 1.2 情報の符号化

先にデジタルデータを 2 進数化して表現することを論じた。情報内容を変えることなく、この 2 進数化のように用途 (記録・通信 etc...) に応じて変換を加えることを符号化あるいはエンコードという。必ずしも 1/0 で表すわけではなく、文字を手旗信号で表すのも符号化の一例といえる。

ここでは整数や文字と言った典型的なデジタルデータの 2 進数による表現方法 (=符号化) を説明する。また、やや歴史的な符号化の例としてモールス符号を、デジタルでアナログを近似している例として小数の表現方法を紹介する。

なお、これ以降の議論については、二元状態として **bit** (binary digit の略で二進数の意) を用いるものとし、8bit である場合の表現として **1byte** (バイト) あるいは (歴史的経緯により) 特に 8bit であることを強調したい際には **1octet** (オクテット) を用いる<sup>2</sup>。

### 1.2.1 整数の表現

2 進数での表現として最も単純なのは符号なし整数であろう。符号なしの整数をそのまま 2 進数に変換し、これをビットパターンとする方法である。 $N$ bit により、0 以上  $2^N$  未満の整数が表現される。

<sup>2</sup>2008 年まで、1byte=8bit であることは主流でこそあったものの明確な定義として定まっておらず、1byte が 8bit でない環境も存在していた。2008 年に ISO (国際標準化機構) や IEC (国際電気標準会議) において 1byte=1octet が正式に定まったため、現在では 1byte=8bit として考えて問題はないが、追従できていない環境などでは 1byte が 8bit でない可能性もありうる。このことから、本書においては 8bit であることが特に重要な場合には旧来の 1octet という表記を用いることとした。

一方、符号付き整数を扱う場合にはどうだろうか。例えば、先頭の1ビットを符号に割り当てるなどの方法が考えられる。だが、この方法では+0と-0が出る、符号の扱いの都合でCPUの実装等が複雑になるなどの難点がある。これらの理由の回避のため、近年の電子計算機では2の補数表現が用いられることが多い。

この表現は、次のようなものである。

まず、0を基準とする。10進法における0を、2進法における0と対応させるのは自然な考えであろう。さて、これから1を引いた数、すなわち-1をどう表すべきであろうか??

4ビットで考えることとしよう。1=0001から1を引くと0=0000になる。では、それより1小さいものはどうすればいいか。これは、1111とすれば良いのである。ビットがあふれる部分への繰り上がりあるいはその部分からの繰り下がりを見捨てれば、0000-1=1111であるし、足し算で考えても1111+0001=0000, 0000+0001=0001となる。また、嬉しいことに、この方法を用いた場合でも、先頭ビット0は+または符号なしを、1は-を示す。このように、-aのビット表現を、0からaを引いたものとして表現するのが2の補数表現である。

参考として、表1.1に、2の補数表現を用いた4ビットの場合の符号なし整数・符号あり整数一覧を示す。

表 1.1: 4bit 符号なし整数・符号あり整数の一覧

ビット列	符号なし整数	符号あり整数	ビット列	符号なし整数	符号あり整数
0000	0	0	1000	8	-8
0001	1	1	1001	9	-7
0010	2	2	1010	10	-6
0011	3	3	1011	11	-5
0100	4	4	1100	12	-4
0101	5	5	1101	13	-3
0110	6	6	1110	14	-2
0111	7	7	1111	15	-1

この例を見てもわかるように、(桁溢れの繰り上がりを見捨てれば)2進数・10進数とも順に並んでいるのが2の補数表現の特徴である。また、符号を用いた表現と違い0が1つになっており、-8から7までと、負の側は絶対値が1大きい値まで収まる。

#### 【補遺】補数の意味

本節は大筋に影響しない。  
難解あるいは興味索然たるものと感じる折には  
飛ばして次節を読みたい。

先の解説では2の補数表現について補数とは何か説明せずに仕組みのみを述べた。本節では、関心ある読者諸賢に向け、補数の意味について解説する。



ある基数  $b$  における自然数  $a$  の補数とは、 $a$  に足した時に桁が 1 増えるような最小の自然数  $c$  のことをいう<sup>3</sup>。10 進法における 74 に対する補数は 26 である。2 進数において  $1100_{(2)} = 12_{(10)}$  の補数は  $100_{(2)} = 4_{(10)}$  である。

先に出てきた 2 の補数表現とは、2 進数における  $a$  の補数を  $-a$  として使う ( $a > 0$ ) という考えである。但し、一般の補数では直上の 1 桁が増えるかどうか見るところ、この表現においては扱うサイズで桁溢れが起きるかどうかにより判断をする。換言すれば、「符号なし整数の最大値+1 から  $a$  を引いた値」といえる。先に挙げた 4bit における 5(0101) の補数は、4bit が桁溢れを起こすような最小の自然数、つまり  $16(10000)-5(0101)=11(1011)$  である。そこで、1011 という表現を -5 の表現とする。同じように、 $a$  の補数の符号なし整数表現 (2 進数表現) を  $-a$  として使う ( $a > 0$ ) というのが、2 の補数表現である。

### 1.2.2 文字の表現

---

デジタルデータとして文字を表現する場合には、それぞれの 2 進数値に対応した文字を定める。これを文字コードと呼ぶ。例えば、最も一般的な文字コードと言える ASCII(American Standard Code for Information Interchange) の場合、数字の 9 は 57 となり、文字 A は 65 に割り当てられている (表 1.2)。ここでは ASCII の例を出したが、文字コードには様々な種類があり、それによって同じ数字でも違う文字に割り当てられることがある。

1 オクテットでは 256 通りの文字しか表わせないため、日本ではひらがな・カタカナ・漢字を表すのに不十分である。そのため、日本語を扱う際などには、それらにも対応した Shift\_JIS や EUC-JP(Extended UNIX Code Packed Format for Japanese)、UTF-8 などの日本語の文字コードを使うことになる。Web ページなどを閲覧している時に文字化けして見えることがあるが、この多くは異なる文字コードを使っていることによるものである。異なる文字コードであれば当然違う文字が割り当てられているため、本来と違うコードによって解釈されてしまうと意図した文字と異なる文字として扱われるということである。

### 1.2.3 モールス符号

---

文字列の符号化として多くの制御文字がある ASCII より歴史ある例として、モールス符号があげられる。これは短点 (・。日本語ではトンと呼ばれる) と長点 (ー。同じく日本語ではツーと呼ばれる) の 2 元状態を用いた符号である。英文中の発生頻度の高いアルファベットには短い符号、発生頻度の低いアルファベットには比較的長い符号を割り当てることで、平均的な符号長を短縮したものである (表 1.3 参照)。

長点は短点の 3 倍の長さで定まっており、また点と点の間には短点 1 つ分の間隔を空ける。文字間隔は短点 3 つ分、語間隔は短点 7 つ分と定められている。

### 1.2.4 小数の表現

---

小数は (無限小数なども含めれば) 文字や整数と違い、連続的な (アナログな) データである。このため、先に挙げたデジタルデータの利点である「元から離散的で有限個」という性

<sup>3</sup>厳密にはこの  $c$  を ( $b$  進法における) 基数の補数 ( $b$  の補数) と、 $c$  を 1 小さくしたものを ( $b$  進法における) 減基数の補数 ( $b-1$  の補数) と呼ぶ。本書で出てくるのは全て前者である

表 1.2: ASCII コード一覧 (16 進接頭辞 0x は省いた)

10 進	16 進	文字	10 進	16 進	文字	10 進	16 進	文字
0	00	NUL(Null 文字)	16	10	DLE(伝送制御拡張)	32	20	(空白)
1	01	SOH(ヘッダ開始)	17	11	DC1(装置制御 1)	33	21	!
2	02	STX(テキスト開始)	18	12	DC2(装置制御 2)	34	22	"
3	03	ETX(テキスト終了)	19	13	DC3(装置制御 3)	35	23	#
4	04	EOT(転送終了)	20	14	DC4(装置制御 4)	36	24	\$
5	05	ENQ(照会)	21	15	NAK(受信失敗)	37	25	%
6	06	ACK(受信 OK)	22	16	SYN(同期)	38	26	&
7	07	BEL(警告)	23	17	ETB(転送ブロック終了)	39	27	'
8	08	BS(後退)	24	18	CAN(とりけし)	40	28	(
9	09	HT(水平タブ)	25	19	EM(メディア終了)	41	29	)
10	0a	LF(改行)	26	1a	SUB(置換)	42	2a	*
11	0b	VT(垂直タブ)	27	1b	ESC(エスケープ)	43	2b	+
12	0c	FF(改頁)	28	1c	FS(フォーム区切り)	44	2c	,
13	0d	CR(復帰)	29	1d	GS(グループ区切り)	45	2d	-
14	0e	SO(シフトアウト)	30	1e	RS(レコード区切り)	46	2e	.
15	0f	SI(シフトイン)	31	1f	US(ユニット区切り)	47	2f	/

10 進	16 進	文字	10 進	16 進	文字	10 進	16 進	文字	10 進	16 進	文字
48	30	0	68	44	D	88	58	X	108	6c	l
49	31	1	69	45	E	89	59	Y	109	6d	m
50	32	2	70	46	F	90	5a	Z	110	6e	n
51	33	3	71	47	G	91	5b	[	111	6f	o
52	34	4	72	48	H	92	5c	\	112	70	p
53	35	5	73	49	I	93	5d	]	113	71	q
54	36	6	74	4a	J	94	5e	^	114	72	r
55	37	7	75	4b	K	95	5f	_	115	73	s
56	38	8	76	4c	L	96	60	`	116	74	t
57	39	9	77	4d	M	97	61	a	117	75	u
58	3a	:	78	4e	N	98	62	b	118	76	v
59	3b	;	79	4f	O	99	63	c	119	77	w
60	3c	<	80	50	P	100	64	d	120	78	x
61	3d	=	81	51	Q	101	65	e	121	79	y
62	3e	>	82	52	R	102	66	f	122	7a	z
63	3f	?	83	53	S	103	67	g	123	7b	{
64	40	@	84	54	T	104	68	h	124	7c	
65	41	A	85	55	U	105	69	i	125	7d	}
66	42	B	86	56	V	106	6a	j	126	7e	~
67	43	C	87	57	W	107	6b	k	127	7f	DEL(削除)

表 1.3: 国際モールス符号のアルファベット

英字	出現率	符号	英字	出現率	符号	英字	出現率	符号
A	6.42%	・ —	J	<u>0.08%</u>	・ — — —	S	5.14%	…
B	1.27%	— …	K	0.49%	— ・ —	T	<b>7.96%</b>	—
C	2.18%	— ・ — ・	L	3.21%	・ — …	U	2.28%	… —
D	3.17%	— …	M	1.98%	— —	V	0.83%	… —
E	<b>10.31%</b>	・	N	5.74%	— ・	W	1.75%	・ — —
F	2.08%	… — ・	O	6.32%	— — —	X	0.13%	— … —
G	1.52%	— — ・	P	1.52%	・ — — ・	Y	1.64%	— ・ — —
H	4.67%	…	Q	<u>0.08%</u>	— — ・ —	Z	<u>0.05%</u>	— — …
I	5.75%	…	R	4.84%	・ — ・	空白	<b>18.59%</b>	

質がなく、デジタル表現のために「最も近い表現可能な値」に丸める必要がある。これは次章のアナログデータの表現で出てくる量子化というステップであり、これによる誤差 (プログラミングの世界では一般に丸め誤差という) は量子化雑音に相当する。

コンピュータ等で扱う小数は、数学等の小数と違い、表せる個数が限られたデジタルデータである。本来はアナログデータである小数を出来る限り近い値で表せるように考えられたデジタル表現とも言える。その表現方法には固定小数点数表現と浮動小数点数表現がある<sup>4</sup>。

### 固定小数点数

ある定まったビットまでを整数部として、そこから下の部を小数部としてみなすことでしょうすうを表現する方法を、固定小数点数表現という。例えば、16ビットある時に、上位8ビットがその数の整数部を示し、下位8ビットが小数部を示すようにすれば良い。この時、 $2^0 = 1$ を示すビットのすぐ下のビットは $2^{-1}$ を、その下は $2^{-2}$ を… というように、桁が連続的に定まっていくのが普通である。

この方式は表現できる数の範囲はごく限られるものの (例えば先の 16bit の例ならば、10進数で  $1/256$  刻みの値しか扱えない)、定められた範囲においては、誤差を出すことなく高速に計算できる。10の累乗で表現するような値が頻出する科学計算では利用しづらいが、桁数がある程度定まっている上に誤差が大きな問題となる経済計算などで用いられる。

### 浮動小数点数

浮動小数点数表現は、小数点の場所を上手く変えながら (浮動) 表す方法である。浮動小数点数では、表したい数をまず以下の形式で表現する。

$$(\text{符号})(\text{仮数}) \times (\text{基数})^{(\text{指数})}$$

この内、基数は通常 2 が用いられる。また、仮数は 1 以上 2 未満の数である (これにより、上記の形式での表現が一意に定まる)。そして、符号 (1bit)、指数 (符号付き整数)、仮数の小数

<sup>4</sup>それぞれ、固定-小数点-数、浮動-小数点-数と区切る。対応する英語も、fixed point number, floating point number である。

部 (固定小数点数) を各々保存する。例えば、-4.8 という数字を浮動小数点数で表現することを考えよう。 $-4.8 = -1.2 \times 2^2$  である。したがって、符号部には-が、仮数部には0.2が、指数部には2が格納されることになる。なお、指数・仮数を何 bit とするかは規格によって異なるが、近年のコンピュータでは IEEE754 という規格に従っていることが多い。

固定小数点数型に比べ、指数部の表現によって科学計算などで用いるような大小様々な値に比較的近い値を表現できるが、誤差が出たり、計算の時間がかかったりといった問題点もある。固定小数点数型は狭い範囲を等間隔に区切っている表現であるが、浮動小数点数は広い範囲をその絶対値に応じてスケールリングして区切っているという見方もできる。

## 1.3 符号化効率の改善と圧縮

データの記録や通信の資源は限られているため、少しでも効率よく記録・通信したい。先に挙げたモールス信号は光あるいは音を人間が認識して行うことから伝達に時間がかかる。そのため、少しでも短縮するために文字の出現頻度を考慮して符号の長さが決まっていた。同様に、何らかの手法を用いてデータを圧縮して符号化する、あるいは符号化したものを圧縮する方法が考えられた。

### 1.3.1 符号化効率の改善

先に扱った例では、取りうる値各々に定長を割当てて情報を表していた。しかし、何らかの手法でこの定長を短くしたり、可変長にしたりすることで、同じ情報でもより短い符号にて記録できる。

サイコロの出目を考えよう。6つの面が等確率で出る普通のサイコロの場合、単に定長を割り当てるなら 3bit が必要になる。しかし、00,01,100,101,110,111 という 6通りの状態を出目に対応させると、平均符号長は

$$\frac{1}{6}(\text{の確率}) \times 2(\text{bit}) \times 2(\text{通り}) + \frac{1}{6}(\text{の確率}) \times 3(\text{bit}) \times 4(\text{通り}) = \frac{8}{3}$$

となり、3bit の定長で割り当てる場合に比べておよそ 11% 短いデータで表現できている。また、ここで表した 6つの符号は「先頭から読む際、0 を読んだら 2bit, 1 を読んだら 3bit」というルールを課すことで混同することなく元の情報を取り出すことができる。

このように、データ量を節約することは保存の上でも無論便利であるが、より短時間での通信や記録が可能となるため利便性が大きい。

#### モールス符号の平均符号長

モールス符号は文字の出現頻度を考慮して符号の長さが決まっていたと記したが、実際の平均符号長はどのぐらいか計算してみよう。表 1.4 に各符号の符号長 (符号の文字数) と、その点長 (一つの文字を表すのにかかるのが短点の何倍になるか) を記した。

ここで、点長は、長点が短点の 3 倍の長さで、点と点の間が 1 倍の長さであることから、長点の数  $\times 3$  + 短点の数 (空白も含む) + 文字数 - 1 で計算している (実際には、これに字間・語間の長さが加わるが、これは無視している)。

表 1.4: 国際モールス符号の英字符号長

英字	出現率	語長	点長	英字	出現率	語長	点長	英字	出現率	語長	点長
A	6.42%	2	6	J	0.08%	4	13	S	5.14%	3	5
B	1.27%	4	9	K	0.49%	3	9	T	7.96%	1	3
C	2.18%	4	11	L	3.21%	4	9	U	2.28%	4	7
D	3.17%	3	7	M	1.98%	2	7	V	0.83%	7	9
E	10.31%	1	1	N	5.74%	2	5	W	1.75%	4	9
F	2.08%	4	9	O	6.32%	3	11	X	0.13%	4	12
G	1.52%	3	9	P	1.52%	4	11	Y	1.64%	4	13
H	4.67%	4	7	Q	0.08%	4	13	Z	0.05%	4	11
I	5.75%	2	3	R	4.84%	3	7	空白	18.59%	0	1

表 1.4 を元に、平均符号長を計算する。出現率 × 語長の総和を取ると、2.12 となり、空白を含めて 27 通りのデータであるにもかかわらず、二元状態 2 文字強とかなり短くなっていることがわかる。また、同様に各文字の点長の平均は (短点を 1 として) 5.18 の長さである。仮に文字を全て固定長 5bit にし 1 を長点・0 を短点・字間の空白は同じと考え、(最も効率よく当てはめたととしても) 2 倍以上の長さとなるため、長さを 2 元状態にするという制約下においてモールス符号は効率化されていることがわかるだろう<sup>5</sup>。

### 1.3.2 可逆圧縮・不可逆圧縮

圧縮には大きく分けて、「本来の情報のうち不要な部分を捨て去ることにより符号を短くする」手法と、「本来の情報の何らかの特性を利用してより短い表現を用意し符号を短くする」手法がある。前者のように、元の情報から何らかの情報が捨てられるケースでは捨て去られた部分のデータを復元して元のデータに戻すことは不可能であるので、不可逆圧縮 (あるいは非可逆圧縮と呼ばれる)。一方、後者の手法は元のデータに戻すことが可能なので可逆圧縮と呼ばれる。本来の圧縮の手法では元のデータの復元が可能であることが条件なので不可逆圧縮は圧縮とは呼べないともされるが、現実的には圧縮の一種として扱われている。

電気を使わない世界での不可逆圧縮の例を考えてみよう。不可逆圧縮としては、電車の遅延証明書が挙げられる。電車の遅延証明書は「この時間帯に最大で○分の遅れが出たことを証明する」と言う形式になっている。それぞれの電車についてどの駅で何分遅れたかは、データとしては存在するだろう。しかし、そのような細かい情報は遅延証明書を必要とする側にとっては不要な情報である。また、それぞれについて証明書を発行する手間も膨大なものになる。そこで、時間帯と最大の遅れによって遅延証明書を一律化したのである。このように、大勢に影響ない情報を捨て去るのが不可逆圧縮である。電気通信の世界では、音声データの可聴域以外の部分を捨てる (MP3) などの例が見られる。

一方、電気の世界でない可逆圧縮の例としては、歌詞カードがわかりやすい。歌詞カードには、同じ歌詞の部分に (\* くりかえし) などと書かれる。これによって、歌詞という情報に欠損は生じない。しかし、全体としての文字数や行数は減り、狭い歌詞カードのスペース

<sup>5</sup>但し、モールス符号は最適な割り当て方をしているとは言えない。平均符号長を最適化すると 1.99 となってその時の点長の平均は 4.95 と 5 を切る。点長の平均を最適化した場合は 4.84 となり、平均符号長は 2.07 と、より良い方法もある点には留意されたい。

を節約したり、一覧性を向上させたりできる。歌詞カードを読む側は(\* くりかえし)をあてはまる部分と置き換えることで(解凍)元の歌詞を復元できる。これが可逆圧縮である。電気通信の世界では、例えばデータを zip ファイルや tgz ファイルなどが挙げられる。

本書ではデータの特性などを個別に論じないため、可逆圧縮のみを扱う。

### 1.3.3 エントロピー符号

モールス符号である程度考慮されていたように、元の情報の頻度に偏りがある場合、高頻度のものを短い符号に割り当てることで平均的な符号を短くすることができる。

先に、サイコロの出目を考えた時は全て等確率とした。しかし、サイコロの出目が必ずしも等確率で出るとは限らない。市販されている一般のサイコロでも、その目の彫りや塗装で厳密に等確率とは言えないのだそうである<sup>6</sup>。また、直方体の”サイドタ”や、おもりを入れた”グラサイ”など、明らかに等確率ではないものもある。例えば、2つの出目が1/3の確率で出て、残りの目が1/12で出るようなサイコロ(?)のデータを考えてみる。この時、先に考えたのと同じく 00,01,100,101,110,111 を割り当てるなら、00 と 01 の 2bit の符号は、1/3 の確率で出る目に割り当てたほうが全体としては短くなるだろう。この時の平均符号長は

$$\frac{1}{3}(\text{の確率}) \times 2(\text{bit}) \times 2(\text{通り}) + \frac{1}{12}(\text{の確率}) \times 3(\text{bit}) \times 4(\text{通り}) = \frac{7}{3}$$

となり、等確率の場合に比べてより短くできている。

このように、出現確率によって長さを変えて割り当てる符号をエントロピー符号と呼び、次のような種類がある。

#### エントロピー符号の例

- |                     |                    |
|---------------------|--------------------|
| ● アルファ符号            | ● (整数の) ユニバーサル符号   |
| ● ガンマ符号             | ● フィボナッチ符号 (KZ 符号) |
| ● シャノン符号            | ● レーベンシュタイン符号      |
| ● ハフマン符号            | ● 算術符号             |
| ● CBT 符号 (先のサイコロの例) | ● レンジ符号            |

以下、この内符号語が整数であり、アルゴリズムが易しいなど、比較的シンプルなものを見ていこう。

#### アルファ符号

アルファ符号は単進符号とも言われ、あとに示すガンマ符号の基礎となっている考えの符号である。

<sup>6</sup>プレジジョンダイスや”世界最速のサイコロ”といった、限りなく等確率に近づけたサイコロもあるようだ。

符号化したいデータの出現頻度が  $n$  番目に高いとすると、これを  $n$  桁の符号に変換する。符号は末尾が 1 で、それ以外の桁を 0 としたものなどが代表的である。つまり、1 番目は 1, 2 番目は 01, 3 番目は 001, 4 番目は 0001 という具合である。これ単体ではデータ数が多い時に非効率な符号となる可能性も高いので、単体で用いられることは少ない。

### ガンマ符号

ガンマ符号は、アルファ符号と同様に符号化したいデータの出現頻度が  $n$  番目に高い時に、 $n$  に応じて長くなる符号である。 $n$  を 2 進数表記し、その上に  $n$  の 2 進桁数-1 個の 0 をつける。例えば、1 番目は 1, 3 番目は 011, 6 番目は 00110, 10 番目は 0001010 となる。例示したケースでは固定長にしたほうが効率よく見えると思われるが、データパターン数が多く偏りが強いほど、ガンマ符号の効率が良くなる。

### CBT 符号

CBT 符号 (Complete Binary Tree 符号) は、先の 6 面サイコロの例として挙げたもので、固定長符号を元にしながら、不要な箇所を省いて符号を短くする方法である。

$k$  ビットの固定長符号では  $2^k - 1$  までの数値を表現できる。しかし、全データが  $m$  より小さければ、 $m$  以上の部分は無駄になる。そこで、 $2^k - m$  未満の数を  $k - 1$  ビットの 2 進数に割り当て、 $2^k - m$  以上の数は、その数に  $2^k - m$  を足した数の  $k$  ビット 2 進表現に割り当てる。例えば 10 通りの数を表す場合、4bit あれば十分なので、0 から 5 はそれぞれ 000 から 101 に割り当てる。6 から 9 は 1100 から 1111 に割り当てる。この時、割り当てる順番を出現頻度降順にすれば最も良い効率の割当となる。

### フィボナッチ符号 (KZ 符号)

フィボナッチ符号は  $F_{n+2} = F_{n+1} + F_n$ ,  $F_1 = F_2 = 1$  という式により計算されるフィボナッチ数列を用いた符号化法である。フィボナッチ数列を順に記すと 1, 1, 2, 3, 5, 8, 13, 21... となるが、この内最初の 1 を除いた、1, 2, 3, 5, 8, 13, 21... を用いて数値を表現する。ゼッケンドルフの定理と呼ばれる定理では「任意の正の整数が、連続するフィボナッチ数を含まず、同じフィボナッチ数を 2 度含まない形で、相異なる 1 つ以上のフィボナッチ数の和として一意に表現できる」とされる。例えば、42 であれば、 $8 + 34 (F_6 + F_9)$  により表現できる。この時、前から順に  $F_2, F_3, F_4, \dots$  の有無を 1/0 で表して並べたものをフィボナッチ表現と呼ぶ。先の 42 であれば 00001001 であるし、33 であれば 1010101 となる。

出現頻度が  $n$  番目に高いデータを、数  $n$  のフィボナッチ表現にして表すものがフィボナッチ符号である。ゼッケンドルフの定理の「連続するフィボナッチ数を含まず」の件からわかる通り、フィボナッチ符号には連続した 1 が出ないため、11 などを符号の制御用に利用できる。特に、データの接頭辞として 110 を付す形のを **KZ 符号** (Kautz-Zeckendorf 符号) と呼ぶ。これは、データを先頭から見なくともどこがデータの区切りかはつきりわかるという利点がある。

## ハフマン符号

エントロピー符号の中でも最も代表的な例がハフマン符号である。これは、次の手順により符号化を行う手法である。

### ハフマン符号の手順

1. すべての情報源シンボルを節点 (葉) とする 2 進木 (binary tree) を書き、葉にはそのシンボルと発生確率を併記する。
2. 最も発生確率の低い葉を 2 枚選び、それらに対する新しい節点を作り 2 枚の葉と 2 本の枝で結ぶ。このとき 2 本の枝にそれぞれ 1、0 を割り当てる。さらに新しい節点には 2 枚の葉の確率の和を書き、この接点を新たな葉と見なす。
3. すべての情報 (葉) を枝で結ぶまで 2 を繰り返す。
4. 最後に作成した新しい節点を根と呼び、根から各情報 (葉) に至る枝をたどりながら、2 でその枝に割り当てた 1、0 を順に読んでいくことで符号化を行う。

具体的に、8 種類の文字をハフマン符号化した例を図 1.1 に示す。

Alphabet	Probability	Tree of code	Code word
A	0.5		1
B	0.2		01
C	0.1		0011
D	0.08		0010
E	0.05		0001
F	0.04		00001
G	0.02		000001
H	0.01		000000

図 1.1: 8 種類の値のハフマン符号化 東邦大学メディアネットセンター 情報通信理論 (佐藤洋一) より引用

ハフマン符号は整数の符号語長という制約のもとでは、常に最適な符号を構成できるという特徴がある。(実数の符号語長を許す場合には、レンジ符号などより高効率なものがある。) これはシャノンの情報源符号化定理と呼ばれる定理により説明できる<sup>7</sup>。

<sup>7</sup> 数学的な取り扱いがやや高度であるため、本書では示さない。ここでは「圧縮できる限界を示した定理」と考えれば良い



### 1.3.4 辞書式 (ユニバーサル) 符号

データにおいて値が無秩序に並んでいることは少なく、なんらかの関連性があるものである。データ値の並び方のパターンを抽出し、そこに元よりも短いコードを振って出力することでデータの情報量を減らすことができる。この手法を辞書式符号またはユニバーサル符号と呼び<sup>8</sup>、次のような種類がある。

ユニバーサル符号の例

- バイト対符号化 (BPE)
- ランレングス (RLE)
- LZ78
- Deflate

以下、この内の何種類かを見ていこう。

#### バイト対符号化 (BPE)

バイト対符号化 (Byte Pair Encoding) は、2 バイト (2 文字) の複数個の塊を辞書にする圧縮方法である。例えば、日本語で「ふらふらとふらつく」という文を考えてみよう。この文には「ふら」という文字列が3度出てくる。そこで、この2文字を仮にAに変えると「AAとAつく」と文字列がずいぶん短くなる。

このように、2文字の同一の文字列があるときこれを使われていない別のバイトに置き換えていく（再帰的に適用する）ことで、文字列をより短くできる。そして、置き換えた一覧（辞書）とともに送付することでバイト列を圧縮する。圧縮には時間がかかるが、展開はすぐに行えるため、受信側が非力な機器である際に利用される手法である。

#### LZ78

**LZ78** は読み込んだ記号列から動的に辞書を作成して、それをもとに入力記号列を置き換えていく動的辞書法とも呼ばれる圧縮法である。先に示した BPE と違い、辞書を別途送ることなくデータを圧縮する。

この圧縮手順は次のようになる。

1. 辞書を空にし、コード値の最大を0とする。
2. データ列を走査し、データパターンが辞書に存在するか調べる。辞書に存在した場合はそのコード値を出力する。なければ、コード値0を出力する。
3. コード化を行ったデータパターンの次のデータ1つをそのまま出力する。

<sup>8</sup>エントロピー符号に出てくる (整数の) ユニバーサル符号とは別の意味であるので注意。英語でも、整数の方は universal code、辞書式の方は universal source coding と名称が異なる。整数の方のユニバーサル符号とは、大きい数のほうが出現頻度が低い分布において効率的に (最適な符号化と同程度のオーダーで) 符号化可能な手法の総称である。

4. 前項・前々項で出力したデータについて、コード値を辞書の値に置き換えたデータを、新データパターンとして辞書に登録する。このときのコード値は、コード値の最大+1とする。
5. データ終端に至るまで、上記の2～4項を繰り返す。

わかりづらいので、AAABBBAAABBBBAAA という文字列を圧縮することを考えてみよう。

1. 最初、辞書は空である。データ列を最初から見ていったときのパターンはAである。Aというパターンは辞書にないから、コード値0とその次のデータ(A)を出力し、辞書にはコード値1としてAを出力する。

文字列：A AAABBBAAABBBBAAA

出力：0A

辞書：0-(空),1-A

2. 2文字目はAであるので、1-Aに合致する。このため、コード値1とその次のデータ(A)を出力し、辞書にはコード値2としてAAを出力する。これにより、3文字目のAまでが表されたことになる。

文字列：AAA BBBAAABBBBAAA

出力：0A1A

辞書：0-(空),1-A,2-AA

3. 4～6文字目についても同様に行う。

文字列：AAABBB AAABBBBAAA

出力：0A1A0B3B

辞書：0-(空),1-A,2-AA,3-B,4-BB

4. 7文字目～9文字目は、コード値2のAAにAをつけたものであるから、2Aとする。2Aに対応するAAAを辞書に登録。

文字列：AAABBBAAA BBBBAAA

出力：0A1A0B3B2A

辞書：0-(空),1-A,2-AA,3-B,4-BB,5-AAA

5. 10文字目～12文字目は、コード値4のBBにBをつけたものであるから、2Bとする。2Bに対応するBBBを辞書に登録。

文字列：AAABBBAAABBB BAAA

出力：0A1A0B3B2A4B

辞書：0-(空),1-A,2-AA,3-B,4-BB,5-AAA,6-BBB

6. 以下、同様に繰り返す。

文字列：AAABBBAAABBBBAAA

出力：0A1A0B3B2A4B3A2(空)

辞書： 0-(空), 1-A, 2-AA, 3-B, 4-BB, 5-AAA, 6-BBB, 7-BA, 8-AA(空)

出力のデータはもとのデータより短くなっており、また出力内容から辞書を作成しつつ解凍することができるので、辞書を別途送る必要がない。これが動的辞書法と言われるゆえんである。なお、この例ではあまり長い例がないため圧縮効果が低いが、実際には辞書に長いデータが入って同じデータが頻出する場合などより効果的な圧縮となる。

またこの手法は、記号列の長さを増して部分列に分解していくことから、増分分解法とも呼ばれる。ユニバーサル符号の代表的な手法である。

## ランレングス (RLE)

ランレングス (Run Length Encoding) あるいは連長圧縮は、同一のデータが複数続いているとき、その個数とデータで表現することにより全体の長さを短くする方法である。例えば、"AAAAAABBBBBB" というデータがあったとき、"A が7個、B が5個" の意味で "7A5B" とすると文字列が短くなる。これは特に、データの種類の数が少なく、また連続している可能性が高い場合に大きな効果を発揮する。とりわけ、白黒のファクシミリでは、白か黒かの2値であることと、よほど細かい市松のような状況でない限りは白・黒の連続性が高いと考えられることから、ランレングスがよく利用される。

### 1.3.5 差分 (デルタ) 圧縮

符号の圧縮の多くは、先に挙げたエントロピー符号または辞書式符号に分類される。しかし、その双方に分類されない（というより分類の前処理にあたる）のがここで扱う差分圧縮またはデルタ圧縮である。この方式は、通信よりも寧ろデータのバックアップなどによく用いられる。

定期的にシステムやデータのバックアップを取る場合を考える。その期間にもよるが、毎回全てのデータを上書きする必要はないことが多い。期間が短いほどその変化は少ないだろう。そこで現行と以前のあるバックアップ時との差分を用意し、これを保存しておく。このように、ある完全なデータを一つ用意した上で、そこからの差分を用いて圧縮するのがこの差分圧縮である。性質上、前述のエントロピー符号や辞書式符号と併用することができる。特に、差分データについては"差分なし"あるいは"僅かな差分"の部分が大半を占めることが多いため、これらに短い符号を割り当てることにより圧縮効率が高まると期待できるのである。

この例ではバックアップを出したが、関連性のある複数のデータであれば同様の効果が期待できる。現実にも定点観測のデータやソフトウェアのアップデートなどでこの手法が利用されている。

## 演習問題

- 1 古いゲームにおいては、容量の制約から 16bit の整数型などを採用せざるを得なかった。そのため、例えば (0 以下の場合は 0 以下になった時の処理が必要となる) 体力などの数値は 32768 が最大となっていた (現実には、これを 3~4 データ用意するなどして増強を図っていた)。ところが、この最大値は 2 の補数表現を採用した 16bit 符号付き整数型の最大値と 1 異なる。内部上の表現は確かに 2 の補数表現で間違いなく、処理によってこの違いが生じているのだが、どのような処理をしていると考えられるか説明せよ。
- 2 データの種類数と、各データの出現比率が与えられるとき、そのハフマン符号化の例と、その際の平均符号長を出力するプログラムを作成せよ。例えば、4 種のデータが表 1.5 のような比率で入力される場合、平均符号長は 1.75 となる。なお、簡単のため

表 1.5: ハフマン符号化を考えるデータ例

データ種別	A	B	C	D
出現頻度	4	2	1	1
符号例	0	10	110	111

に出現頻度は正整数値で、その合計は 50000 を超えないものとして良い。

- 3 浮動小数点数の配列を圧縮して送りたい場合、単純なランレングスでは効果が出づらい。そこで、各値の同一ビットのみを集めてランレングスをかけ、これを順に記すことで圧縮することを考える。すなわち、1 ビット目のランレングス情報、2 ビット目のランレングス情報…と書かれたデータを作成して圧縮するのである。この手法について以下の問いに答えよ。
- (a) 浮動小数点数の配列にどのような傾向があるときにこの圧縮は効果的か。
- (b) 整数や文字の配列の場合、このように同一ビットのみを集めた圧縮は効果的と言えるか。データの特性とともに考えてみよ。

## アナログデータの記録

## 第II部

### 電気伝送の方法



## 電気信号の同期



## 第 4 章

# 変調

## 伝送品質の担保

## 伝送路の方式

## 第III部

### コンピュータ・ネットワークの基礎



## 通信プロトコル

## ネットワークの構成

## ネットワークアーキテクチャ



## ネットワーク機器

## Layer1：物理層

## Layer2 : データリンク層

## Layer3：ネットワーク層 (1)IP とアドレス

## Layer3 : ネットワーク層 (2) ルーティング

## Layer4 : トランスポート層

## Layer5～7：上位3層

## 第Ⅳ部

### ネットワーク・プロトコル





## ネットワークモデル・再論

# telnet

# DNS

## DHCP

# FTP

# HTTP

## メールに纏わるプロトコル



## 認証に纏わるプロトコル

## 第Ⅴ部

### 安定通信網の構築と運用



## 冗長化

## セキュリティ

## トラブルシューティング手法の基礎

# 索引

## 1-9

2 の補数 ..... 18

## B

bit ..... 17

byte ..... 17

## C

CBT 符号 ..... 25

## I

IETF ..... 8

Internet Engineering Task Force. → IETF

## K

KZ 符号 ..... 25

## L

LZ78 ..... 27

## O

octet ..... 17

## R

Request For Comments ..... → RFC

RFC ..... 8

## あ

圧縮 ..... 22

アルファ符号 ..... 24

暗号 ..... 9

エンコード ..... 17

エントロピー符号 ..... 24

## か

可逆圧縮 ..... 23

ガンマ符号 ..... 25

基数 ..... 15

記数法 ..... 15

口伝 ..... 2

位取り記数法 ..... 15

固定小数点数表現 ..... 21

## さ

差分圧縮 ..... 29

シーザー暗号 ..... 9

辞書式符号 ..... 27

10 進数 ..... 15

10 進法 ..... 15

シャノンの情報源符号化定理 ..... 26

ジョーク RFC ..... 8

増分分解法 ..... 29

## た

デルタ圧縮 ..... 29

動的辞書法 ..... 27

## な

二元状態 ..... 14

2 進法 ..... 15

## は

バイト対符号化 ..... 27

ハフマン符号 ..... 26

非可逆圧縮 ..... 23

フィボナッチ符号 ..... 25

不可逆圧縮 ..... 23

符号化 ..... 17

浮動小数点数表現 ..... 21

平均符号長 ..... 22

補数 ..... 18

## ま

モールス符号 ..... 19

文字コード ..... 19

## や

ユニバーサル符号 ..... 27

## ら

ランレングス ..... 29

連長圧縮 ..... → ランレングス