第 2 章 情報の表現

この章では、情報とはいかなるもので、その表現とは何を意味するのか、情報を表現するモノ(メディア media、媒体)にはどのようなものがあり、その特徴は何かといった事柄を、情報科学やコンピュータサイエンスの基礎となる考え方に立ち戻り問い直していきます。そして、コンピュータを用いて問題を解決したり、国境を越えた通信網を通してコミュニケーションを行ったりする場合、私たちの目に見えないところで電子的に表現された情報が大活躍していることを学んでいきます。

情報社会は、人間の頭脳労働のある部分を肩代わりするコンピュータと、それらが国内外にクモの巣のように張り巡らされた電気通信ネットワークによって支えられている社会です。

情報社会を飛び交う多種多様な情報は、コンピュータや電気通信に相応しい形で表現されています。この表現は、たった 2 つの異なった記号の組み合わせで多様な情報を取り扱っていることから、二進デジタル信号と呼ばれています。しかし「二進デジタル」という言葉は、多くの場合簡単に「デジタル」と呼ばれることが多いので、ここでも特に強調するとき以外は「デジタル」という言葉で進めていきます。

本章の目的は、デジタル信号という言葉の意味を正しく理解し、デジタル信号による文章・数値・画像・音など多様な表現と、その原理を学ぶことにあります。

2.1 情報と記号

コンピュータが出現する以前の社会で、人間はどのような工夫をして情報を表現してきたのかを振り返り、そこに見られる工夫と人間の認知能力との関わりを簡単に振り返っていきます。このことは、人間とコンピュータが情報をやり取りするときの窓口(マン・マシン・インターフェイス、man・machine・interface)のあり方を考えるときの基礎にも関わっています。

2.1.1 情報、データ、知識

情報に似た言葉にデータ(data)というものがあります。普段、情報やデータという言葉は曖昧に使われることが多いですが、誤解なく議論する場合には言葉の意味をある程度きっちりと約束しておく必要があります。

いま、個々の動物の外部に存在する物体や生物を総称して「モノ」と呼ぶことにします。すると、人間に関わらず、ある程度進化した動物は、他の生物や物体が発する「モノ」を感覚器官を通して受け取り、それらを脳を働かせて処理し、その結果に応じて食料を得たり外敵から逃れたり、仲間と連絡を取るというようにして生存しています。この一連の活動でやり取りされる「モノ」のうち、人間が判断や評価づけするための素材や資料をデータ(data)、データにしたがって判断や行動に役立つものを情報(information)とします。次の例はデータと情報の区別の必要をよく表わしています。

【例】天気予報、兵庫県南部、一日中晴天、最高気温 32 度、最低気温 25 度

この天気予報は、「情報」なのか「データ」なのか、どちらでしょう。今この天気予報が夏の全国高校野球夏の大会中に出されたものとします。この予報は、球場でお弁当を売っている人にとって、明日何個のお弁当を用意しておけばよいかの判断にとって極めて重要な「情報」といえます。しかし、野球に関心が無く、また関東で営業しているお弁当屋さんにとっては「データ」に過ぎないないのです。

このように、ある人にとっては情報でも、他の人にとっては特に意味の無いデータになるものも多く、日常生活で情報とデータが曖昧に使われる理由の一つがここにあります。

さて、現在のところ、コンピュータは人間のように自分で価値判断ができないため、コンピュータにとっては取り扱うものはすべてデータといえます。つまり、コンピュータが処理したりネットワークを流れるものはすべて「データ」であり、人間と関わってくる場合に「情報」となったり「データ」に留まったりするわけです。勿論、「データ」と思っていたものが、状況の変化によって「情報」に変わることもしばしば起こります。

【用語】データ(data):人が判断や評価を下すための元となる素材のようなもの。

【用語】情報(information):人が特定の状況で判断や評価を伴って利用できるようなデータ

私たちは判断や評価を行う際に「知識」を利用します。では、知識とはなんでしょう。さきほど天気予報を例に見たように、情報には「そのときその場面での判断のため」というニュアンスが感じられます。これに対し、ある人が天気予報といっしょに出される天気図を見て、「この気圧配置は夏にしては異常だなあ。今年の夏は冷夏になりそうだ」という意見を言ったとします。この人は、天気図というデータを情報として取り入れ、それから判断を下すもとになる「何か」を持っているわけです。この「何か」が知識といわれるものなのです。

【用語】知識(knowledge):情報やデータを体系的にまとめ、判断や思考に利用活用できるように蓄積されたもの

知識を蓄える目的は、さまざまな場面での問題解決のためであり、クイズやペーパー試験で正解を増やすためではありません。単に「辞書のように詰め込んだ知識」では、具体的に起こってくる諸問題に対処できないことが多く、このような知識は「死んだ知識」といわれています。

【演習】学問的に使われる言葉は、専門用語と呼ばれます。日常生活で使われる言葉が同時に専門用語として使われる場合は要注意で、多くの場合、各学問分野ごとに定義されています。情報、データ、知識という言葉もそれにあたります。そこで、これらの言葉を、まず普通の国語辞典で調べ、次に学問分野別の事典で詳しく調べてみましょう。そして、それぞれの定義がどのような背景で生まれてきたかについて考えてみましょう。

【演習】データ、情報、知識という言葉をそれぞれ 1 つづつ使った短文を作り、次に他の言葉と置き換えて文章のニュアンスの違いについて話し合ってみましょう。

2.1.2 記号の発明とその処理

人間を他の動物と区別するものはなにかについてはいろいろの考え方がありますが、ここでは情報を取り扱う頭脳の働きの面から人間の特徴を見ていきます。

人間と記号

極めて原始的な生き物は、その行動のほとんどが遺伝子に組み込まれた情報によって支配されています。しかし、動物が進化して高等になればなるほど、誕生後に学習されたことが重要に成ります。学習は、動物が生活する環境に存在する「モノ」との関係を通して行なわれ、その経験が知識として蓄積されます。ここで「モノ」は、無機質な物質や動植物、そして人間まであらゆるものを指しています。

人類はやがて、情報を記録伝達する人工的な「モノ」を発明しました。このような情報を記録伝達する「モノ」を"記号(symbol)"と呼ぶことにします。

記号の発明はとりもなおさず「かく」ということの発明でもあります。「かく」は恐らく絵のような物に具体的な表現である「描く」から始まり、やがて道路標識のような象徴化や図形化を経て、言葉を「書く」文字へと進歩を進めました。

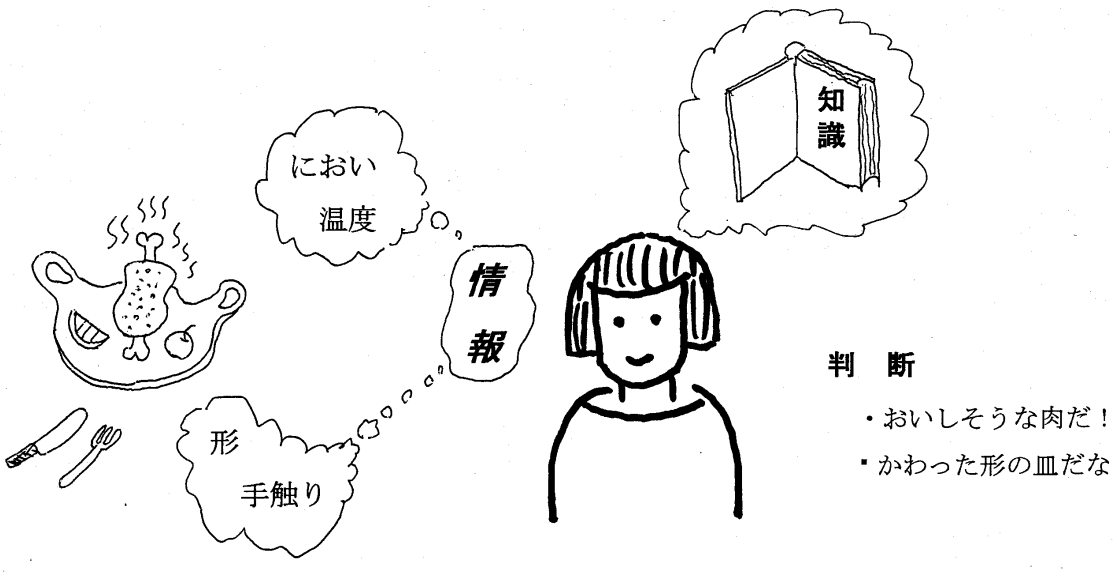


図 2.1: 情報と判断

人間は、記号を利用して遠くに離れた所に情報を伝達したり、歴史を伝えたりしてきました。記号の活用は、言葉での直接的なコミュニケーションの限界を超えた複雑な情報伝達を可能にし、人類に他の動物に見られない文明や文化をもたらしました。人類の文明や文化は、食料や生活材といった「モノ」自体の生産はもとより「記号」の活用による情報伝達とによって作られているといえます。

ところで、例えばアルファベットという「記号」を知らない、あるいは記号のまとまりとしての単語や文法を知らない人にとって、アルファベットで書かれた英文の商品注文書

や小説は、それがいかに価値ある内容を持っていたとしても情報として活用したり、感動したりすることはできません。つまり、「記号」が価値を持つには、まず第一に、その記号を扱うものにとって解釈可能な形で表現されている必要があります。

情報社会では、人間の頭脳労働の一部を肩代わりをするコンピュータでデータが処理され、処理結果を人間が情報として利用しています。ところで、コンピュータは電気によって動いています。このため、コンピュータで情報処理させるには、情報やデータが電気的に取り扱えるように表現されていなければなりません。

この表現は、私たち人間が慣れ親しんでいる表現とは随分異なっています。しかし、そこで生じる技術上の問題の幾つかは、人間が文字や数字といった古くから利用されてきた「記号」を取り扱う際に経験してきたことに通じるものも多くあります。情報社会の本質を学ぶことは、一面、何気なく使っている文字や数字といった「記号」の本質を問い直すことでもあるのです。

【ちょっと気をつけて!】 ここでは記号を「人工的に発明された文字や数字など」としていますが、記号の意味を広く考えると、遺伝子や脳内の神経伝達物質、蝶の羽の模様など、情報のやり取りに関係するあらゆる現象を、情報関連学問の成果を通して考えることができるようになります。逆に、脳の研究成果や動物のコミュニケーションの研究が、新しい原理のコンピュータや情報ネットワークの研究に影響を与えています。このように、情報の研究は幅広い学問分野と関わりながら進められています。

実体としての「モノ」と抽象体としての「記号」

「古池や、蛙飛び込む、水の音」という芭蕉の有名な俳句の価値について考えてみましょう。今この俳句の価値を「俳句が表現している言葉(字句)の意味」という面に限定して考えれば、その俳句が口頭で伝えられようが、紙に筆で書かれていようが、活字印刷の本になろうが、言葉の意味としての価値に変わりはありません。つまり、言葉の意味としての情報的価値は、それを表現する媒体(メディア)によらず、何度も複製(コピー)でき、読めさえすれば多少の字の汚れは関係ないといった特徴を持っているのです。

一方、芭蕉直筆の俳句が書かれた短冊という実体に価値をおく人にとっては、俳句の言葉としての価値に留まらず、文字の形、文字の配列といった視覚的効果もまた芸術的骨董的価値を持っています。では、直筆の短冊の実体的価値、つまり書かれたメディアという実体まで含めた価値は、どのような特徴を持っているのでしょう。実体は他の実体に置き換えることができません。真似して作った偽物(贋作)は、本物の持つ実体としての価値の多くを失っています。また、短冊という実体は、作られた瞬間から劣化が始まるという宿命を持っており、保存状態が悪いと劣化は急速に進みます。

では、絵画や彫刻の場合はどうでしょう。俳句といった言語芸術と比べ、絵画や彫刻の芸術としての価値は、それらを形作る「モノ」という実体を離れては存在できません。つまり、芭蕉直筆の短冊と同様、絵画や彫刻を表現している実体そのものが価値の正体なのです。

音楽の場合はどうでしょう。作曲家は作品を楽譜に記録します。楽譜に残された音符という記号は、演奏家にとって作曲家の産み出した情報としての価値を持ちますが、聴衆に

とっての音楽の価値は演奏そのものであり、演奏が終わると音楽も消えてしまいます。音楽にとって楽器はその芸術的価値を生み出す「モノ」であるが、楽器は音楽そのものではなく、空気の振動という「モノの状態」が音楽の価値なのである。

以上見てきたように、これまでの私たちの文明や文化では、「記号」、「モノ」、「モノの状態」の差違が比較的明確でした。しかし、情報革命の進展に伴うデジタル表現された情報社会では、このような見方を超えた新しい表現世界を切り開きつつあるのです。

【演習】身近にある物を、それがもつ情報に意味がある物と、そうでないもの、どちらともいえない物などに分類してみましょう。自分の分類結果と他の人の分類結果とを比較してみましょう。

2.2 情報表現からみた記号の本質--文字と数字を振り返る--

コンピュータの内部では、コンピュータの特性にあった電気の状態で情報が表現されています。しかし、コンピュータに入力するデータや処理結果として出力されるものは、私たちが見慣れた文字や数字で出力されるため、普段コンピュータの内部での文字や数値がどのように表わされているかを意識することはありません。しかし、コンピュータに誤りなく計算させたり、また異なったコンピュータ環境で誤りなく文字をやり取りするには、コンピュータ内部での文字や数値の表現についてよく理解しておく必要があります。このことはまた、コンピュータで画像や音がいかにして処理されているかを理解するための基礎でもあるのです。

2.2.1 文字の発明と進化--象形文字、表意文字、表音文字--

本来文字は、「モノ」や「モノの状態」と「モノの関係」を記録するため、記録したい事柄と 1 対 1 に対応させる形で作られました。最初の文字は、記録したい事柄を表わす絵を図式化した象形文字といわれています。象形文字では、「太陽」や「木」といった名詞のみならず、「食べろ」とか「持ってこい」といった動詞、「大きい」「すぐに」といった形容詞や副詞など、言葉になるものすべての図式化が図られました。

文明や文化が発展すると、表現すべき事柄が増え、事柄ごとに対応した文字を作っていったのでは、人間の記憶能力を越えるという問題が発生します。この問題を解決するため、ふたつの工夫がなされました。

工夫の一つは、漢字の偏(へん)と旁(つくり)に象徴されるように、複数の記号を組み合わせることで文字数を減らすという方法です。同時に文字の形もより抽象化の進んだ形になり、一見しただけではその形が何に由来して作られたのかが分からないものや、また一つの文字で複数の意味を持つものも現れました。しかし、「文字の形が意味を持つ」という象形文字以来の本質的性格は変わっていません。このような文字は表意文字と呼ばれ、漢字はその代表例の一つです。

いま一つの工夫は、アルファベットや平仮名に代表されるように、「文字を要素として組み合わせ、その組み合わせに意味を持たせる」というものです。この場合、要素となる文字自体は、具体的な意味を持っていません。つまり、口からでる言葉と同じ仕組みになっているのです。このような文字は表音文字と呼ばれています。

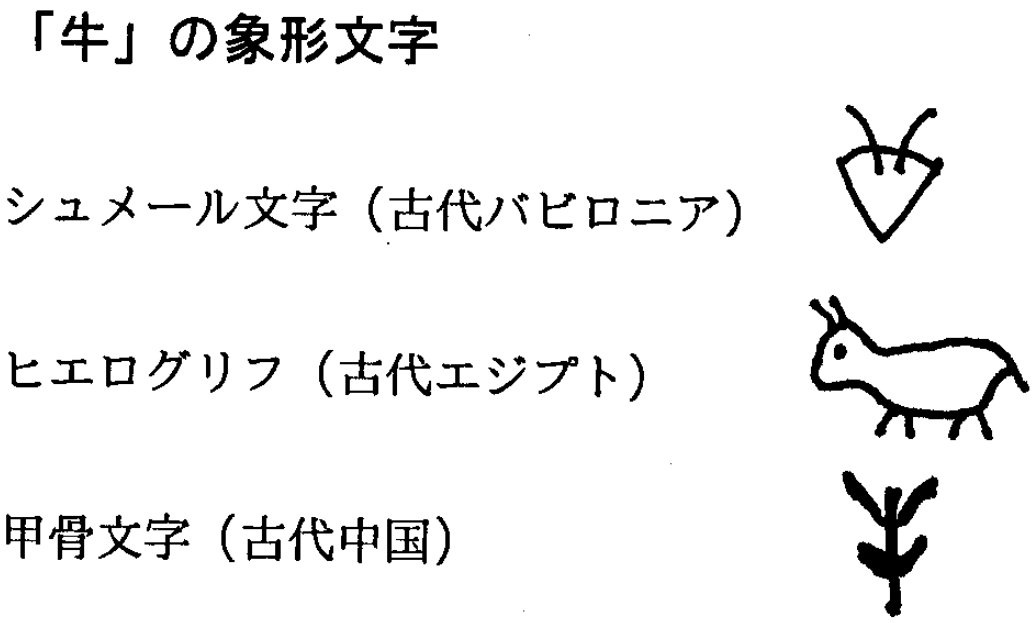


図 2.2:象形文字の例

表意文字、表音文字の利点と欠点

表意文字は形で事柄を表現できるため、一見して分かりやすく、また少ない文字数(場合によっては 1 つ)で豊富な内容を表現できるという利点を持っています。

【メモ】当て字という誤りが起こる原因が、表意文字の持つ文字としての表現力に起因しているといえます。また、中央教育審議会を中教審、大規模小売店舗法を大店法と意味を推測できるように略記できることにも関係していると言えます。一方、表音文字の短縮は、あたかも新しい単語を生み出すように、元の意味を推測できないことが多いものです。例えば、ANK は半角の英数字や記号、カタカナ文字の総称ですが、なぜ ANK か分かりません。実は、ANK は「Alphabetic Numeric and Kana」が元になっています。ちなみにペットボトル（PET bottle）の PET は「polyethylene terephthalate」という化学用語から来ています。愛玩動物のペットとは何の関係もありません。

【ちょっと気をつけて!】昔のコンピュータで「印刷せよ」という仕事をさせるには、「copy」といった命令をキーボードから入力するほかありませんでした。しかし、最近では画面に置かれたアイコン(Icon)という一種のボタンから「copy」に対応するもの探して押すだけで同じことさせることができるようになっていますが、これは象形文字への先祖返りともいえます。なお、たとえばキーボードから c,o,p,y と順に入力してコンピュータに印刷処理をさせる場合でも、コンピュータの内部では「copy」という文字列がそれに対応した命令コ

ードに変換され、そのコードをコンピュータの頭脳にあたる部分(CPU)に送られて処理されています。

このような利点を持つ表意文字ですが、表意文字には表現したい事柄が増えるにしたがって多数の文字記号が必要となるという欠点が避けられません。たとえば、漢字で書物を活字印刷する場合、多種多量の活字を用意しなければなりません。これに比べ、表音文字文化圏では、少数の単純な活字で済ませることができます。表音文化圏でタイプライターが早くから発明され、日常生活で広く普及した理由がここにあります。

コンピュータでの文字利用でも同じです。コンピュータで文字を処理する場合、コンピュータの内部に予め利用する文字と１対１に対応したコードをメモリーに準備(記憶)しておく必要があります。このため、コンピュータで漢字を利用するにはアルファベットに比べ膨大なメモリーが必要になります。いまでこそメモリーは、技術革新によって安価になりましたが、つい最近までメモリーはとても高価でした。このため、コンピュータでの漢字の利用は、アルファベットの利用に比べずっと遅れ、当初はよく使われる一部の文字(日本語ではカタカナ)だけが使われました。

【ちょっと気をつけて!】
近年、メモリーが安価になり利用できる漢字の数は随分増えましたが、未だすべての漢字を取り扱えるには至っていません。実は、漢字の追加にはいろいろと解決せねばならない問題があります。

先ず、どの文字を追加文字として選定するかがあります。より重要なのは、コンピュータ内での割り当てる際に既存の文字とどう調整するかということです。

アルファベットでは文字を組み合わせて意味を持たせるわけですから、そもそも必要なすべての文字記号がコード化されていました。その際、コードとの対応順序はa,b,c....x,y,z,A,b...といったように決められました。

漢字の場合、たとえば、亜、会、居、胃...というように「あいうえお」順にコード化していたとします。ここで、「医」という漢字を追加するとします。単純に亜、会、居、医、胃...というように「居」と「胃」の間に「医」を追加すると、古いコンピュータシステムで作られた文章ファイルを新しいシステムで使うと、「胃」で書かれた部分が「医」と置き換わってしまうのです。当然、それ以降のものもすべてずれてきます。かといって後のほうに単純に追加していったのでは、当初の「あいうえお」の順が崩れてしまうわけです。

この問題をうまく解決するため、国際的に調整会議が行なわれています。

2.2.2 数字と記数法、数値表現の発見と進化

数値を表わす記数法も文字同様、記録から始まりました。大昔は、取り扱う数値も小さな正の整数に過ぎず、記録したい整数値分だけ点や線を書き並べるだけでした。しかし、人間が瞬時に個数を識別できる能力は意外に少ない 4 個から 5 個程度のため、トランプに見られるように点や線の配置を工夫したり、5 とか 10 の固まりごとに新しい記号を作るという工夫がなされました。この段階での記数法は象形文字同様、単に表わしたい数値と数字を 1 対 1 に対応させるというものでした。このようにして作られた数字を「基本数字」と呼びましょう。

古代数字の例（その１）

図 2.3: 古代数字の例 1

束の考え方と非位取り記数法

記録すべき整数が大きくなると、少し工夫がなされました。たとえば「10 個の束」を作り、その束に「10 の位用の記号」を作り、さらにその束が 10 個集まった「100 個の束」に「100 の位用の記号」を作り...というように各位に対応した「位数字」が発明され、それらを必要な個数だけ書くという方法が取られました。

古代数字の例（その２）

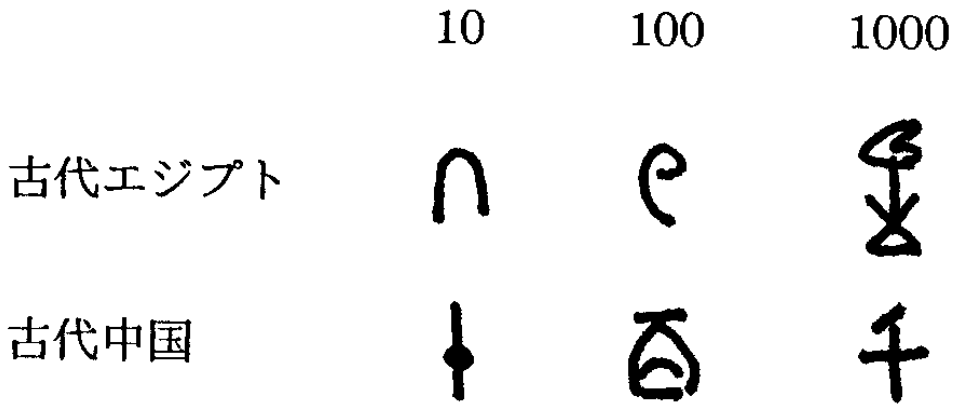


図 2.4: 古代数字の例 2:大きな数値

たとえば、685 を「百百百百百百十十十十十十十十五」と書くやり方です。このやり方では、財布の中のお金の様子を直接写し取ったものと同じで、具体的な個数との対応が見えやすいことや、足し算だけは容易に出来るという意味からか、長い間利用されてきました。

やがて、1 の位の「基本数字」とその個数を掛け算を利用して表現するという、今日の漢数字型の記数法が発明されました。この「必要な数字の個数を掛け算を利用して簡潔に数値表現する」という発想は、先で学ぶ「情報の圧縮」につながる発想の原点といえます。

「百百百百百百十十十十十十十十五」→六百八十五

しかし、この方法でも扱う数値がどんどん大きくなるにしたがって、どんどん新しい「位数字」を追加する必要があるという欠点は相変わらず残りました。

【演習】漢数字では何種類の位数字があるかを調べ、漢数字で表現できる最大の数値を書き表わしてみましょう。

【ちょっと気をつけて!】数値と数字

コンピュータの内部では、数字は文字の仲間として扱われており、計算される数値そのものは、文字とはまったく異なった扱いがなされています。詳しくは先で学びますが、「数値」と「文字としての数字」の違いをきっちり理解しておくことが重要です。

ゼロ（0）の発見と位取り記数法

いま私たちは、0 および 1 から 9 という十個の基本数字だけを位に応じた位置に置くことにより任意の数値を表現できる「位取り記数法」を多く使っています。この記数法の発見は他の記数法より遅れました。その最大の理由は、ゼロ(0)の発見が他の数字に比べ大幅に遅れたことによります。

数字に限らず、文字は本来「存在するモノや事柄に対応させて記号を作り記録する」という必要性から発明されました。このため、ほとんどの文明圏では、「何もない」という状態を記録するために数字、つまり 0(ゼロ)が作られませんでした。

今日に繋がるゼロ(0)は、インドで発見され、やがて位取り記数法が生まれました。位取り記数法は、アラビア文化圏を通してヨーロッパにもたらされ、金利計算に極めて便利なため、商業活動の広がりと共に普及し、筆算技術の改良と結びついて近代科学の発展に重要な貢献をすることになったのです。このような数字は特に算用数字と言われています。

表現したい任意の数値を、十個という限られた数字の配列で表現するという位取り記数法の考え方は、アルファベットの組み合わせで任意の単語を表わす表音文字といえます。位取り記数法には表音文字と同様の利点と欠点が存在します。たとえば、500003062 とあった場合、一目で「五億三千六十二」だと読み取るのは難しいことです。

記数法と計算

私たちは数字を使って計算(筆算)するのが普通と思っています。しかし、数字が発明されたとき求められた機能は記録のため、数字に対して筆算機能はさほど重要視されませんでした。計算の工夫は、むしろ計算を補助する道具(たとえばソロバン、計算棒)にあったとい

えます。いま私たちは電卓や表計算ソフトといった計算道具(機)で計算を行ない、必要な結果を利用するということが多いですが、道具の力を借りて計算し、結果を記録するというやり方はなんら現代社会の特徴的ではないのです。

このことから分かるように、算用数字の十進位取り記数法が広く世界に普及して行った最大の要因は、数字を使って簡単に計算できる（筆算）ことにあったといえます。さらに、わずか十個の数字でどんな小さな数から大きな数でも表わすことができるため、特に自然科学の研究では欠かすことのできない記数法になりました。なお、算用数字を用いた十進位取り記数法の普及は、数値記録の歴史から見ればごく最近のことといえます。

【演習】漢数字を使って計算、特に掛け算や割り算をやってみよう。

【演習】計算の歴史を、計算手順(アルゴリズム)に着目しながら調べてみよう。

【ちょっと気をつけて!】算用数字が普及する以前は、今日の目から見るととても奇妙に見える指を操る計算方法とか、計算結果をあらかじめ示した表の利用などの計算の手順(アルゴリズム)が多数考えられ、教育されていました。

【メモ】算用数字は字形が単純で、また位取り記数法では数字を追加して簡単に数値を書きかえることができます。つまり、算用数字を用いた手書きの十進位取り記数法は書き換えによる偽造の問題があります。実際、アラビア文化圏からヨーロッパに算用数字がもたらされた当時、偽造がはやったため「算用数字使用禁止令」が出たほどです。今日でも登記簿や借用書などの大事な記録では漢数字、それも一、二、三を壱、弐、参と書く場合があります。

2.3 二進デジタル表現への道

--情報社会を支える情報の表現とは--

コンピュータは、文字や数字、そして数値、さらに音や映像などのすべてのデータを「電流が流れる・流れない」「磁気が N 極が S 極か」といった単純な電磁気現象の２つの状態の変化を利用して、情報処理しています。より正確に言うと、データが２つの状態の組み合わせによって表現され、記憶され、さらにコンピュータにデータを加工したり計算させたりするための指示（命令）も 2 つの電磁気現象の組み合わせで行なわれています。

ところで、私たちを取り巻く自然は無数の色や音に満ちており、普段扱っている文字やそれを組み合わせた言葉にも多数の種類があります。また、数値は理論的には無限に存在し、数値修理には複雑な計算を伴うものもあります。このような多くのことが単純な 2 つの電磁気現象で扱えるのはとても不思議な気がします。

2.3.1 文字や数字の個数を決めるもの

私たちが扱う文章に現れる文字の種類は幾つぐらいあるのでしょう。例えば英文は、26 個(大小あわせれば 52 個)のアルファベットと.や?、!などの幾つかの記号で出来ています。算用数字は 0 を含め 10 個です。そして日本語では、それらのほかに何万のもの漢字が使わ

れています。しかし、文章を記録したり数値を扱うためには、必ずしも 26 個や 10 個である必要はないのです。いまある記号の数を決めてきたのは、人間の記憶能力や五感に関係しており、したがって人間でないコンピュータで情報を取り扱うには、日常利用する文字の個数とは無関係に考えることが出来ます。このことを、文字数を増やしたり減らしたりしながら考えていきます。

文字記号を増やした世界

例えば、アルファベットに 27 番目としてθという記号を追加し、英語でよく出てくる th に当てはめると、英単語はもっと簡潔に短く表現できます。

this → θis there → θere three → θree

数の場合も 9 の次に△▽といった数字を追加し、12 で繰り上がる十二進法を位取りの基準にすると次のような表記ができます。

8 個 9 個 10 個 11 個 12 個 13 個（十進法）→ 8 個 9 個△個▽個 10 個 11 個（十二進法）

このような表現が奇妙に思われるのは単に慣れだけの問題に過ぎず、歴史を調べてみるとこのような表現上の奇妙な表現の例は多数に見つかります。実際、英語の数字の読み方にはかって十二進法が使われていた痕跡が残っていますし、フランス語の場合はすっきりした十進の読みに慣れた日本人から見ると、とても複雑で奇妙なものです。

(例) 数の読み方

日本語

十進位取りの表記と対応した読み方「じゅう、じゅういち、じゅうに」

英語

表記は十進位取り、読みは十二進法的な「ten、 eleven、 twelve」

仏語

70（soixante - dix ）日本語で訳せば「60 と 10」

80（quatre - vingt）日本語で訳せば「4 かける 20」

【演習】いろいろな国の命数法を調べてみましょう。そこからかってどのような文化の交流があったかについて考えてみましょう。

文字記号を減らした世界

文字や数字の数を減らせばどうでしょう。例えば英字アルファベットから z を省略します。この場合、たとえば z の含まれている単語は残りのアルファベットの組み合わせ(例えば aa に置き換えればいいわけです。そして出来上がった単語が既存の単語と一致していなければそれを新しい単語表現と決めればよいのです。

zoo(動物園) → aaoo
zero(零) → aaero
fuzzy(あいまい) → fuaaaay

一見ややこしそうに思えますが、それは既に存在する英語をイメージするからで、最初からアルファベットが 25 文字しかないと思えば、そんなものと了解できるはずです。実際、26 文字より少ないアルファベット系の言語も存在します。

数値の場合は、例えば 0、1、2、3 の 4 つの数字だけを用いて 4 になると繰り上がる四進法にすると、次のような表記が考えられます。

2 個 3 個 4 個 5 個 6 個（十進法）→ 2 個 3 個 10 個 11 個 12 個（四進法）

単語の場合も数値の場合も、元になる文字や数字を減らすと表現は長くなり判断に時間はかかりますが、憶える記号の数は少なくてすみます。一方、記号を増やすと表現は短く簡潔にできますが、こんどは記憶の手間が増えてしまいます。このように「一方を立てれば他方がたたない」という関係をトレードオフ(trade off)といいます。トレードオフは、コンピュータサイエンスに限らず、多くの分野で見られます。

情報の表現に必要な最小の記号数

情報を表現するための記号は、どこまで減らせるのでしょう。単純に考えると記号はただ一つでよいように思えます。例えば、表現したい文字が N 個ある場合、基本記号をひとつ用意し、それが 1 個なら一番目の文字、2 個なら二番目の文字...、N 個なら N 番目の文字というように対応させていくのです。いま基本記号をピーという 1 秒間の音とする。以下、2 秒間のピー...N 秒間のピーというように、長さの異なったピーと文字とを、図のように対応させてきます。図の-が 1 秒のピーを表わしています。

- → 水
-- → 島
--- → 男

図 2.5: 音を利用した一つの記号による表現例:漢字の場合

小文字 : a(-)b(--)c(---) ～z(26 個の-)
大文字 : A(27 個の-)B(29 個の-) ～Z(52 個の-)

図 2.6: 音を利用した一つの記号による表現例:アファベットの場合

このようにしてアルファベットの大文字小文字は、一つのピー(-)で表わすことができま

した。しかし、この方法で単語を作ろうとすると、次のような大きな問題が起ります。つまり、「---」とあった場合、それが、aaa か ab か ba か c かの区別がつけられないからです。このことの解決を考えてみましょう。

その 1:分離記号という考え方による解決

個々の文字を他の文字と分離するため、もう一つの記号を導入します。それをポーとし、・で表わします。こうすると、ab は「-・--」と表現することができるわけです。

次に、単語を続けて文章を作ることを考えます。英語はアルファベットの組み合わせで単語を作る表音文字のため、各単語を「分かち書き」する必要があります。かりにその文章を、「bc a abc」とします。ちょっと考えると、ポー(・)を使ったのだから、単語と単語の分離(分かち書き)には空白(スペース)に対応する第三の記号パー(*)が必要に見えます。つまり、「--・---*-*-・--・--」というわけです。しかし、個々の文字を区別する区切りはポー(・)を一つ、単語の区切りにはポーを二つ並べるとするのです。こうすると、「bc a abb」は「--・---・・-・・-・--・--」というように、2 つの記号で表現できました。

【ちょっと気をつけて!】 私たちは、英文に見られる単語間の空白(スペース)を単なる「紙面の地肌」と思いがちですが、情報の表現から見れば、空白は単語を区切り文章を作る際の重要な分離記号(文字)なのです。さらに、アルファベット系の文章は、表音文字だけで出来ていると思いがちですが、そうではありません。空白やピリオド(.)、クエスチョンマーク(?)などは表意文字なのです。

【演習】図 2.6 の例に示したアルファベットの-表現と、区切りのルール(・および・・)を用いて、This is a pen を表現してみましょう。

【演習】図 2.6 のように、いま a に 1 個の-、b に 2 個の--という対応させてアルファベットを表現するやり方は、文書作成時の手間や保存、伝達の際の経済効率が悪いという欠点があります。その理由を考えなさい。(ヒント、出現頻度、モールス信号の工夫)

その 2:一定個数の記号の並べ方パターンによる解決

いま、N 個の場所を用意し、その場所に-と・のどちらかを置く場合、幾つのパターンが出来るか考えてみましょう。

場所が 1 つのとき-と・の 2 通り

場所が 2 つのとき--と-・と・-と・・の 4 通り

場所が 3 つのとき---と--・と-・-と-・・と・--と・-・と・・-と・・・の 8 通り

一般に、場所が N 個あると、-と・を使ったパターンは 2^N 通り出来ます。

【演習】場所が N 個あると、-と・を使ったパターンは 2^N 通り出来ることを証明してみよう。

次に、N 個数の場所に置いた場所-と・を置くことで出来る 2^N 個の異なったパターンに各

種のアルファベットや数字、文章の終わりのピリオド(.)やクエスションマーク(?)、+、-、空白などといった特殊な記号を対応させて割り当てることを考えます。では、一体 N としてどのような値を取ればいいのでしょうか。

アルファベットの小文字大文字は全部で 52 個、数字は 10 個、これに特殊記号を含めると、通常の英文は 100 種類ほどのパターンが必要となります。100 種類程度のパターンを作るために必要となる N の値は、次の不等式を満たす最小の整数解を求めることで分かります。

$2^N>100$ を満たす最小の整数 N とは?

答えは 7 で、可能なパターン総数は 128 個です。

【演習】-- -- -- --をパターンの最初、・・・・・・・をパターンの最後として、128 個の全パターンを書き漏れないように工夫しながら、実際に書いてみましょう。

このようにして出来る 128 個のパターンの一つ一つにアルファベットや空白などの特殊記号を対応させておくと、任意の文章を書くことが出来ます。その際、個々のパターンを空白で区切る必要はありません。なぜなら、読み取るときには先頭から 7 つづつに区切って読めばいいのですから。

このように、決まった個数 N の場所においた-と・のパターンで表現する方法では、各文字のパターンの長さ(-と・をおく場所の個数)が決まっているため、固定長による表現といいます。

固定長で表現された文章中の-と・の総数は、「文字数×N」という簡単な計算で求めることができます。一方、先に示したよう分離記号を利用した表現方法は、文字によって与える記号数が異なるため、-と・の総数は、簡単な「文字数×N」では決められません。一連の情報の切れ目を便利記号で行う方法は、可変長といいます。

固定長と可変長という考え方は、コンピュータ利用の多くのところで見られます。

ビット

文字や数字のように記号化できる情報は、2 つの異なった状態をとるものを使って表現できることが分かりました。この「2 つの異なった状態をとるもの」をビット(bit)といい、情報科学の立場で情報の大きさ（量）を計るための最少単位となっています。bit という言葉は、「2 進法のひと桁」(binary digit)から来ています。

【ちょっと気をつけて!】 情報科学や情報工学で「情報の大きさ（量）」という場合は「その情報を表すのに何ビット必要か」ということで、その情報が人間にもたらす価値としての大きさではありません。

たとえば電子メールで送られてきた、「論論み花花」と「明日は休校」があったとします。この２つは共に５文字ですから、情報科学や情報工学的には「同じ大きさの情報量」です。しかし、もたらされた情報の「価値としての大きさ」はまったく異なります。

これから先で出てくる「情報量」という言葉はほとんどが情報科学や情報工学の意味で使っています。

ビットという言葉を使って、これまで述べてきたことをまとめると次のようになります。

1. 記号化できる任意の情報は、文字や数字によって表現できます。
2. 文字や数字は、幾つかのビットによって表現できます。
3. ビットをつかって文字や数字記号を表現する方法には、異なるビット数(可変長)を文字や数および分離記号を対応させる方法と、一定個数のビットの組み合わせパターン(固定長)に記号を対応させる方法があります。

コンピュータは、ビットで表現される情報を電磁気現象で操作している機械なのです。

2.4 コンピュータ内部での情報の取り扱い

さまざまな情報をビット列に対応させて表すことを符号化(コード化)といい、その対応づけの規則を符号化規則といいます。たとえば「雨が降っていない」を 0、「雨が降っている」を 1 であらわす、というのも符号化規則です。

【用語】符号化:情報をビット列に対応させて表すことを符号化といいます。

【ちょっと気をつけて!】 情報の送り手と受け手で同じ符号化規則を使っていないと、情報が正しく伝わりません。このため、情報の種類ごとに標準の符号化規則をさだめて、共通に使うようにするのが一般的です。

実際の文字や数値をビットのパターンに対応させる際、ある規則性を持った形で作ったビットパターンに通し番号(順序数)をつけ、次にその番号順に決まった文字や数字を対応させると、数学の知識が使えるので何かと便利です。このようなことから、コンピュータでは、-と・といった記号ではなく、0 と 1 という 2 つの数字だけで数値を現すことができる二進位取り記数法(簡単に二進法と書く)を基にビットが取り扱われています。具体的には、「電流が流れている、流れていない」、「0 ボルトと 5 ボルト」、「磁石の N 極と S 極」といった 2 種類の電磁気現象で 0 と 1 を扱っています。

2.4.1 二進位取り記数法の原理

文字や数値を二進法で取り扱うことを学ぶには、まず二進法の基本を知っておく必要がありますので、簡単に見ておきましょう。

私たちは、二百五十六という 3 桁の数を十進法位取り記数法では 256 と表わしますが、その意味は次の通りでした。なお、十進法位取り記数法を簡単に十進法と書くことにします。

$(256)_{十}= 2\times 10^2+ 5\times 10^1+ 6\times 10^0$

一般に、n 進法で f 桁の数値(abcde)n をべき乗表現は、次のようになります。

$(abcde)_n= a\times n^4+ b\times n^3+ c\times n^2+ d\times n^1+ e\times n^0$ 式 1

ここで、a、b、c、d を仮数、n を基数といいます。仮数は、n 種類の数字で、そのうち一つは 0(ゼロ)でなければなりません。

【演習】二進法で 100111011110 と表される数値を、十進法で書きなさい。(ヒント、式 1 で n を 2 として計算します。答 2526)

【演習】十進数で 536 という数値を二進数に直すには、536 を 2 で割った余りを並べていけばできます。この方法で任意の十進法を二進法に直せることを示しなさい。

```
2)  53
 -----
2)  26    ...  1↑
 -----          ↑
2)  13    ...  0↑
 -----          ↑
2)   6    ...  1↑
 -----          ↑
2)   3    ...  0↑
 -----          ↑
     1    ...  1↑
     →→→→→
```

(答)$(110101)_{二}$

図 2.7: 二進十進変換

$$
\begin{aligned}
(110101)_{二} &= 1\times 2^5+1\times 2^4+0\times 2^3+1\times 2^2+0\times 2^1+1\times 2^0 \\
&= 32+16+0+4+0+1 \\
&= 53
\end{aligned}
$$

(答)$(110101)_{二}$

図 2.8:十進二進変換

【演習】ここに示した二進法から十進法への変換や、演習問題 b の十進法から二進法への変換方法は、正確には「十進法を基準とする立場にたって、変換」です。「二進法を基準とする立場」に立つと変換方法も変わってきます。「二進法を基準とする立場」での変換方法を考えましょう。
(ヒント:図 2.7 や図 2.8 の計算は十進の「九九」や加減が基準となっています。「二進法を基準とする立場」なのですから、「九九」は「二二」に、加減も二進法のルールで行います。)

答、図 2.7 と図 2.8 のアルゴリズムが逆転する。

二進法と十六進法

使う数字の少ない二進数は、十進数に比べ同じ数値を表現するのに桁が長くなります。このことは、書物に書いたりする上で不便であり、読み間違いも起こりやすくなります。このため、コンピュータや通信関係では二進法そのものでではなく、十六進法が使われるのが普通です。十進法ではなく十六進法が使われるのは、二進法と十六進法の変換方法が二進法と十進法の変換に比べ簡単にできるからです。同様の理由で八進法もよく使われます。

十六進法では 16 個の数字が必要となります。その際、0(ゼロ)から 9 までは算用数字を使い、それ以上はアルファベットの A から F を利用するのが一般的です。

次の図 2.9 は、ゼロから十六までの数を十進法と二進法および十六進法で現したもので、ビットパターンが見やすいように各位の 0 を省略せず書いたものです。

漢数字	十進	二進	十六進
零	00	0000	0
一	01	0001	1
二	02	0010	2
三	03	0011	3
四	04	0100	4
五	05	0101	5
六	06	0110	6
七	07	0111	7
八	08	1000	8
九	09	1001	9
十	10	1010	A
十一	11	1011	B
十二	12	1100	C
十三	13	1101	D
十四	14	1110	E
十五	15	1111	F

図 2.9: 数の相互変換

二進数と十六進との相互変換

10011100110 の場合

1. 与えられた二進数を、 下の位から 4 つずつのグループに分ける。 最上位のグループでは 4 つにならないことがあるが、 その場合は 0 を補う。 0100、 1110、 0110。
2. 各グループを、 対応表をもとに十六進の数字で置き換える。 0100→4、 1110→E、 0110→5。 答 4E5
3. 十六進法から二進法に変換するには、 逆の置き換えをすればよい。

このように、 十六進法と二進法との変換は、 十進法と二進法の変換に比べずっと簡単です。

【演習】このことを証明しなさい。
ヒント: $16 = 2^4$
$1\times 2^6 + 0\times 2^5 + 1\times 2^4 + 1\times 2^3 + 0\times 2^2 + 1\times 2^1 + 1\times 2^0$
$=(1\times 2^2 + 0\times 2^1 + 1\times 2^0)\times 2^4 + 1\times 2^3 + 0\times 2^2 + 1\times 2^1 + 1\times 2^0$

【ちょっと気をつけて!】 ビットによる情報の表現の考え方自体を学ぶには、十六進法の知識は必ずしも必要ではありません。しかし、現代のように変化の激しい時代では、学校を卒業した後も生涯に渡って新しい事柄を学び続ける必要があります。十六進法に慣れておくことの意義はここにあります。

2.4.2 文字の表現の実際

コンピュータで文字や数字のような記号を扱う第一歩は、各文化圏で使われている記号の内、どの記号を選ぶかから始まります。

次に、選んだ記号ごとに適当なビット列に対応づけて符号化されます。この対応のことをコード系(character encoding system)、対応を示した表を、コード表と言います。

文字や数字の符号化では、各人が勝手な符号化をさせるとソフト相互のデータのやり取りや通信において混乱か起こることが容易に分かります。これを防ぐため、文字規格が幾つか提唱されました。代表的なものに ISO/ITU-T(国際標準化機構/国際電気通信連合)や JIS の規格、パソコンで主に使われる ASCII(American Standard Code for Information Interchage)等があります。

【ちょっと気をつけて!】同じ日本国内でも複数の規格が存在します。それは、過去において性能が大きく異なったパソコン、汎用機、ワークステーションというコンピュータ商品群が比較的独立したそれぞれの文化圏を作っていたためです。しかし、異なる文字コードを通信用に使っているコンピュータ同士では、ネットワークを用いた情報交換はできません。このため、どうしても情報交換を行なう必要がある場合には、通信の際に異なった規格間で文字コードの変換を行いながら通信されています。

規格に関するいま一つの問題は、世界には多数の言語圏があり、独自の文字文化を持っていることです。パソコンの能力が飛躍的にあがった今、これに関しても統一規格が話し合われていますが、ことが文字という文化の根幹に関わることのため、なかなか難しい問題があります。

【演習】日本語の五十音 (ゃ、 ぅ、 などの促音も含めます) や句読点や他の記号のそれぞれに適当な番号を振って、 コード表を作ってみましょう。 また、 自分のコード表を、他の人が書いたものと比較してみましょう。

符号化の実際　その１　--ISO/ITU-T7 ビット ASCII による英文字--

現実の文字や数字の符号化のやり方と工夫を知っておくことは、ネットワーク社会の諸問題を考える上で重要な知識となります。漢字は個数が多いため、コード表の全貌を見渡すことが大変ですので、英文字のコード表で符号化の実際を調べていきます。

次の表は 7 ビットの ISO/ITU-T7 ビット ASCII の例です。

この表で「列(C)」と「行(R)」と書いてある行と列の記号は十六進法表示したビットパターンを示しています。この表より、A は十六進法で(41)、二進法で(1000001)というビットパターンであることが分かります。表で SP とあるのは、空白(スペース)を、何も書いていないところは対応させる記号が不要のための未使用領域を示します。

また、0 列と 1 列には NUL とか CR といったものや、F 行 7 列目には DEL というものがありますが、これらに文字ではなくコンピュータに特定の機能を実行させる役割が割り当てられています(機能コード)。例えば CR に対応する十六進法の(0D)、つまり二進法の(1001101)は、「改行せよ」とコンピュータに命令することにあたるわけです。機能コードは、コード系の切り替わるところを示す制御文字列としても利用されています。

				b7	0	0	0	0	1	1	1	1
				b6	0	0	1	1	0	0	1	1
				b5	0	1	0	1	0	1	0	1
b4	b3	b2	b1	行(R) ＼ 列(C)	0	1	2	3	4	5	6	7
0	0	0	0	0	NUL	DLE	SP	0	@	P	`	p
0	0	0	1	1	SOH	DC1	!	1	A	Q	a	q
0	0	1	0	2	STX	DC2	"	2	B	R	b	r
0	0	1	1	3	ETX	DC3	#	3	C	S	c	s
0	1	0	0	4	EOT	DC4	$	4	D	T	d	t
0	1	0	1	5	ENQ	NAK	%	5	E	U	e	u
0	1	1	0	6	ACK	SYN	&	6	F	V	f	v
0	1	1	1	7	BEL	ETB	'	7	G	W	g	w
1	0	0	0	8	BS	CAN	(	8	H	X	h	x
1	0	0	1	9	HT	EM	)	9	I	Y	i	y
1	0	1	0	A	LF	SUB	*	:	J	Z	j	z
1	0	1	1	B	VT	ESC	+	;	K	[	k	
1	1	0	0	C	FF	FS	,	<	L	|	l	
1	1	0	1	D	CR	GS	-	=	M	]	m	
1	1	1	0	E	SO	RS	.	>	N	^	n	
1	1	1	1	F	SI	US	/	?	O		o	DEL

図 2.10: ISO/ITU-T7 ビット ASCII

なお、7 ビットの ASCII ではフランス語やドイツ語のアルファベットが扱えないため、あと 1 ビット加えた 256 種類の記号が表現できる 8 ビットの拡張 ASCII というものも使われています。8 ビットは二進法との相性もいいため、1 バイト(Byte)という単位名で呼ばれています。

英文字だけの場合でも、8 ビットを使うと情報を取り扱う際の信頼性を上げることができ

ます。

【ちょっと気をつけて!】　一つの英文字を表わす単位に８ビット集まったバイトという単位を作りました。ところで、私たちが通常扱う文章は多くの文字から出来ていることが普通ですので、ビットやバイトを単位としてその大きさを指定していくと、とても大きな数値になってしまいます。先で詳しく学びますが、音や画像をビットで取り扱うと、さらに膨大になります。このため、メートル法で 1000m（メートル）を１Km（キロメートル）というように、情報の分野でも大きな単位が使われます。さて、私たちが普段使う十進法では、基準の千倍の単位を K（キロ)、その千倍を M(メガ)、さらにその千倍として G（ギガ）という単位を使っています。ちょっと考えると、情報や通信の分野でも十進法のｋ（キロ）や M（メガ）をそのまま使えばよさそうですが、情報では二進法を使うため、きっちりと千倍というわけには行きません。そこで、千倍に近い数値として、２の１０乗の 1024 ビットが次の単位として使われ、以下、その 1024 倍が次の単位として使われています。ややこしいことに 1024 倍も、その 1024 倍も十進法と同様、K、M という記号が使われています。この混乱を避けるため、情報の場合は K をキロではなくケーと呼びます。このことは案外知られていません。カタログなどに１M ビットのメモリーとあった場合は、正確には１００万ビットのメモリーではなく、1024x 1024 の 104 万 8576 ビットのメモリーなのです。

パリティ・チェック

電子回路は極めて精度高く作られていますが、それでもビットパターン伝送の途中などで、特定のビットが 1 から 0 に置き換わってしまうということが起こり得ます。処理すべきビットパターンが正しいか否かを調べるため、本来の文字のためのビットパターンにチェック用のビットを追加するという方法が考えられました。つまり、情報表現に余裕(冗長性)を持たせるのです。この内、もっとも簡単なのがパリティ・チェックと呼ばれています。

パリティ・チェックの考え方は、元のデータの先頭に 1 つのビット(パリティ・ビット)を追加し、データのビットパターンにある 1 の個数が偶数なら 0、奇数なら 1 というようにしてデータを扱う方法です。ASCII の場合では「パリティ・ビット 1 ビット+文字コード 7 ビット」の合計 8 ビットを単位として文字コードを伝送し、受け取った側は図 2.11 のように、文字部分の 1 の偶奇を調べ、パリティ・ビットと一致していれば「誤っていない」、一致していなければ「誤っている」と判断するのです。

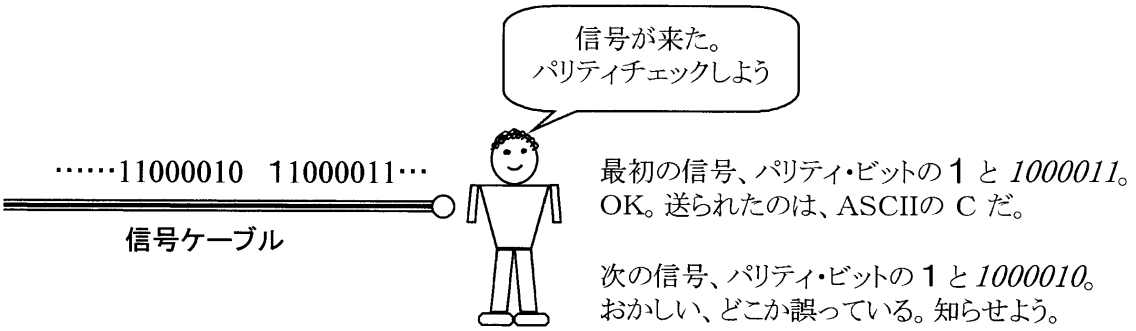


図 2.11: 1010011 が送られてきたぞ。

文字ビットの 1 の数は 3 の奇数、パリティ・ビットは 1。正しい C だ次は 1010010 だ。
文字ビットの 1 の数は 2 の奇数、パリティ・ビットは 0。おかしい、誤っています。
単なるパリティ・チェックでは、偶数個所で誤りがあると誤りは検出できず、また誤ったのがどの位のビットかの判断はできません。しかし、一個所が誤る確率を P(現実にはかなり小さい)とすると、2 個所同時に誤る確率は P^2 程度になるため非常に小さくなるので、簡便なパリティ・チェックはよく使われています。なお、数値の伝送時に、例えば 10 個おきにそれまでの数値の合計を送って誤り検出力をあげるといった特別な工夫をすると、どの数値が誤って伝送されたかの判定もできます。しかし、誤りの検出力を増やすには取り扱うべき情報の量(ビット数)が増えるというトレードオフが起こってしまいます。

【演習】次の文字列を、ASCII 文字コード表にしたがって数の列に置き換えて、十進数で表現してみましょう。

Ima no Kion wa, 20 Do Desuka?

符号化の実際その 2--半角カタカナ--

情報処理機械としてのコンピュータの価値が社会に認識がされるにつれ、日本語のコンピュータ処理が求められるようになりました。しかし、初期のコンピュータは処理スピードも遅く、またメモリーも高価だったため、日常普通に使う漢字をすべて扱うことは費用の面で困難がありました。そこで、先ずカタカナと日本語に特有の、「、」、「。」といった記号の利用から始まりました。カタカナだけなら、アルファベット同様 7 ビットで扱えるからです。
この当時に決められた符号化されたカタカナは、今日「半角カタカナ」といわれています。「半角」と「カタカナ」でわかるように、今日ワープロや通信で通常使われている日本語漢字(全角)の半分の幅しか持たないカタカナと関連する文字を指します。

【ちょっと気をつけて!】全角漢字は 16 ビットで表現されています。半角カタカナと全角カタカナは、画面や紙面では似た形をしていますが、符号化からいえばまったく異なったものです。先で詳しく見るように、混ぜて使うことには大きな問題があります。

半角カタカナの表現 JIS7 ビットコードと JIS8 ビットコード

半角カタカナの具体的な取り扱い方法として 2 つのやり方が取られました。一つは、分離記号の発想を使ったものです。まず、ASCII の 2 列目から 7 列目のビットパターンに、例えば、$(41)_{十六}$は A と半角チというように、同じビットパターンにアルファベット系とカタカナ系のふたつの記号を割り当てます。そしてアルファベット系とカタカナ系の判別は、ASCII の 0 列と 1 列に割り当てた機能コードの$(OF)_{十六}$の SI と$(0E)_{十六}$の SO で挟まれたものをカタカナ系、挟まれていないとアルファベット系とするのです。

例)AB チツ AB の表現

7 ビットカタカナ始り(0F)　7 ビットカタカナ終り(0E)
A　B　チ　ツ　A　B
(41)　(42)　(0F)　(41)　(42)　(0E)　(41)　(42)
(41)等は十六進法表現

いま一つの考え方が ASCII コード表を、7 ビットではなく、256 のビットパターンが作れる 8 ビットに増やしてカタカナ系も使えるようにしたものです。この発想は固定長の発想と同じです。具体的には、7 ビット ASCII の最上位にもう 1 ビットを加え、例えば、$(01101000)_{二}$が h、$(01101000)_{二}$がネ(半角)というように、付け加えたビットが 0 のときは、ASCII と同じ割り当てを行い、1 のときはカタカナや日本語に特有の、「、」、。、などを表すと約束するのです。

【演習】JIS7 ビットと JIS8 ビットのコード表を探し、カタカタ等が実際どのように割り付けられているか調べてみましょう。

符号化の実際その 3--全角漢字--

コンピュータの高機能低価格化が進むと、半角カタカナ以外の漢字の利用が強く求められるようになってきました。ここで漢字とは文字通りの「漢字」から「ひらがな」なども含みます。

さて、漢字利用のための符号化は、多数の文字記号がある漢字ではそう簡単ではありません。日本だけに限っても、漢字の種類を歴史的にさかのぼって拾って行くとその数は何万個にもなりますが、その大多数は私たちの日常生活ではほとんど使われていません。したがってすべての漢字を符号化してからコンピュータに乗せるというのは、実用的にも費用の面からも得策ではありません。そこで先ず、実用的に困らない程度の漢字符号化が行なわれました。

この作業には二つの解決しておくべき問題が起こります。一つは漢字の選別作業で、いま一つは選別した漢字をどのような順序で符号化するかという問題です。漢字には音読みとか訓読みといったように複数の読み方を持っています。このため、アルファベットのａ、ｂ、ｃように単純ではありません。このようなことを考えながら先ず最初に作られたのがコード体系が JIS 第一水準という漢字群です。

JIS 第 1 水準では普通の文章を書くにはそれほど不便はありませんが、事務作業で漢字を扱うと案外足りないものが出てきます。その典型的なものに人名や地名に使う漢字があります。そこで漢字の追加が行なわれました。このようにして追加された漢字を JIS 第 2 水準の漢字といいます。それ以降も幾つかの漢字が追加されています。

漢字の追加では、すでに社会で普及している文字とビット列との対応関係とどのように調整するかという問題も起こります。つまり、従来使われていた漢字のコードを追加する

漢字で利用し、元の漢字を別のコードに移し変えたりすると、混乱が起こるからです。

JIS 第 1 水準と第 2 水準の実際

日本語処理は 16 ビット=2 バイト(65536 種類のビットパターン)で漢字、(カタ)仮名、アルファベットや各種記号を扱かっています。16 ビットで符号化された漢字を 7 ビットや 8 ビットの半角のアルファベットと区別して全角漢字と呼びます。現在 JIS で決められている漢字の種類は、使用頻度の高いものを集めた JIS 第 1 水準 2965 文字とその他の第 2 水準 3388 字の合計 6353 字に過ぎません。これらの漢字の取り扱い規格として良く使われているものに次のようなものがあります。

- JIS コード
- シフト JIS コード
- 日本語 EUC コード

なお、通信ネットワークの普及に従い、漢字をより統一的に取り扱う規格として、Unicode とよばれる新しい文字コードが ISO で作られました。今後広く普及すると考えられています。

【演習】各種の漢字コードを調べて、どのようなルールで符号化されているかを調べてみましょう。

【演習】順不同に並んでいる人名などを「あいうえお順」に並べ替えることを「並び替え（ソート)」といいます。データベースや表計算ソフトで漢字で書かれた人名をソートすると、うまく「あいうえお順」に並ばないことがしばしば起ります。調べた漢字コード表をもとにその理由を考えましょう。また、正しく「あいうえお」順にならべるためにはどうすれば良いかも考えましょう。

2.4.3 通信における文字符号化の諸問題

受け取った電子メールを読もうとしたとき、時々、わけの分からない文字列の羅列が並んでいて意味が分からないことがあります。このような現象は、文字化けと呼ばれています。

文字化けが起こる原因には２つが考えられます。一つは、単純な漢字だけで出来ている電子メールが文字化けする場合です。この原因は、発信された通信文の漢字が別のコードの文字と置き換わることに起因しています。いま一つは、近年は電子メール機能を使って画像や表計算ファイル、あるいは装飾の入ったワープロ文章を「添付ファイル」という方法でやり取りする場合に起こる場合です。電子メールは本来、単純な文字コードの通信から始まりました。この本質はいまも変わらず、画像などの添付ファイルも伝送時はいったん文字コードに変換されて通信されています。

添付ファイルは「添付ファイル→変換プログラム→文字コード→復元プログラム→添付ファイル」という手順で伝送されますが、復元プログラムが適切でない場合に起こります。また、添付ファイルはそれを作ったワープロや表計算ソフトと同じソフトを相手が持っていないと、たとえうまい復元プログラムがあっても受信者は添付ファイルを見ることが出来ません。これは文字化けとは別の問題といえます。

ここでは文字だけの電子メールで起こる文字化けについて文字化けはなぜ起こるのか、そして自分が文字化けを起こす可能性のある電子メールを出さないためにはどのような点に注意しなければならないか、について学んで行きます。

【ちょっと気をつけて!】電子メールの文字化けと同様の問題は、自分の使っているコンピュータが標準的に装備している漢字コードとは異なる日本語の漢字コードを用いているファイルを読む場合にも起ります。このような問題をさけるため、異なった漢字コードを変換する、「文字コード変換プログラム」で処理をする必要があります。

半角カタカナの問題点

文字化けの原因の一つに、電子メールをやり取りしているコンピュータ同士が、異なる日本語の漢字コードを用いている場合があります。また、通常電子メールは複数のコンピュータを中継しながら送られるため、たとえ送り手と受け手の漢字コードが同じでも、中継するコンピュータのどれかに漢字処理が不充分なものがあると、やはり文字化けが起こります。

インターネットの電子メールは、通常幾つもの異なった仕様のコンピュータを経由して送られます。このため送信された電子メールは、文字コード変換を繰り返しながら伝送されることがしばしば起ります。ところで半角カタカナは、パソコンやワークステーションといった異機種のパソコンがインターネットで繋がるという時代よりずっと古く決められた規格であり、またその利用もほぼ国内にかぎられていました。このため、半角カタカナの規格は国際的通用する規格として普及していませんし、これからの普及も考えられません。

さて、いま JIS8 ビットの半角カタカナコードの混じった電子メールがインターネットに発信されたとします。この電子メールが幾つかのコンピュータを通って伝送されている際、国際的には規格として通用していない 8 ビットカタカナがアルファベット ASCII の 7 ビットのコンピュータを通過したとします。するとこのコンピュータは 8 ビットに付いて来た先頭の１ビットを不要ものとして切り捨て処理をしてしまいます。このビットが落ちた形に変換されると、もはやもとの電子メールはまったく異なった意味の無い文字列に化けてしまうのです。

このため、電子メールで半角カタカナを使わないのがマナーとなっています。

【ちょっと気をつけて!】国内にあるコンピュータには外国製のものも多くあり、また国産のコンピュータであってもその基本ソフトの仕様も外国で決められたものに準拠しているため、コンピュータの世界では国内とか国外という発想ではなく、国際という視点でものごとを考えなければなりません。

コンピュータの入力では一つのキーボードを、あるときは漢字入力、あるときはアルファベット入力というように切り替えながら使っています。このため、電子メールを作っているときでもついうっかり半角カタカナを入力してしまうことが起ります。半角カタカナ自体は全角の半分のため、字の幅に注意をすればすぐに判別できますが、句読点を表す「､｡」

や、促音をあらわす「ー」、中グロを表す「・」の半角カタカナ文字は、画面上でもなかなか判別がつきませんので特に注意が必要です。

すべて全角文字で書いた場合: ＡＢＣ１２＠？
すべて半角文字で書いた場合: ABC12@?
全角・半角文字を混ぜた場合: ＡBC1２@？

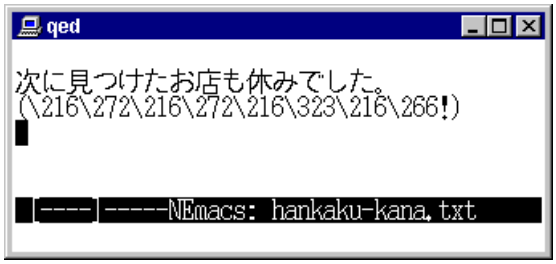


図 2.12: 半角カタカナの表示が異常な例

図 2.13: 半角カタカナの表示が正常な例

例に上がっているのは、「半角カタカナ」 を正常に表示不可能なソフトウェアと、「半角カタカナ」 を正常に表示可能なソフトウェアに無理矢理に 「半角カタカナ」 を含む文書を表示させた例です。 「ココモカ!」 の部分の文字がおかしくなっています。

外字の問題

半角カタカナ以外にも思った文字が遅れないものに外字と呼ばれるものがあります。

漢字の JIS コードで使っているのは JIS 第 1 水準 2965 文字とその他の第 2 水準 3388 字の合計 6353 字に過ぎません。このため、ワープロソフトを使っているときに、人名などの固有名詞では JIS に登録されていない漢字が必要になり困ることがあります。

このため、ワープロソフトには利用者が漢字を作って登録できる機能(外字作成機能)がついています。外字は、16 ビットで作られるビットパターンの内、JIS 第 1 水準と第 2 水準が割り当てられていない未使用のビットパターンに登録されます。このように、外字は利用者が登録した順に空き領域に振り分けられるため、作った字のコードはバラバラになります。同様、たとえメーカーが予め登録してあるものでもメーカー独自の振り分けのため、

メーカー毎に異なっていることが多いのです。外字は規格外の世界なのです。

最近はワープロから直接電子メールが送れるようになっていますが、電子メールで外字を送るとほとんどの場合、正しく伝えられません。半角カタカナ同様、電子メールで外字を使ってはいけません。

しかし、人名のようにどうしても外字でないと書けない文字を通信しなければならないときがあります。この場合は、その文字の形を別途の方法で伝えたり、その文字の形の画像ファイルを同時に送信したりする必要があるでしょう。

現在、これらの問題点を解決すべく、JIS 漢字コードの拡張が検討されています。この拡張によって、外字の問題のほとんどが解決することが期待されています。

【演習】インターネットで使ってはいけない文字についてまとめてみましょう。

【演習】JIS コード表に載っていない漢字を漢和辞典などからさがしてみましょう。それらの漢字をインターネットを使って通信するにはどのようにするべきでしょうか?

日本語以外の漢字

前節で取り上げた漢字の話題は、日本語の漢字の話題です。中国の漢字、台湾の漢字、韓国・北朝鮮の漢字のコードは日本語の漢字とは全く異なったコード体系を用いているので、受けとる人が漢字文化圏の国の人であっても、JIS が定めている日本語の漢字コードを解釈する OS を使っていない限り、日本語の日本語の漢字コードで書いた電子メールを送っても無意味です。

例えば、日本語の漢字コードを使って「我是田中」と書いても、日本でこれらの漢字に割り当てている番号と、日本以外の漢字文化圏の国でこれらの漢字に割り当てている番号が異なりますから、日本以外の漢字文化圏の国のコンピュータの画面には、この文字を表示することができません。

同じように、いわゆる「全角文字」の英字アルファベットを使っても、英語を母国語とする国では全く読むことができなくなります。

【演習】世界中の言語にはどんな言葉があるかを調べてみましょう。また、それらの言語で使われている文字を画面に表示したり、E-mail でやりとりしたりするための工夫を考えてみましょう。

2.4.4 テキストファイルとバイナリーファイル

ASCII や JIS コードの文字コードをテキストデータといい、テキストデータだけで出来ているファイルをテキストファイルといいます。文字コード表から分かるように、テキストデータは文字の種類と改行 (CR)などの少数の機能コードだけを規格として約束しているにすぎません。いわば文字だけで情報をやり取りする際の最低限度の約束にすぎません。

しかし私たちはワープロソフトを使って、文字の色や大きさといった文字属性を指定したり、下線や罫線などの文字飾りを行なって見栄えのよい文書を作っています。この基本的な考え方を見てみましょう。

--表現力のある文章を作る--

話を簡単にするために、次ぎのような具体的な場面を考えてみましょう。

いまワープロを作っているメーカーが顧客にアンケート調査をしたところ「黒以外の色や異なった大きさの文字が使えるワープロがほしい」ということが分かったとします。そこでメーカーは次ぎのように考えました。

メーカーの対応

A 社の場合(我が社は、まず文字色の増やすことに重点をおいて改良していこう)

「我が社では拡張 ASCII8 ビットの文字コードの先頭に 4 ビット追加した 12 ビットで 1 文字を扱うようにしよう。先頭の 4 ビットで出来るパターンは 16 種だから、16 色が使えるぞ。たとえば、0000 は黒、0101 は赤だ」

A 社のワープロでの文字の扱い　→　色指定に 4 ビット+ASCII8 ビット

例)赤色 A 黒色 B　0000　01000001　0101　01000010
　　黒　　A　　赤　　B

B 社の考え方(我が社では、色と大きさの両方を扱うことにしよう)

「我が社では A 社と同じ 12 ビットでも、色に 2 ビット使い 4 色を、黒 00、赤 01、青 10、黄 11。別の 2 ビットで、特大、大、標準、小の 4 サイズも用意しよう。標準サイズは 00、特大は 01 といった具合だ。」

B 社のワープロでの文字の扱い　サイズ 2 ビット+ASCII+色 2 ビット

【演習】A 社のワープロソフトで黒 A 赤 B という文字を、B 社のワープロで読むとどうなるでしょう。

(ヒント)
ビット列　0000　01000001　0101　01000010　を
00　00010000　01　01　01010000 10　と切り直し
B 社のルールで読みなおす

いま書いてきたことは極端に単純化してありますが、要点は、どのようなビット列に文字属性や文字飾りに対応させるのかは、応用ソフトメーカーやコンピュータメーカーの独自性にゆだねられているということです。そして、ビット列を切り分け、具体的な文章表現として実現させているのがメーカー毎のワープロソフトというプログラムなのです。

ワープロ文書の形式がメーカーごとで異なっており、そのメーカーのワープロソフトでなければ正しく読めないということは、ある意味でユーザーにとってはやっかいなことです。たとえば、電子メールでワープロ文章が添付ファイルとしてやり取りされる場合、受

け手の側に送り手と同じワープロが無いとうまく添付ファイルが読めないのも同じ理由です。

しかし、メーカーを縛る規格が少ないことは一面で技術競争を進めるという利点もあります。たとえば最近のワープロでは、他社のワープロで作った文書ファイルでもファイル変換プログラムを用意してできるだけ同じ文書イメージのまま読みこめるようになっているものも少なくありません。これは他社メーカーのワープロを使っているユーザーを何とか自社のワープロユーザーにしたいという競争原理が生み出した結果ですが、ユーザーにとってはワープロを変えても過去の文章を再利用できるという効能を生んでいます。

バインディング

今の例のように、テキストデータに文字色や文字サイズのような情報を付け加えることをバインディングといいます。ワープロソフト作った文章を特に指定すること無く保存するとバインディング情報を含めたファイルとして保存されます。これに対し「テキスト形式として保存せよ」といった指定をして保存すると、バインディング情報を省いたテキスト部分だけが保存できます。バインディングの考え方は、ワープロ文書に限らず表計算ソフトの罫線(けいせん)やセルの色など多くのソフトで見られます。なお、バインディング情報が増えると表現は豊かになりますが、当然ファイルの大きさも大きくなり、処理速度も遅くなるというトレードオフが起ります。

ワープロ文章の多くは、文字に色をつけるとか下線をつけるといったための装飾用のバインディング情報用領域をあらかじめ用意しているのが普通です。この領域は、ワープロで単に黒色の文字だけの文章を作っている場合でも確保され、しかも随分大きなものになります。実際、プリントアウトした紙の印刷物では同じに見える黒一色で文字飾りの無い単なる文字文章を、テキストだけを扱うエディタで作ったファイルとワープロで作ったファイルの大きさ（ビット数）を比べると、ワープロファイルはエディタファイルの 20 倍以上になっている場合も珍しくありません。文字だけの電子メールをワープロの添付ファイルで送ることの問題がここにもあります。

【演習】自分の通っている学校の校歌を例に、いま説明してきたことを実際に試して見ます。

（1）ワープロで校歌を入力し、ファイル名「校歌１」という名前のテキスト形式で保存する。

（2）次ぎに「校歌２」という名前に変えて、ワープロ文書形式で保存する。

（3）次ぎにいろんな色で文字飾りしたり、下線をつけたり、文字サイズを変えたりしたものを「校歌３」という名前に変えてワープロ文書形式で保存する。

以上、３つのファイルのサイズを確かめて、本文で説明したことを確認してみよう。

進化と再利用

バインディング情報はワープロソフトメーカー毎で異なっているので、多くの場合、異なったメーカーのワープロ文書ファイルは、別のメーカのワープロでは利用できません。

ワープロの文書ファイルに限らず、表計算などのファイルもメーカーの違いによって相互に利用できないものが多く、このような相互利用できないことを「互換性がない」といいます。こうしてみると、テキストで出来ているファイルは、もっとも互換性が高いファイルといえます。

ファイルの互換性の問題は、同一のメーカーのソフトでも起ります。ソフトは、ユーザーの声を聞きながら改良されていきます。改良は、新しい機能の追加のほか、古い機能が捨てられることもあります。このため、たとえ同じメーカーのワープロソフトであっても、改良に伴うモデルチェンジ(バージョンアップ)によって読み取れなかったり異なった働きをしたりする場合があります。

【ちょっと気をつけて!】互換性の問題は、プログラミング言語でも起ります。生徒の皆さんが中学校で学んだプログラミング言語にも幾つかの種類があり、また進化しています。

ソフトを利用しているときは、ソフトが進化するということを頭に置き、将来利用する可能性(再利用)のあるファイルは、必要に応じてファイル変換するなどの対応をしておくなどの必要があります。ソフトの改良は生物の進化に似ているので、コンピュータ科学でも「進化」と言われて、ソフトを考える上で重要な用件の一つとなっています。

【ちょっと気をつけて!】ファイルに互換性が無いと、再利用の面で不便なことが多くなります。このため普及度の高いソフトで作られるファイルでは、別のメーカのソフトでも利用できるようファイルに変換するソフトが開発されています。当然ながら、変換されるのは双方に共通なものという制約がつきます。なお、同一メーカーのソフトのバージョンアップでは、古いソフトの機能に新しい機能を追加するという形でなされることが多いため、「旧ソフトで作ったファイルは新ソフトでは扱えるが、新ソフトで作ったファイルは旧ソフトでは使えない」となるのが常です。このことを「上位互換」をとったバージョンアップといいます。

【演習】再利用には昔作った文書ファイルをそのまま利用するといったことの他にもあります。たとえば体育大会のプログラムのように、似た形式のものを何回も使うときには雛形を作っておき、変化した部分だけを差し替えるようにしておくと、作業効率が上がります。ワープロ文書や表計算、プログラム言語などで、再利用の考え方がどう使われているか調べてみましょう。

バイナリー(binary)データとバイナリーファイル

テキストデータにバインディングされたデータに限らず、テキストデータ以外の形式で出来ているデータをバイナリーデータといい、バイナリーデータで出来ているファイルをバイナリーファイルといいます。表計算ソフトや画像ソフトで作られるファイルはその例です。

例えばプログラムで COPY と書いた場合、8× 4 の 32 ビット必要ですが、プログラム言語のように命令数が 256 以下だと 8 ビットで全命令に対応するコードが作れるので、コンパ

クトになります。「全国国民体育大会」を「国体」といったり「parsonal computer」を「パソコン」あるいは「PC」といったりするのと同じです。

【ちょっと気をつけて!】以上の話は、書いたプログラムを形式的に小さくする話です。プログラムを実際コンピュータで実行できる形式に変換する場合は、たとえば copy という命令をコンピュータで実行できる機械語というものに変換するため、実行ファイルは大きくなります。ちょうど「レモンスカッシュを３人分下さい」という注文を、「レスカ３」というのが最初の話。注文を聞いた人が「レスカ３」を頭の中で「レモンスカッシュの元を３人分用意し、それを３つのカップにいれ、次ぎに炭酸水と氷をいれ、レモン片を乗せて仕上げる」と考えて実際作るのが「レスカ３」という命令を実行形式の作業手順に置きかえるのが実行プログラムに展開することにあたります。

【演習】画像ファイルを電子メールに添付して他の人に送ってみましょう。また、そのとき送られた電子メールの大きさを調べてみましょう。さらに、送ったメールを受け取ってもらったら、どんな画像ファイルだったかを E-mail で返信してもらいましょう。

2.4.5 数値の表現

コンピュータは、計算機と翻訳されるように当初は計算を目的に発明されました。では、コンピュータの内部で数値はどのように表現されているのでしょうか。

私たちは算用数字を並べて数値を表わし、それを操って計算しています。このため、365 という数値は、コンピュータ内部で「3」という数字のコードと「6」「5」というコードをつなげたビットパターンで数値を表現していると考えがちですが、ほとんどのコンピュータではこのような形で数値を扱っていません。前に述べたように数字はあくまで文字の一種としての記号であり、数値を表現しているのではないからです。私たちが普段使っている十進法の数値は、それに対応させた二進法として符号化されているのです。

数値の符号化では、文字に見られない次のような問題があります。

1. 文字と違って数値は理論上、+無限大から-無限大まで存在するので、単純にビットパターンを割り当てると、無数のパターンが必要となる。この問題は、表現できる数値の範囲を限定し(数値に使うバイト数の制限、固定長)、この範囲を超えた数値の取り扱いはプログラム(program)を工夫して取り扱うという方法で回避されている。これは 8 桁の表示部分を持たない電卓をつかって 20 桁の足し算をするときの工夫と似ている。
2. 私たちが扱う数には、0、正の整数のほか、負の整数もある。これらの表現はコンピュータの計算回路とも関係するので、文字のような単純なビットパターンとの対応ではいけない。

正の整数に限った表現

もっとも単純な考え方は、二進法の表現を位取りの原理にしたがって計算した結果を十進法の数値と対応させるというものです。つまり、$(101)_{二}$のビットパターンを $1\times 2^2+0\times 2^1+1\times 2^0$ で示される整数値の 5 と見るのです。この考え方を 2 バイト(16 ビット)の

固定長に適応すると、2 バイトで表されるビットパターン 65536 だから、0 から 65535 までの正の整数が表される。このように 2 バイト全部を 0 と正の整数に対応させる方法を「符号なし整数」といいます。

例として 3 ビットを用いた 2 進数を示します。

数	ビット列
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

数値は文字と異なって計算という問題があります。いま 3 ビットを数値に割り当てたとします。そこで、1+2 という計算を行うと、計算結果の 3 は 011 ビットパターンで表現できますが、4+5 の場合、答の 9 に対応させるべきビットパターンはありません。この問題は 3 ビットでの表現にかかわらず、いかなるビット数を数値に与えようと起る問題です。そこで、計算結果が用意したビットパターンにおさまらないときは、コンピュータは「桁溢れ(オーバーフロー)」というエラー処理を行います。エラーが起ったかどうかの判定は、計算部分に桁溢れを監視する仕組みを用意しておき判定しています。

【ちょっと気をつけて!】数学では、$a^2+b^2=(a+b)(a-b)$、が成り立ちます。しかし、コンピュータではこの等式は必ずしも成り立つといえません。たとえば、3 ビット表現の場合で a=3、b=2 の場合の値を計算してみます。すると、左辺は $3^2=9$ のため桁溢れのエラーとなります。一方、左辺は$(3+2)(3-2)=5^{x}\ 1=1$ だから桁溢れなく計算できるのです。このように、コンピュータで正しく計算させるには工夫が必要なことがしばしば起ります。コンピュータで正しく計算させることを研究する分野は「数値計算」と呼ばれています。

正負の両整数の表現

16 ビットの 65536 のパターンに、0 と正負の整数を割り当てる方法を考えてみましょう。素直に考えれば、符号なし整数と同様の方法で表現で全パターンの半分の、0000000000000000=$(0000)_{十六}$から 0111111111111111=$(4FFF)_{十六}$を 0 から正の整数 32767 に割り当て、残りを負の整数に割り当てるという方法が浮かびます。

問題は、残った

1000000000000000=$(8000)_{十六}$

から

1111111111111111=$(FFFF)_{十六}$

までのビットパターンにどのような規則で負の整数に割り付けるかです。当然計算上でう

まく扱えるように対応させる必要があります。このようにして考え出されたのが、「補数」による表現なのです。

補数による負の数の表現

負の数の表現を考えるヒントは、絶対値が同じ正の整数と負の整数を加えると 0 になるといこうことです。なお、二進法の足し算の基本ルールは次の単純な 4 つに過ぎません。

足し算

0 + 0 = 0

1 + 0 = 1

0 + 1 = 1

1 + 1 = 10

この計算ルールをもとに、各正の整数と、負の整数候補に残したビットパターン

1000000000000000(8000)$_{十六}$から 111111111111111(FFFF)$_{十六}$

と正の値とのの足し算を行ってみると、面白いルールが発見できます。桁数が多いと見ずらいので 3 ビット 8 種類のパターンで行ってみます。まず、3 ビットで負以外の整数は 0、1、2、3 の 4 種類で、それぞれ(000、001、010、011)に対応ずけしました。問題は残りの負の整数-1、-2、-3、-4 を候補パターン(111、110、100)にどのように対応させるかとなります。

そこで、絶対値が同じ正の数と負の数を加えるとゼロ(0)になるということを考えながら、幾つかの組み合わせの計算をしてみると、次のような興味深い組み合わせが見つかります。

十進法	二進法表現
1	001 001 + 111 = 1000
2	010 010 + 110 = 1000
3	011 011 + 101 = 1000

正の数と候補パターンの和の例

いま「最上位ビットを無視して下 3 桁を見る」というルールで和の結果を見てみると、とすべて 0 の表現と一致していることがわかります。どうやら 111、110、101 を 4 ビットでの-1、-2、-3、残った 100 を-4 としてよさそうです。

念のため、今のルールで十進法表現の「3-1=2」や「1-4=-3」の計算でもうまくいくか確かめてみました。

011 + 111= 1010 下 3 桁の 010 は、 十進法表現の 2

001 + 100= 0101 下 3 桁の 101 は、 いま決めたルールで十進法の 3

011 + 111= 1010 下 3 桁の 010 は、 十進法表現の 2

001 + 100= 0101 下 3 桁の 101 は、 いま決めたルールで十進法の 3

このようにうまくは、次のような二進数の計算と対応しているからです。

```
1- 1 =  1+ (1000- 1)-1000 = 1+111-1000
10-10 = 10 + (1000-10)-1000 = 1+110-1000
```

そこで 3 ビットの正負の表現として次の対応をさせてみます。

数	ビット列
0	000
1	001
2	010
3	011
-4	100
-3	101
-2	110
-1	111

同じビット数で負の数まで表すため、正の数については表せる範囲が半分になっています。
この表は次のルールで正負を判定します。
最上位(一番左)のビットが 1 なら負、0 なら正
しかしこの対応には大きな問題があります。

十進法	二進法	比較
(-4) + (-1) = -5	100 + 111 = 1011	1011 =(3)十 ?a
(+2) + (+3) = +5	010 + 011 = 101	101 =(-3)十 ?b

?a のケースは、桁溢れ(オーバーフロー)のルールで対処できますが、?b のケースはそう単純ではありません。

【演習】実際のコンピュータでは、表で示した対応で正負の数値を扱っています。この表で間違いなく計算できるように、桁溢れ(オーバーフロー)に対応するルールを見つけてみましょう。(ヒント、いろいろなケースの組み合わせを行い、規則を発見する)

一般に、ある正の数の二進法表現のビットパターン A が与えられたとき、すべての 1 と 0 を入れ替えて出来るビットパターンを、「A に対する 1 の補数」といいます。「1 の補数」に 1 を加えて出来るビットパターンを「A に対する 2 の補数」といい、コンピュータでは、絶対値 A に対する負の数(-A)として扱っています。なお、「2 の補数」を使ったこのようにして 0 および正と負の整数を扱うやり方では、最高位のビットが 0 のときは正、-1 のときは負というようになっています。このような方法で正と負の数の表現を補数符号付き整数といいます。

【ちょっと気をつけて!】2 の補数を使うと、いま見てきたように引き算が足し算でできます。このことは、足し算回路 1 つで加減という 2 つの計算ができることになり、コンピュータの演算回路を単純にしできます。しかし、2 の補数を作る回路が複雑なら意味がありません。しかし、機械的にビットパターンを反転させるという回路は足し算の回路よりなお簡単ですから問題はありません。

【演習】十進法で「12345-5678」「5678-12345」を、いま行った「2 の補数」を利用した 16 ビットの 2 進法表現で行ってみよう。

【演習】16 ビット表現の符号付き整数で扱える整数は-32768 から 32767 に過ぎません。このため、より大きな整数を扱う 32 ビット表現(4 バイト)も利用されている。32 ビット表現」で扱える符号付き整数の範囲を確かめてみよう。

実数の表現

小数点以下の値も含んだ数値を実数といいます。実数を表す場合には、先に出て来たように「小さい値を単位として」表すこともできますが、それだと(絶対値が)非常に大きい数や小さい数を表すのが不便なので、指数方式または浮動少数点と呼ばれる表現を使うのが普通です。この方法では次の情報を組み合わせて 1 つの数を表現します。

- 全体の符号(1 ビット)。数が正か負かを表す。
- 仮数の値。0〜1 の値を適当なきざみ幅で表す。
- 指数の符号(1 ビット)。指数が正か負かを表す。
- 指数の値。

2 進数のままだと見にくいので、2^{10}=1024 ですから、たとえば指数も仮数も 10 ビットずつ(全体では符号を含めて 22 ビット)で仮数は 0:000〜1:000(0:001 きざみ)、指数も 0〜1000 の値だとして 10 進数の形で表して考えてみましょう(図 2.14)。

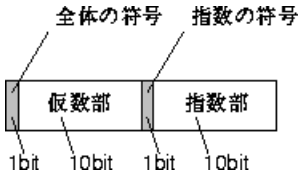


図 2.14: 実数の浮動少数点表現

この表現では次のような数が表せます。

○ 1:000 101000 表せる数で絶対値が最大のもの

○ 0:001 101000 表せる数で絶対値が最小のもの (ただし 0 は除く)

非常に広い範囲の数が表せそうですが、しかし次のものは表せません。

× 3:142=0:3142 101 仮数部は 3 桁までしか表せない

× 1234 = 0:1234 104 仮数部は 3 桁までしか表せない

つまり、指数方式の表現では有効数字の桁数が決まっているため、絶対値が大きくなると表せる値のきざみも大きくなってしまいます。

【ちょっと気をつけて!】実際のコンピュータでは、実数の表現に 64 ビット以上使うことが普通ですから、上の例よりはもっと細かい数値が表せます。しかし原理は上と同じですから、限界があることに変わりはありません。

数値表現と誤差

実数を取り扱う場合には、常に誤差の問題が付きまといます。このことは普段の十進法の計算でも見られます。たとえば、私たちは普段πを 3.14 として計算することが多いですが、πは無理数ですから正しくは 3.14159....と無限の桁まで存在します。もっと単純な例で言えば、3 分の 2 という分数も小数で表わせば 0.66666....と無限に続きます。この誤差は人間が有限の桁でしか小数を扱うほかないからです。このことは、コンピュータの場合も同じです。

コンピュータで計算をさせると、これとは別の種類の誤差が存在します。それは十進法では有限確定値となる出ない小数が、二進法で数値を扱うコンピュータでは無限に続く小数になり、それを有限で打ち切るための発生する誤差です。

例．　$(0.2)_{十}=(0.011011011.....$無限に繰り返す$)_{二}$

十進法の 0.2 とは、単位 1 の長さを十分の一$(0.1)_{十}$の小単位 2 個分ということです。では同じ長さを二進法で表わせばどうなるでしょう。二進法の$(0.1)_{二}$に対応する小単位は、単位 1 の二分の一(十進法の 0.5)ですからこれでは大きすぎます。そこでよりこの小単位のさらに二分の一(十進法の 0.25)、さらに二分の一の単位(十進法の 0.125)というようにどんどんちいさな単位を作っておき、それらの組み合わせで十進法の 0.2 を作ろうとすると$(0.00110011011.....)_{二}$となるわけです。この関係は三分の一を十進法の小数で表わすと$(0.3333....)_{十}$と無限小数になるが、三進法なら$(0.1)_{三}$となるのと同じです。

このような計算に伴う小数の誤差の問題をいかに小さくするかも、数値計算という分野の研究対象となっています。

2.4.6 文字や数値の混ざったビットパターンの読み取り

私たちがコンピュータを使う本来の目的は、情報を処理し、何らかの目的を果たしたいためです。このため、文字や数値の表現を決めただけではまだ不十分なのです。ちょうど、個々の単語を知っていても文法を知らなければ文章を作れないように、ビットパターンで表現された文字や数値の集まりは、処理手順を書いたコンピュータプログラムにしたがって、正しく読み取られるようになっていなければいけません。文字と整数型や実数型の数値が混在しているファイル(ビットパターンの集合体)から、正しく文字と数値を読み取るやり方を、例を通して見てみましょう。

たとえば、2 バイト(16 ビット)で表されるビットパターンで数値を表現していると、必ず JIS 漢字コードの表現と重なるものが出てくるという問題があります。文字では使わないビ

ットパターンということがありますが、数値を一定の長さのビットで表現する場合、数値の性格からいってすべてのパターンが起りうるわけです。

このような重なり問題などを解決し、誤りなく「この場合は数値のビット列」「この場合は文字」とプログラムが解釈できるように、要素となる文字コードや数値コードの集まりは、ある工夫によってメモリーに配置されています。その基本となる考え方はすでに説明してきた固定長と分離記号の発想をもとに、いわば「読み取りのための文法」を作るっておくのです。説明のため、仮に次のような読み出しルールを決めてみましょう。

例) 説明のための仮想的なものである
このデータファイルは次のルールによって読み出してください。
(1) ファイルの先頭の 1 ビットは文字か数値を指定している。0 なら文字、1 なら数値です。
(1-1) 0 のとき(数値とわかったとき)
(a) 先頭に続く 2 ビットは、数値の種類を示します。00 なら、符号なし整数、01 なら符号付き整数型、10 なら符号付き小数、11 なら符号なし小数です。
(b) 続く 2 ビットは、数値サイズの固定長を示します。0 なら 2 バイト長、1 なら 4 バイト長です。
(c) 続く 2 バイトは、何個の数値があるかを符号なし整数型で示してあります。例(b)で 2 バイト型で、ここの値が 150 なら、300 バイト先で数値データは終わります。
(d) 続く 1 バイトが 0 のとき、ファイルは終わりです。1 なら(1)に戻って同じ事を繰り返してください。

(1-2) 1 のとき(文字とわかったとき)
(a)先頭に続く 2 ビットは、文字コードの種類を示します。00 なら ASCII、01 なら拡張 ASCII、01 なら JIS 漢字、11 ならシフト JIS です。
(b)続く 2 バイトは、何個の文字があるかを符号なし整数型で示してあります。例(b)で JIS 漢字で、ここの値が 150 なら、600 バイト先で文字データは終わります。
(c)続く 1 バイトが 0 のとき、ファイルは終わりです。1 なら(1)に戻って同じ事を繰り返してください。
この例のように、0 と 1 というビットが並んでいるファイルから、適切に文字や数値を誤りなく切り取って読み出すことができるのです。

2.5 デジタル信号とアナログ信号

私たちの文明や文化は、文字や数値という「記号」や、絵画、音楽、物の重さ、電流の値といった「モノの状態」、食料、衣服といった「モノ」自体によって出来ています。このうち、記号で表現できる情報は、ビットパターンによって表現できることが分かりました。それは、そもそも人間の扱う記号が「数えることのできる」という性質を持っていたからです。

さて、人間の作った記号に限らず、数えることのできるものを一般的に離散量といいます。離散量は、電磁気的な 2 種類のビットから作られる組み合わせパターンに対応させることにより、コンピュータで処理が可能となります。

これに対し、絵画の色や音楽の音、電流の強さ、物の重さといったものは、たとえば 1

グラムと 1.1 グラムの間には、1.05 グラム、1.032014...というように無限の値が可能です。このように、連続的な値を取りうる量をアナログ(analog)量といいます。

離散量や連続量を情報として取り扱うとき、それぞれデジタル情報、アナログ情報、あるいは通信ではデジタル信号、アナログ信号といいます。

今日広く普及しているコンピュータは、正確にはデジタルコンピュータというように、デジタル情報を処理する機械です。このため、アナログ情報は、旧来のテープレコーダーや電話の音声というように、たとえそれが電磁気的に表現されていても、その扱いがアナログ情報である限りコンピュータで処理することはできません。アナログ情報をコンピュータで処理できるには、何らかの方法でデジタル情報に変換する必要があり、その分、手間がかかります。それでもアナログ情報をデジタル情報に変換して処理する背景には、デジタル化にはその手間を上回る利益があるからなのです。

2.5.1 アナログ情報の伝送と記録

私たちが会話する過程をすこし細かく振り返ってみよう。いまある人が「おはよう」という言葉を他の人に伝えるということを細かく見てみましょう。

まず声帯が「おはよう」という言葉に応じた振動をします。声帯の振動は、その振動状態を写し取る形で周りの空気を振動させながら広がってきます。空気の振動が鼓膜に届くと、鼓膜は空気の振動を写し取る形で振動するのです。このようにして、声帯の発した「おはよう」という言葉が別の人に伝えられるのであす。

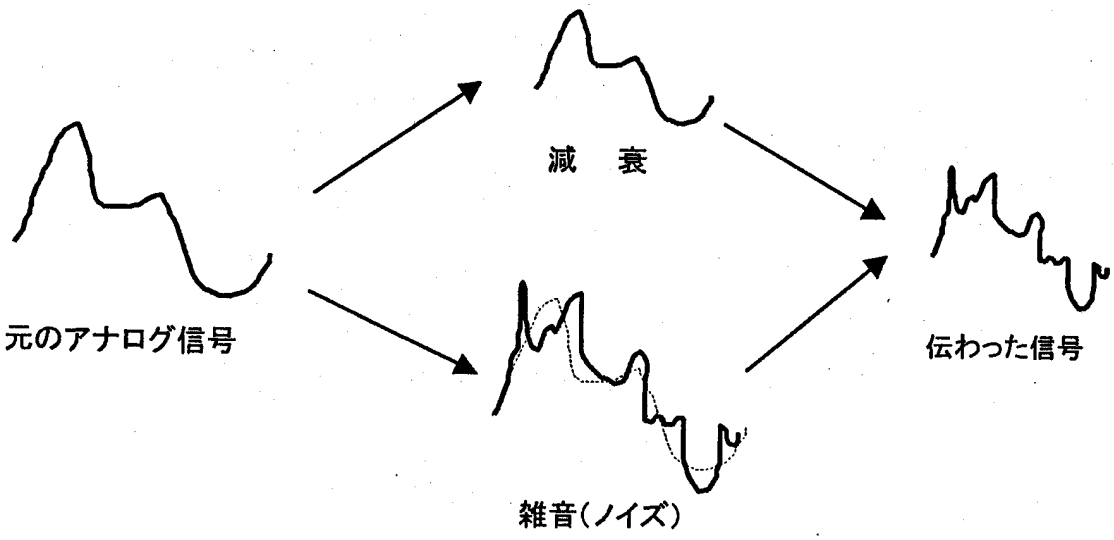


図 2.15: 減衰とノイズ

この例のように、情報源(音源)のアナログ情報(振動)を、「そのままの形」で別のもの(媒体、メディア)の状態(振動)に写し取りながら送る方法をアナログ伝送といいます。

音声伝達における伝達媒体の変更

通常の会話は、空気を媒体にして行われるますが、空気の変わりに糸を利用したのが糸電話、糸の振動の変わりに電線を流れる電流振動にしたのが普通の電話、電波にしたのが

無線電話です。

アナログ伝送の問題減衰、雑音

こうして比較してみると、以上の音の伝送は、途中の伝送媒体が変わっても、「発信側-アナログ状態を媒介するもの-受信側」という仕組みは変わっていないことが分かります。アナログ伝送には、次のような問題点が常につきまといます。

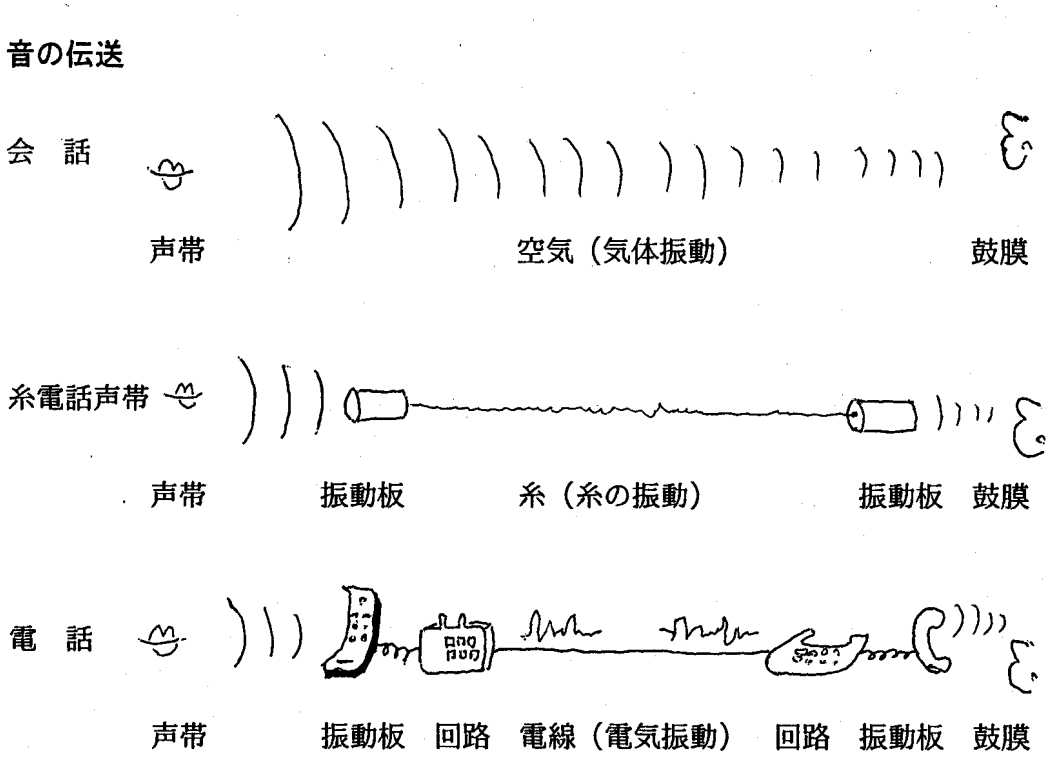


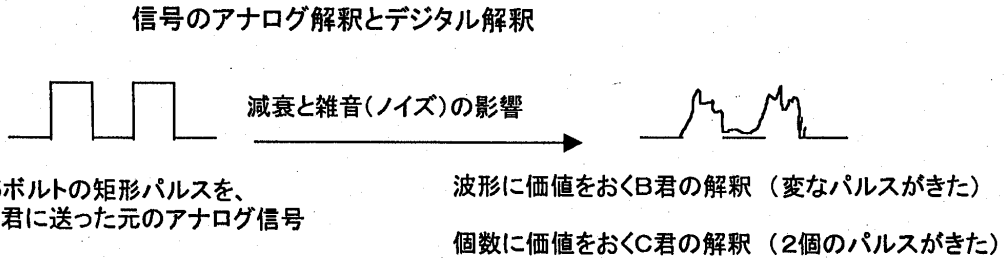
図 2.16: 音の伝送

【用語】減衰:
音を遠くまで伝えると音が小さくなる。このように、元の情報が伝送にしたがって小さくなることを、減衰といいます。

【用語】雑音:
音を伝えるとき、伝えたい音以外の音を雑音といいますが、これを一般化して、取り扱いたい情報にくっ付き歪めるものすべてを雑音(ノイズ、noisu)といいます。雷が鳴るとテレビの画面が乱れることがありますが、これは画像情報を伝える電波に雷の電波がノイズとして加わったためなのです。ノイズは積み重なる性質があるので、一つ一つのノイズは小さくとも伝送が長くなるにしたがってノイズの影響は大きくなります。

【ちょっと気をつけて!】減衰対策としてには、減衰を起こしにくい媒体の開発や途中で信号を大きくする(増幅)ことが行われていますが、増幅には本来の情報と同時にノイズも増幅すると厄介な問題が入ってきます。雑音対策に、雑音を拾いにくい材質を開発するとか、

本来の情報は通してもノイズだけを通さないという性質をもったフィルタ(ふるい)という装置を挟むことにより除去が行われています。しかし、フィルタで効果的に取り除けるの



は、元の信号と性質が大きく異なったものに過ぎず、完全に取り除くことはできません。

図 2.17: 信号のアナログ解釈とデジタル解釈

アナログ記録の問題記録媒体の劣化

さて、先の図 2.16 の鼓膜にあたる部分(振動膜)に針をつけ、その針を回転する板にあてます。すると、針の振動に応じて板に溝が刻まれていきます。逆に、この板に出来た溝に針を置き、板を同じ回転で回すと、針は溝の形と同じ振動をします。そこでこの振動を振動膜に伝えると音が再現できるわけです。これが音を初めて記録したレコードという装置の基本原理なのです。

この例のように、情報源の状態を、「そのままの形」で別のもの(記録媒体)にコピーしながら記録する、というのがアナログ記録の本質なのです。アナログ記録には次のような問題点があります。

【用語】劣化:
記録媒体は物質であるため、時間とともに変質する。このため、記録されたアナログ情報の質は時間とともに劣化する。この劣化は、アナログ記録である限り避けられない。

アナログ伝送にせよアナログ記録にせよ、アナログ情報をもとの状態そのままで取り扱う限り、減衰、雑音、劣化という問題点から逃れることは出来ません。アナログ情報を取り扱う技術はこれら問題点の克服にあったといっても過言ではありません。

【演習】アナログタイプの複写機で新聞(オリジナル、親情報)をコピーする。次に出来上がったコピー(子)から、さらにコピー(孫)する。この操作を、曾孫...と何度も繰り返し、画質の落ちかたを確認してみよう。同様のことを、カセット、ビデオでも確かめてみよう。

2.5.2 アナログ情報のデジタル化発想の転換

芭蕉が作った俳句の短冊には、2 つの異なった価値があることを説明しました。一つは、俳句の書かれた短冊そのものも骨董的な「モノとしての価値」で、いま一つは、俳句が表現している「意味としての価値」である。俳句の「意味としての価値」は、書かれた文字

に虫食い(雑音)があっても、短冊がある場所から離れていって見える文字が小さくなって(減衰)、墨跡が薄くなって(劣化)いても、文字として読めさえすれば、価値に変わりないのです。それどころか、目の不自由な人のために、点字で表されても同じです。アナログ情報のデジタル化の本質は、このことに深く関係しています。

コンピュータ内部やデジタル通信では、ビットパターンを電気パルスとして取り扱われています。いま一つの電気パルスを電流の波形(アナログ情報)に注目すると、波形は電線を流れるに従い、雑音の影響を受けたり減衰します。しかし、波形ではなくパルスが「来た、来ない」ということに注目すると、一つのパルスが一つとして認識できさえすればデジタル情報としての価値は減りません。次のことはこのことを象徴的に示している。

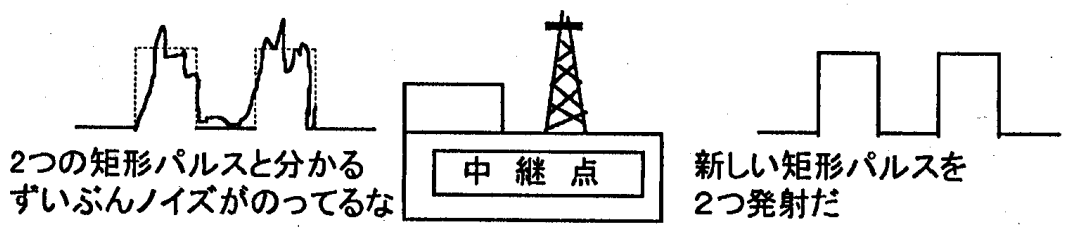


図 2.18: パルスの中継

C 君はあらかじめ、3 ボルト以上に飛び出した波形があれば、それを「5 ボルトの矩形パルス」が送られた、と決めていたのである。

このように、やり取りするパルスを、形(モノ)ではなく個数(意味)に着目すれば、雑音や減衰の影響は大幅に改善されます。さらに、もし雑音や減衰の影響が無視できないほど遠くに送る必要があるときは、途中にパルス波形を整形する中継点をおけばよいのです。

アナログ情報とデジタル情報の相互変換

アナログ情報をうまくデジタル情報として表現できれば、アナログ情報の持つ諸問題、つまり伝送や記録、そして複写時におこる減衰、雑音、劣化という問題が解決できることを意味します。たとえば従来のカメラで撮影したカラー写真は、時間と共に色が変わってきますが、デジタルカメラでとった写真は色の劣化は起こりません。もっとも、記録媒体が時間的に品質低下を起こすとそもそも読めなくなるかもしれないという恐れがありますが、この場合は定期的に新しい記録媒体にバックアップコピーすれば解決できます。なぜなら、デジタル信号は、親、子、孫と何段階複写しても品質劣化が起こらないからです。

【ちょっと気をつけて!】他人の作った音楽テープを無断でコピーすることは著作権違反という犯罪にあたります。アナログ全盛の時代では、ダビングを繰り返すと音質が悪くなるため、不法コピーの問題は深刻でありませんでした。しかし、雑音の影響を受けず、また記録媒体の劣化がパルスに影響の来る前に再度コピーするということを繰り返すと、いつでも、また幾つでもオリジナルとまったく同じものが複製できるというデジタル情報記録技術の出現により、不法コピーの問題は一気に深刻になりました。このため、デジタルテープレコーダーでは、オリジナル作品をコピーするとオリジナルでないと分かる仕掛けを

作り、繰り返しの複製ができなくするようになっています。不法コピーの問題は、情報社会で深刻になった問題の一例です。

アナログ情報をデジタル情報に変換することをアナログ/デジタル変換(A/D 変換 analog/digital conversion)といいます。

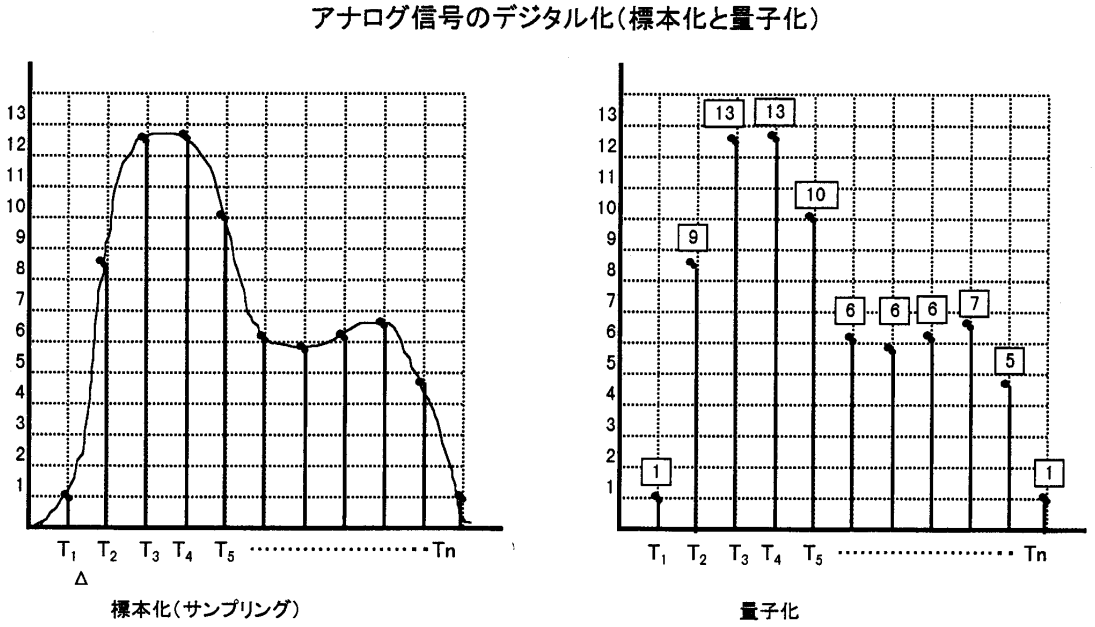


図 2.19: アナログ信号のデジタル化

ところで、たとえば音声を A/D 変換して伝送すると、もとの音声波形という姿ではなく電気パルスとしてのビットパターンとして伝送されるため、受けとった側がそのまま聞いたところで音声として聞き取れません。このため、デジタル変換を元のアナログ情報に戻す必要がありまする。この変換をデジタル/アナログ変換(D/A 変換 digital/analog conversion)といいます。

アナログ/デジタル変換の原理

A/D 変換は、標本化(sampling)と量子化(quantization)という 2 段階によってなされます。音の波形を例に、このことを見てみましょう。

いま音のアナログ情報の波形として図 2.19 のようなものがあったとします。この波形の時間軸に沿って等間隔 W_t の点列、$T_0;T_1;T_2;...;T_n$ をとり、その波高値を読み取ります。この作業が標本化です。

さて、波高値は連続量(アナログ)ですから一般的に小数以下の値が存在しますが、その値にもっとも近い整数値で近似してそれを波高値と見なします。この整数化にする作業が、

量子化です。量子化の単位としては、取り扱う波形を取りうる最大電圧を 2 の N 乗分の 1 とします。たとえば、最大波高値が 5 ボルト、N を 8 とすると、5256 ボルトが単位 1 となる。この場合、量子化された値は 0 から 255 までのいずれかの値となるので、8 ビット(1 バイト)の固定長のビットパターンで表わせます。当然、N を大きくするほど誤差の少ない細かな量子化になります。

最後に、量子化された各波高値を二進法で表現する電気パルス列によって A/D 変換が終了し、元のアナログ波形が、対応する電気パルスの集まりとしてデジタル信号で表現されます。

D/A 変換は、いまの逆で、送られて来きたパルスから量子化された各値を読み取り、それぞれを高さとする幅 W_t の階段型の波形とし再生するのです。

アナログ/デジタル変換に伴う誤差の問題

A/D 変換の結果得られたデジタル情報は、そこから先の伝送等の扱いではアナログの持つ雑音などの諸問題は非常に起りにくくなります。しかし、A/D 変換の原理から分かるように、そもそもデジタル化では標本化と量子化の段階で誤差が入ってきます。しかし、これらの誤差は標本採集のための時間間隔 Wt をどんどん小さくし、量子化のための刻みの個数(N)を大きくすることで数学的には幾らでも小さくすることができます。極端な話、標本化の間隔を無限に狭くし量子化の刻み個数を無限に大きくすれば完全にもとのアナログ情報を再現できます。しかし、誤差を減らすほど取り扱う情報量が増えるというトレードオフが起こります。

標本化と量子化の基準を決めるもの

このトレードオフ問題は、例えば音の場合、人間が聞き取れない音は扱っても意味がないということを利用して解決が図られています。具体的には、音の性質は振動周期(振動数)と振幅で決っていますが、「アナログ波形の最大振動周期(もっとも激しく震動する部分)の半分の間隔で標本化すれば、元の波形が復元できる」ということ(標本化定理)から抽出間隔 W_t の基準値が決まり、音の強弱の違いを聞き分けられる能力(聴力分解能力)から量子化レベル N が求まるのです。

【演習】電話を通して聞こえる CD の音楽は、直接 CD で聞く音と随分違って聞こえます。その原因の一つに A/D 変換が関係しています。その理由を考えてみましょう。(ヒント電話は人間の声を基準に作られている。
一秒間に 1 回の振動を 1 ヘルツ(Hz)といいます。人間の聴力は 1 秒間に 20Hz から 20000Hz 程度の音域を聞き分けられるのに対し、人間の声は 100Hz から 3000Hz 程度の音域しか出せません。音楽を作る楽器の音域は人間の声より広く、また大きな音が出せます)

【ちょっと気をつけて!】音のように時間的に変化するアナログ量をデジタル変換して取り扱うには、雑音といった誤差問題の他に、処理時間の問題があります。例えば千分の一秒間隔で標本化し、N=256 で量子化したとします。いま量子化した値に対応する電気パルスの 1 ビット分の幅が千分の一秒とし、それを 1 本の電線で 8 ビットのパルスを順に送って

いたのでは、波形は最低でも千分の八秒に引き伸ばされたことになり、実用化できません。

これを解決するには、8 本の電線で一度に送るとか、パルスの幅をより短くする(1 秒間に扱える数を増やす)といった工夫が必要なのです。コンピュータのカタログには 32 ビット CPU400MHz という言葉が書かれていますが、これらは 1 秒間に何個のパルスを扱えるかや同時に何ビット処理できるかを示したものです。この値からコンピュータの性能のおおよその目安がつけられます。

音や動画に比べ、文字や静止画の取り扱いでは時間の問題は本質的ではありませんが、キー入力スピードより画面表示が遅いワープロソフトは使う気にならないでしょう。高度情報社会の進展でどんどん増加する情報量を扱うには、より高速に処理できるコンピュータとより高速化な通信網が必要となるのです。

2.6 画像情報のデジタル情報化と活用

映画などを見ていると、写されている物の形が急に変形したり、材質感が変わったり、漫画のキャラクターと人間とが違和感なく同一画面で合成されていたりするシーンをよく見ます。これらは、デジタル表現された画像情報をコンピュータを通して処理することにより作られています。一般に、コンピュータを用いて画像を作ることをコンピュータグラフィックス(CG computer graphics)といいます。いまの例は、コンピュータグラフィックス技術の応用の一例です。ここではコンピュータグラフィックスの基礎となる画像情報のデジタル変換の原理や、画像加工の考え方を学びます。

2.6.1 画像情報のデジタル情報化

テレビや液晶ディスプレー画面を拡大してみると、赤(R、red)緑(G、green)青(B、blue)という光の 3 原色の点で出来ていることが分かります。人間が大きさを見分ける能力(視覚分解能)には限りがあるため、狭い範囲に RGB で発色している小さな３つの点を配置しておくと､私たちはそれらを一つの色点があると思ってしまうのです｡さらに見える色も RGB が混ざった別の色として見てしまいます。

たとえば RGB それぞれの発色の強さ(輝度)が同じ強さで強く発色していれば白、弱くなるとグレー、RG だけが同じ強さで強く発色すれば黄色、弱ければ薄い黄色というわけです。そして RGB がそれぞれ適当な程度の輝度で発色している場合、適当な中間色として見えます。

RGB の各点はドット(dot)、3 ドットのまとまりを画素(ピクセル、pixel)といいますが、コンピュータで画像を扱う場合、画素（ピクセル）が最小単位とすることが普通です。

ピクセルで画像を描く原理

ピクセルで画像を描く基本原理は、マスゲームで人文字を描くのと同じやりかたです。図 2.20 はこのことを模式的に示したもので、A 君が電話を使って離れたところにいる B 君にある模様を描かせる様子を示しています。色は黒 1 色とします。

A 君と B 君は、同じ目盛りの方眼紙を用意します。A 君の目盛りは透明なプラスチック板、B 君は普通の紙とします。

A 君と B 君の約束

塗りつぶしのルール

1. A 君が電話口で、0 か 1 という言葉を順に B 君に伝える。
2. B 君は、0 が来たら白で、1 が来たら黒で桝目を塗りつぶす

塗る桝目指定のルール(A/D 変換)

1. A 君が 0 か 1 を伝える順番は 1 行 1 列目から 2 列目...と最後の列まで行き、次に 2 行目に移る。2 行目でも同じようにし、以下最終行まで繰り返す。
2. B 君が桝目を塗りつぶす順を、A 君と同じにしておく。

A 君の色決めのルール

1. 桝目の中心を選び(標本化)、そこが白なら 0、黒なら 1 とする(量子化)。

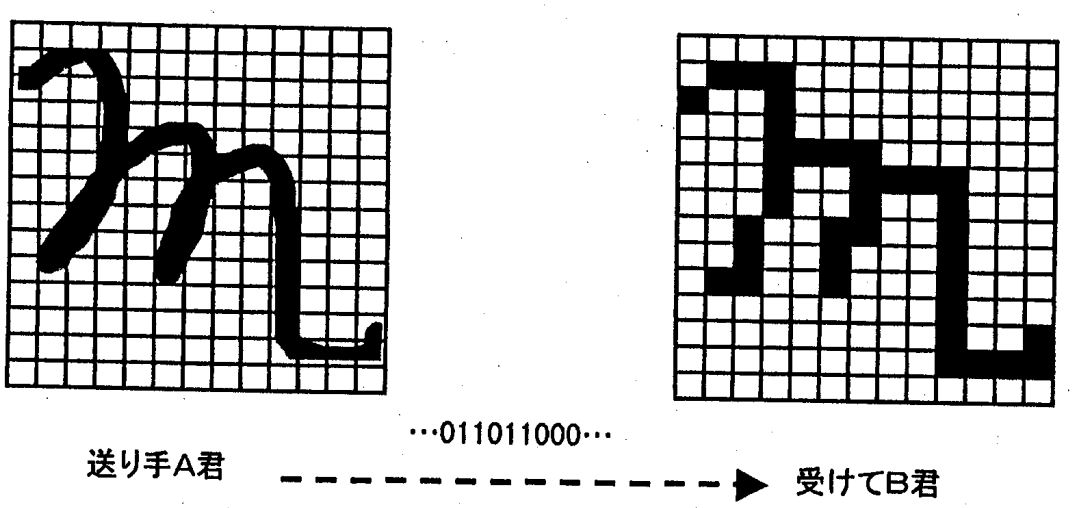


図 2.20: アナログ画像のデジタル化

このルールで画像を送る場合、黒 1 色のため色の誤差(量子化誤差)はありませんが、用意する方眼が荒いと B 君が作る図は元の画像とかなり異なったものになります(標本誤差)。しかし、方眼を半分、その半分と細かくしていくとどんどん元の絵に近づきやがて、人間の目には同じように見えるようになります。当然ながら標本化を細かくするほど扱う 0,1 の量は急激に増えてしまいます(今の場合 2 乗)。画質と扱うべき情報量の間には、このようにトレードオフがあります。

カラー画像の場合

扱いたい画像がカラー写真のような場合も、「色決めのルール」の部分が変わるだけです。ただ、色は連続的に変化するため、方眼の中心の色を幾つかの色に量子化して使うため量子化誤差が入ります。この場合も人間の色差識別能力が量子化の基準となります。

では、基本色 RGB の輝度用に N ビット用意したとき、どれぐらいの色が作られるのでしょう。説明のため、N=2 とします。

N=2 のときの色の種類(1 ピクセルあたり、2 ビット×3 の 6 ビット仕様)(1)2 ビットで出来るビットパターンは、00 01 10 11 の 4 種類(22)(2)各ビットパターンを、輝度 0(なし)、輝度 1、輝度 2、輝度 4(最高)と対応(3)ピクセルの色を C(R 輝度、G 輝度、B 輝度)で表すと、

R 輝度の種類×G 輝度の種類×R 輝度の種類(=4x 4x 4)の 64 色

例) 暗い赤=C(1,0,0)、明るい緑=C(0,4,0)、明るい黄=C(4,4,0)、黒=C(0,0,0)

このように、RGB の輝度をすべて同等に使う、色の利用者が数値で指示しながら色を作れる方法をフルカラー(fullcolor)とかトゥルーカラー(true color)と呼ばれ、本格的な画像処理で使われています。フルカラーで作れる色、RGB の輝度を対等に使うため、必ず $2^N \times 2^N \times 2^N$ というものになります。

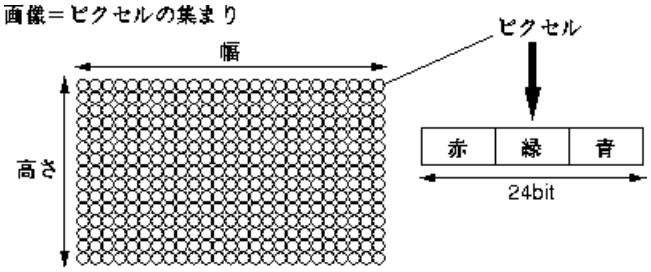


図 2.21: 画像のディジタル表現

フルカラーに対し、文字の色といったようにあらかじめ決められた何色かが使えればよいといった場合は、適当な長さのビット列に色を割り当てるという方法が使われます。この方法では、例えばフルカラーではありえない 4 ビット 16 色というものもあります。当然、この方法では利用者が色を作るということはできません。

【演習】現実に存在する色を表現するには、RGB の各輝度が 256 種類(8 ビット=1 バイト)で十分といわれている。いったい何色になるのだろう。

ビット浪費者としての画像処理

現実に存在する色を本格的に使うフルカラーでは、1 ピクセルあたり 8 ビット×3(1 バイト×3)かかります。そこで、640x 480 というパソコンとしてはもっとも荒い画面とされる 1 画面に画像を描くとすると、約 92 万バイト(約 740 万ビット)が必要です(3 バイト×640x 480=921600 バイト=7372800 ビット)。これがいかに巨大化は、よく使うものと比較すれば分かります。

1. 保存媒体としてよく使われるディスケット 1 枚の容量は 144 万バイト。つまりディスケット 1 枚では 2 画面の保存が難しい。
2. JIS 漢字コードでは、1 文字 2 バイト。480x 640 の画像は約 41 万文字に対応する。よく使われる 400 字づめ原稿紙で何と 1 万枚強にあたります。
3. 家庭用高速デジタル回線(ISDN)では 1 秒間に約 6 万 4 千ビットが送れるに過ぎないため、1 画面送るのに約 144 秒(2.4 分)かかることになります。
4. テレビは 1 秒当たり 30 枚分の画像を扱うことで動きのある画面を作っています。3. の転送時間を考えると、画面をかなり小さくしても、実用的なテレビ電話はとても作れそうもありません。

フルカラーまで色の種類を増やさない場合でも、まず見られる写真画像を扱うには色用として 16 ビット（2 バイト）は必要となります。ともかく画像は莫大なビットを消費するのです。

【ちょっと気をつけて!】漢字やアルファベットといった文字も画面には、例えば 24x 24 といったピクセルのパターンで表示されることは画像と同じです。文字の処理が画像と異なるのは、予め個々の文字画像パターンをメモリーに記憶させて置き、ディスケットへの保存や通信は文字コードだけで処理しているため画像よりは少ないビットで扱えるのです。このことは、祝電を打つとき、「1011 お願いします」で 1011 用に用意してある文例の結婚祝いの電報が打てるのと同じです。

2.6.2 画像の加工

コンピュータで簡単に画像加工ができる最大の理由は、デジタル画像情報が本質的には数値であり、数値は数学的に計算処理できるということにあります。そして、コンピュータは計算機といわれるように計算は大の得意なのです。

色の置き換え

A 子さんから「部屋の壁紙を変えようと思っています。模様は今のままでいいのですが、色に飽きてきたからです。でも、カタログの見本では部屋のイメージが分かりません。いい方法を教えて下さい」という手紙が来ました。この課題をコンピュータで解決してみましょう。

1. A 子さんの部屋をデジタルカメラで写す
2. 画像をペイント系の画像処理ソフトで読み込む
3. 壁紙の適当な部分から、元の色を抽出する(ピクセルの RGB 値を読み取る)
4. 抽出した RGB の値を適当に調整しながら適当な色を作る
5. 画像中、元の色と一致する RGB の値の部分をすべて調整した色で置き換える。家具などはそのままで、壁紙の色だけが変えられる
6. 気に入った色が見つかるまで、3.から 5.を繰り返す

この内、3.から 6.はコンピュータ自動化できます。なお、この考え方は、ファッションデザインや建築内装などの分野で実用化されています。

画像の合成

昔の SF 映画怪獣映画を見ていると、怪獣と背景の合成された部分がなんとなく分かる場合がありますが、最近のものはとてもうまく合成されています。それは、デジタル化された画像は、コンピュータの中で簡単に合成できるからです。その考え方を、テレビの天気予報を例に見てみましょう。

まずテレビカメラを A,B の 2 台用意します。カメラ A は天気図などを写します。カメラ B は解説者を写しますが、解説者の後ろには単色(ブルーが多い)の壁にしておきます。2 台のカメラで写された映像は A/D 変換されコンピュータに送ります。コンピュータは、入力されたカメラ B のある座標のデジタル画像情報が単色に当たる二進法数値の場合は、同じ座標に当たるカメラ A の数値を出力し、単色以外(解説者)の数値の場合はそれを出力します。出力された数値は D/A 変換によってアナログ情報に戻して通常のテレビ電波として発進します。このようにしてすべての座標の色データを電波として送ると、あたかも天気図の前に解説者が説明しているようは画面がテレビで見えるわけです。

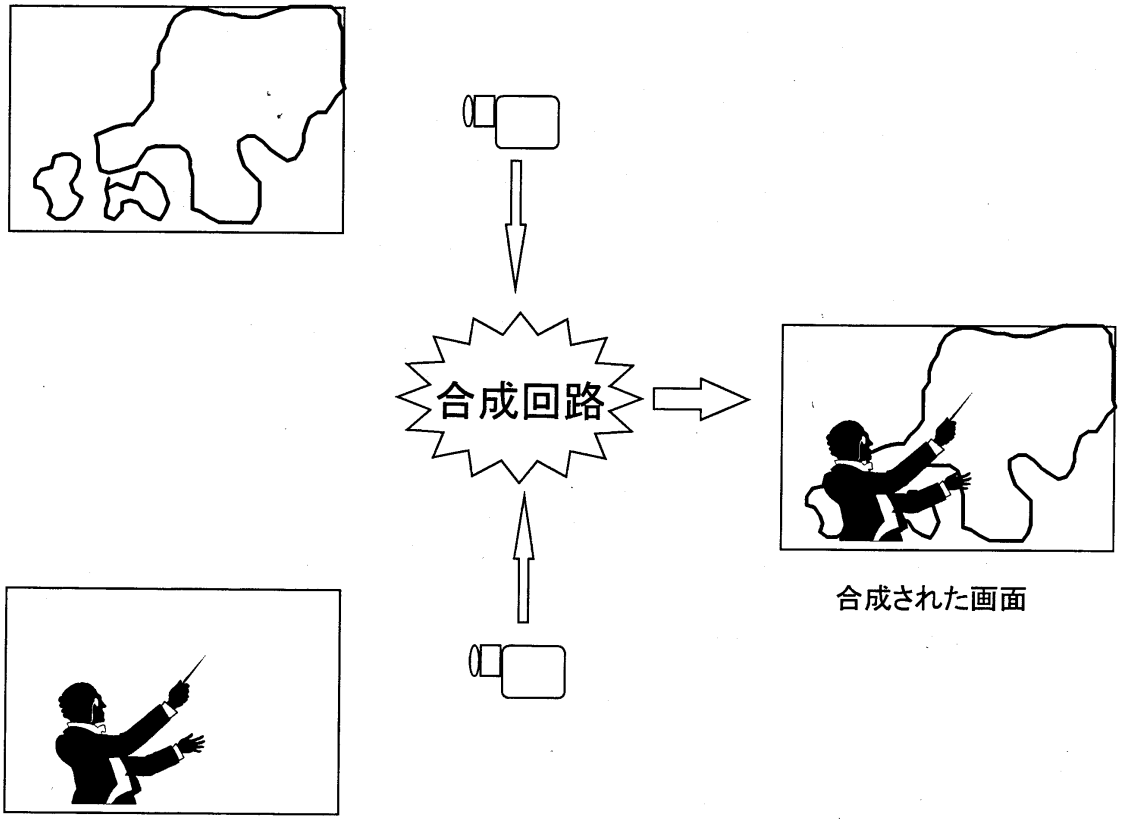


図 2.22: 画像の合成

【ちょっと気をつけて!】もし解説者の顔と頭に壁と同じ色の袋をかぶせ、その上に眼鏡をかけておくと透明人間が解説しているような画面が作れます。

画像の変形

写真をカットガラスごしに見ると、変わった形に変形して見えます。このことを数学的に見ると、ある面 A の図形を別の面 B に写すことにあたります。さて、図形は点の集まりで表せますから、結局 A 面の点(x;y)と B 面の点(s;t)との対応関係を示す関係式を作るという問題に帰結します。この関数が分かれば、デジタルカメラでとった写真を変形してディスプレーに映し出すことができます。これを連続的に行なうと SF 映画のような画像が作れ

ます。
私たちは普段あまり意識していませんが、テレビコマーシャルではこのような原理を利用したものがたくさんあります。

2.6.3 画像情報の圧縮

インターネットで画像を送る場合、各ピクセル毎に色情報をおくっていたのでは情報量が膨大になって実用的ではありません。このため、画像情報の情報量を圧縮する、つまり扱うビット数を減らすという技術がいろいろ考えられています。
すでに学んだように、絵画や写真といった画像はアナログ量で、それをデジタル変換する際に標本化と量子化を行ないましたが、この段階ですでにもとのアナログ情報を圧縮しています。それでも画像が持つ情報量は莫大なため、デジタルの特徴を利用して更なる圧縮が必要なのです。
パソコンで画像を描くソフトには大きく分けてペイント系とドロー系があります。ペイント系のソフトは絵の具で絵を描くようなソフトで、ドロー系はコンパスや定規で作図するようなソフトです。いまドロー系のソフトで円を書いたりそれを保存記録する場合、ドロー系の画像ソフトでは円の中心座標、半径という形で扱っています。したがって、記録しておく情報量も少なくてすみます。この発想は、図形を幾つかのパラメータで扱うという点に着目した画像圧縮技術といえます。しかしこの方法では、写真や絵画のような不定形の養成要素から出来ている画像では扱えません。この場合、風景や絵画の特徴や私たちの視覚分解能などをうまく使うと、画像をデジタル画像に圧縮することができます。その考え方の幾つかを見ていきます。

静止画像の圧縮

私たちが普段目にする風景には、青い空に白い雲が浮かび、芝生の上にピンクの服を着た人がいるというように、同じ色や比較的似た色の広がりでできていることが分かります。このことが静止画像圧縮のポイントになります。

1)同じ色の広がりの場合

同じ色の広がりをデジタル情報に変換すると例えば「赤赤赤赤赤赤赤赤赤」というように、同じ二進法値のビット列が続くことがあります。そこで、A/D 変換した画像に見られるこのような構造を「赤 9 個」というように置き換えていくと、普通は少ないビット数に画像情報を圧縮することができます。元の画像が色として単純なほど、圧縮の程度(圧縮率)は大きくなります。

2)似たような色の広がりの場合

自然に見られる風景では、私たちには同じに見える広がりでも機械を通して測定すると、少しは異なっているものです。私たちは、目の色彩判別能力(分解能)以下の色の違いは分からず、同じ色と判断しています。この原理はすでに量子化誤差の段階でも使われていますが、そこでの基準は実験室で器具を使って濃淡の異なる RGB の各単色を測定するようなものにあたります。これに対し、実際の自然環境下での人間はそれほど細かに、また機械的に一定の分解能で判断していません。たとえば、2 人の人間を真横に並べると 5mm の身長差は分かっても、2m 離れていると同じに見えるのと同じです。実際、256 段階で量子化した約 1670 万色のデジタル画像を数万色に精度を落としても、さほど気になりません。この

ような人間の認知特性（視覚特性）を利用し、似た色を同一色にまとめるという圧縮ができるわけです。

ここで示したような性質をうまく組み合わせると画像の情報量はかなり小さくできます。インターネットでやり取りされる JPEG や GIF 形式画像は、このような原理で圧縮されたものです。

【演習】本文で述べたどちらの方法が JPEG と GIF にあたるか調べてみよう。そして、なぜ２つの圧縮方法が必要とされるか、またそれぞれの圧縮方法の効果が高いのはどのような画像かも調べてみよう。

【演習】画像取り込み装置(スキャナ scanner)でパソコンに取り込んだ画像を非圧縮状態と JPEG や GIF 形式で保存し、ファイルサイズを比較してみよう。また、画像の様子を比べてみよう。

動画像の圧縮

動画圧縮の考え方は、アニメーションと同じです。アニメーションは 1 秒当たり 24 枚必要ですが全部を描いていたら大変です。そこで、アニメーション制作では動かない背景を 1 枚だけ描いておき、人物のように背景の中で動く(変化する)部分だけを透明な用紙に描いたセル画を重ねて撮影しています。これと同じことをデジタル化した画像で行なうと動画の圧縮ができます。映画なら、まずフィルム(アナログ情報)のデジタル化します。変化しない部分に当たるピクセルは同じ二進法の数値、変化部分は異なる数値ですので、コンピュータ内部で 2 画面の数値を比較し、変化した部分だけを取り出します。これを画像情報の差分情報をとるといいます。元の背景に当たる部分と差分とを記録したり伝送したりするのが、動画圧縮の基本的考え方で、代表に MPEG 形式があります。当然ながら、必要に応じて静止画像の圧縮技術も併用されます。

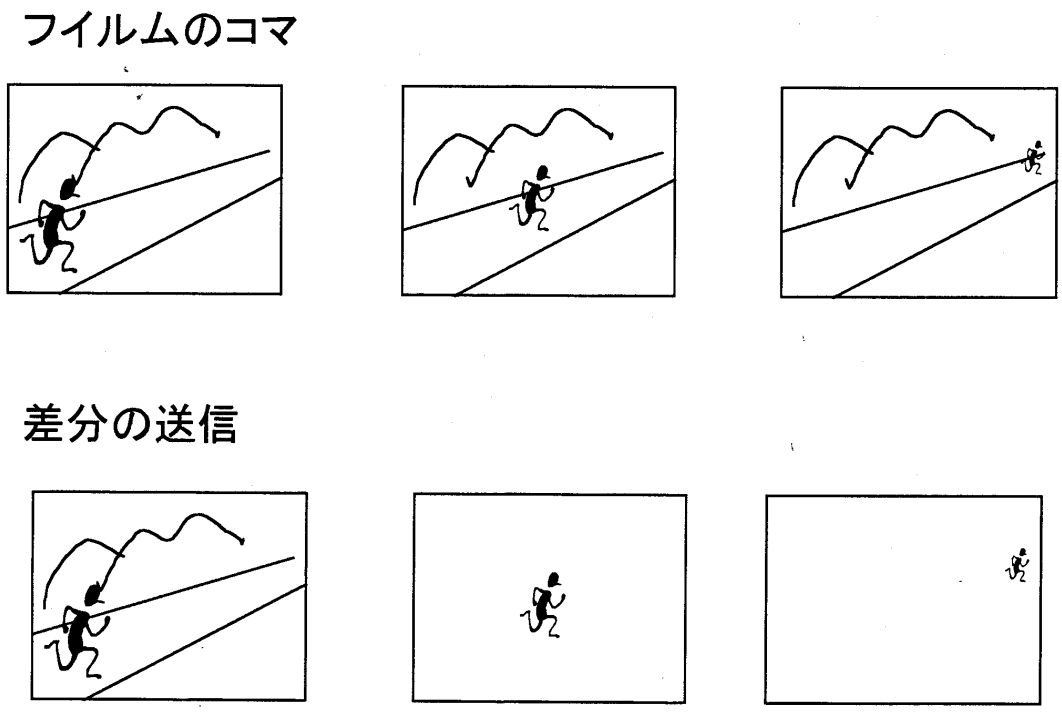


図 2.23: 動画の圧縮(筆者註:下絵のため正確な差分を表わしていない)

静止画にせよ動画にせよ、圧縮はコンピュータ内に入れたプログラムで自動的に素早く処理されますが、それが可能なのは画像がデジタル情報として表現されているからです。

【ちょっと気をつけて!】現在使われているテレビ電話の画面中の動きは非常にぎごちないものですが、その原因は、コンピュータの処理速度よりは現状の電話回線の伝送速度が遅いことに大きく起因しています。テレビ電話の場合も送られてきた画像を一度保存しておき、十分な数だけ溜まってから一気にコンピュータで再生すればまずまずの動きが得られます。しかし、5 秒の動きを滑らかにするため必要な枚数の画像送信に 1 分かけては話になりません。このため、電話では会話のリズムを優先して、画像は 12 分の 1 だけ送るとかして音声とタイミングを取っています。CD-ROM の画像もテレビ電話ほどではありませんが、やはりテレビほど滑らかではありません。CD-ROM 読み取り装置の読み取り速度が十分でなく、また装置とコンピュータ本体との伝送速度がまだ十分でないためです。

【ちょっと気をつけて!】可逆圧縮と不可逆圧縮

画像圧縮は、広く情報圧縮という呼ばれるものの一つの位置分野です。情報圧縮には画像のほか、音や文字でも行なわれています。

最近はコンピュータのプログラムファイルを WWW ページから取る（ダウンロード）ことが多いですが、多くの場合、プログラムファイルは圧縮されています。これは伝送時間を節約するためです。

圧縮されたプログラムファイルはそのままでは使えません。圧縮されたファイルを実行可能なプログラムファイルに戻すことを解凍といいます。

さて、ここで画像圧縮とプログラムの圧縮を比べてみると、同じ圧縮といっても異なっ

た特性のあることが分かります。画像圧縮で色の数を減らすという方法で圧縮した画像は、圧縮された結果の画像の見た目がもっともらしく見えればよいわけですから原図へ戻す必要はありません。これに対し実行プログラムや文章、金銭出納簿などを圧縮した場合は、それが元の状態に復元できなければ意味がありません。

圧縮前の状態に戻せない圧縮を不可逆圧縮、戻せる圧縮を可逆圧縮といいます。

可逆圧縮も興味深いテーマのため、次回の改定で情報伝達の速さとは、を含め追加改定の予定です。

【著者からの予告。本文とは無関係です】当初、可逆圧縮の話は執筆予定外でしたが、高等学校にインターネットが普及するとダウンロードの機会も多くなると考えられます。このため、次回の改定でこの話と、ダウンロード時間のこともありますので情報伝達の速さについて追加改定の予定です。

2.7 融合するメディア

アナログ情報をデジタル情報化することの意義は雑音や減衰の解決に留まらず、情報の表現の上で画期的な出来事なのです。すなわち、本来アナログな「モノ」そのものにくっ付いていた音や画像といった情報が A/D 変換を通して、ビットパターンとして表現されるということは、本質的にデジタルな文字などの人工的「記号」情報とが、同一のものに融合したということです。このことは、音や画像といったアナログ量と、文字や数量といったデジタル量が、ともにコンピュータで取り扱えることを意味します。言い換えれば、従来、異なったメディアで表現されていた情報が同一環境で取り扱い可能となったということです。

この結果、音楽を映像化するとかその逆といった新しい芸術の可能性が模索されたり、書物をスキャナーでコピーするようにしてコンピュータに読み取らせ、それを音読せ、目の不自由な人の福祉に貢献するといったことも考えられます。さらに神経はある種のパルス処理を行っていると考えられますので、画像をデジタル処理し、それを視神経に直接つなげば視力を失った人に視覚をよみがえらせることが可能になるかもしれません。実際、そのような研究もなされています。

また、X 線画像を立体的に表示し、回転させながら指定の場所の音を情報と合成しながら病気を診断するといった明るい応用が多数開発されています。

その一方で、簡単に画像合成できるということは写真の証拠能力がなくなるとか、他人の肖像を勝手に自分の画像を合成してインターネットに流したり、他人の声を合成して他人になりすますといった影の問題も起こっています。

このような情報社会で象徴的な光と影は、画像に限らず、音を含めデジタル情報表現という技術がその底にあるのです。

その一例として、現在、インターネット上で新しいコミュニケーション手段として急速に広がっている Web ページを、マルチメディアの面から見てみましょう。

2.7.1 マルチメディア文書

マルチメディアとは

コンピュータで扱う情報のうちでテキストが占める割合が高いことは説明しましたが、もちろんその他の種類の情報も扱えます。コンピュータで扱うテキスト以外の情報種別のことを総称してマルチメディアと呼びます。まずテキストが基本で、それに他のものを加えるから複数種類の(マルチ)媒体(メディア)、というわけです。しかし技術の進歩とともに、最近ではテキスト情報を全く含まないようなマルチメディアのコンテンツ(内容)も一般的になっています。

【演習】では、情報を伝えるのにテキスト以外にどのような媒体(メディア)の種別があり得るかできるだけ多く挙げよ。続いてそれらを、

1. **既にコンピュータで扱うことが実用化されているもの、**
2. **コンピュータで扱うことが研究されているもの、**
3. **まだコンピュータで扱えるかどうか分からないもの、**

に分類してみよ。

2.7.2 さまざまなマルチメディア素材

ではもう少し具体的に、WWW などで使われているマルチメディアの素材(コンテンツを作る材料)について見てみましょう。それぞれの素材は通常、ファイルの形で保存しておき、必要に応じて取り出して利用します。ある特定種類のメディアの素材でも、ファイルに格納する方法(や、そのメディアの情報を 0、1 に対応づける方法)はひと通りとは限りません。これをファイルフォーマットと呼んでいます。さらに、ファイルの内容が分からなくならないように、ファイルには格納されている情報のフォーマットに対応した名前(具体的には拡張子)をつけるのが通例です。以下では代表的なファイルフォーマットについても簡単に触れます。

- **画像 コンピュータの画面に表示されるさまざまな絵は、基本的な素材です。画像のファイルフォーマットとしては、GIF(.gif)、JPEG(.jpeg または.jpg)などがあります。**
- **サウンド 各種の音(サウンド)も素材としてよく使われます。サウンドのファイルフォーマットとしては、AU 形式(.au)、WAVE 形式(.wav)などがあります。**
- **動画 動く画像は、大変インパクトがありますが、情報の量も多いので大きなファイルになります。動画のファイルフォーマットとしては、Quicktime 形式(.mov)、VideoforWindows 形式(.avi)、MPEG 形式(.mpeg または.mpe)などがあります。**
- **アプリケーション ワープロソフトや、表計算ソフトや、アニメーションソフトなど、各種ソフト固有のファイルを媒体(メディア)として使うことや、任意のプログラムを媒体(メディア)として使うこともあり得ます。その場合、アプリケーションやプログラム独自の機能が使えるという利点がありますが、どこででもそのようなアプリケーションやプログラムが動かせるとは限らないという制約もあります。ファイルフォーマットはアプリケーションやプログラムの動く環境に応じてさまざまです。**

【演習】用意された素材集を見てみます。また、ファイルの大きさと素材の内容にはどの

ような関係があると思うか、考えてみよ。

2.7.3 マルチメディア文書の設計

テキスト以外の情報も含んだ文書のことを、マルチメディア文書と言います。また、使用するソフトウェアにもよりますが、マルチメディア文書ではテキストの部分も、見出し、本文、引用、強調箇所など内容に応じて字体や文字の大きさや色を変えるなど、表現の工夫が可能である場合が多いです。

マルチメディア文書では、テキストだけで表せないことを読み手に伝えられるという点では優れていますが、読み手、書き手の双方において、各種のメディアや表現などに目を奪われてしまい、肝心の伝えたい内容がおろそかになってしまうというおそれもあります。

そのような罠に陥らないためには、文書が伝えたい内容が何であるかを明確にして、文書の構成を決め(ここまでは普通の文書の設計法と同じです!)、それに沿って各種のメディアや表現の使い方を決めて行くべきです。

各種メディアや表現の使い方については、次の点に気をつけましょう。

- 画像は、必要以上に大きくせず、図などテキストで表現できない情報を効率的に伝える、人物の顔を知らせる、確かに伝えようとしている出来ごとがあったことを納得してもらう、など明確な目的を持って使う。
- サウンドや動画は、それ自体が重要な情報である場合(しゃべり方や音曲そのものを伝えたい、動きそのものを伝えたい等)に限定して使うのがよい。
- 文書に「臨場感を与える」飾りものとして画像、サウンド、動画を使う場合には、最小限にとどめる。長いサウンドや動画はくりかえし見る人には退屈であり、コンピュータに与える負荷も大きくなる。
- 文書の表現を工夫するときは、「どんな項目があるかさっと見てとって欲しいから見出しを大きくする」「この段落だけは熟読して欲しいからそこの色を変える」など目的を持って行い、その目的と関係のない表現は控えます。

【演習】先の演習で見た素材を使って、マルチメディア文書を企画しなさい。テーマは先生からもらうこと。

2.7.4 マルチメディア文書の作成

マルチメディア文書を作成する手助けをしてくれるようなソフトウェアを一般にオーサリングソフトと言います。最近ではワープロソフトにも画像などが入れられるものがありますから、両者の区別はあいまいになりつつありますが、(表現の変化はあるにせよ)おもにテキスト情報を使うのがワープロソフト、マルチメディアをおもに考えているのがオーサリングソフトだと言えます。

ここでは空想上のオーサリングソフトを使ってマルチメディア文書を作ってみます。テーマは「本もののトドを見たらすごかったので、その感動を伝える」ことです。構成は次のように設計します。

(1)いつ、どこでトドを見た。
(2)トドはこんなに大きくて、なきごえも迫力!

(3)こんな芸もしました!

なきごえのサウンドファイルと、さか立ちしている写真の画像ファイルがあるので、それぞれ(2)と(3)に使います。どちらも飾りではなく、伝えたい内容そのものを表現していると言えます。

では作り始めましょう。全体の見出しと、上記(1)〜(3)の小見出しをまず作ります。このオーサリングソフトでは、見出しを作るときには、まずテキストを打ち込み、それをマウスで選択してから、選択メニューで「Heading 1」(第 1 レベルの見出し)等を選びます。すると、選択しておいた項目が見出しのスタイルになります(具体的にはやや大きい文字になる)。小見出しは見て本文と区別がつくように、色を変えてみました(そのためには、変えたい部分を選択してから色のメニューで使いたい色を選びます)。また、全体の見出しと内容を区分するように横線を入れました(図 2.24)。

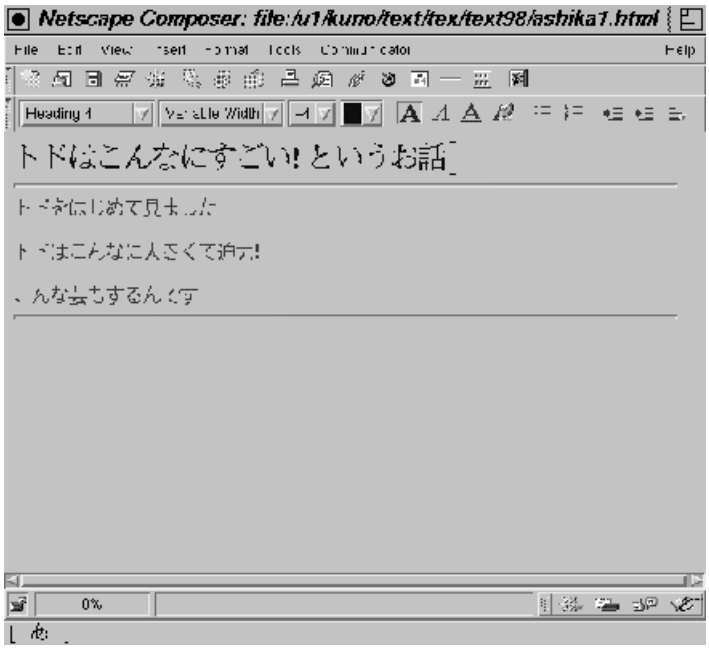


図 2.24: 見出しと横線を入れる

では次に、本文を打ち込みましょう。先頭から順でなくても、書きたいところからでかまいません。コンピュータで文書を作成することの利点の 1 つとして、最終的な順序にこだわらずに作成していけることが挙げられます。見出しを打ち込んだのもこの一環だったのですね。ここでは 3 番目の項目を書いて、そこに写真を入れます。オーサリングソフトの「画像を入れる」機能を選ぶと、ダイアログボックスが表示されて、そこで画像のファイル名や注記、配置などを指定します。指定が終わると、画面に画像が取り込まれて見ることができます(図 2.25)。

サウンドを入れる場合には、音は見えませんから文書に埋め込むことはできず、文書にリンクを用意して、そのリンクを選択すると音が出るようにします。ここでは、「なき声」というテキストをまず選択して、「リンクを作る」機能を選ぶとダイアログボックスが現わ

れますから、そこでサウンドファイルを指定します。別の文書にジャンプするようなリンクも同じようにして作れますが、その場合にはファイルではなく行きたいページの URL を指定します。いずれの場合も、指定を完了すると先に選択したテキストがリンクになっています(図 2.26)。なお、ここではトドの大きさを表の形にまとめていますが、これはオーサリングソフトの表作成機能を使って作れます。

図 2.25: イメージを入れる

図 2.26: リンクと表を入れる

このように、オーサリングソフトを使うと各種のメディアを取り込んだり、いろいろな表現を持った文書を作成することができます。しかし本質的には、各種のメディアというのはファイルであり、マルチメディア文書の側には「このようなファイルをここに埋め込む/リンクする」という情報が(内部的には)記録されているだけです。

【ちょっと気をつけて!】ファイル形式によっては、他のファイルの内容そのものを埋め込むようなものもあります。

【演習】オーサリングソフトを使って、先の演習で企画したマルチメディア文書を実際に作ってみよ。

2.7.5 マルチメディア文書とバインディング

今回使ったオーサリングソフトでは、作成したドキュメントを HTML と呼ばれる形式で保存してくれる。HTML は普通のテキストエディタで見たり編集したりできるので、実際に眺めてみましょう。

```
<HTML>
... 途中略...

<H1>トドはこんなにすごい! というお話</H1>
<HR SIZE=4 WIDTH="100%">
<H4>
<FONT COLOR="#006600">
トドをはじめて見ました
</FONT>
</H1>
<P>○月○日、

水族館ではじめてトドを見ました。
</P>
<H4>
<FONT COLOR="#006600">
トドはこんなに大きくて迫力!
</FONT>
</H4>

<P>
トドは知らないとアシカやアザラシみたいなもののように思えますが、
実物はものすごく大きいです
(下表参照)。
また、
<A HREF="todo.au">なき声</A>も迫力です!</P>
<CENTER>

<HYOU BORDER COLS=2 WIDTH="70%" BGCOLOR="#FFFFCC" NOSAVE>
<TR>
<TD>
<CENTER>体重</CENTER>
```

これを見ると、打ち込んだテキストそのものがそのまま現われている部分と、テキストにない何だかよく分からないものとが混ざりあっています。ここでテキストにない部分は、「この範囲は見出し」「ここはこういう色」「ここはこういう表」といった情報を表しています。

このように、もともとの内容(コンテンツ)に別の情報を付加して全体としてより豊富な表現や機能を提供することをバインディング(束縛)と呼びます。ここでは HTML 形式を見ましたが、ワープロソフトはそれぞれ固有の形式のバインディングを付加していますし、コンピュータで扱う多くのファイル形式にも同様の考え方が見られます。

【演習】先の演習で作った文書を眺めて、内容(コンテンツ)とバインディングの間の関係がどうなっているか考えてみよ。