べた語義

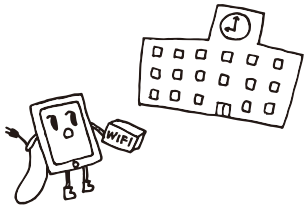
Vol. 105

CONTENTS

COLUMN

1 人 1 台学習者端末について考える

基般



今回の 2018 年 PISA 調査の結果から読解力の低下が報道でとりあげられていますが，単なる読解力ではなく，「ディジタルの世界で求められる読解力の低さ」，つまり日本の学校における「授業において子供たちに ICT 機器を使わせる場面の少なさ」「学習者端末が少なすぎる現状」が明らかになったことのほうが重大な課題だと考えます．政府から「小学校中学校の子供たち 1 人に 1 台ずつの学習者端末を整備」の話が出てきたのは，これまでも言われてきた自治体間の整備率の差を埋めるためにはやや遅すぎるとも思えます．今の整備状況などを把握するため，学校の先生方・教育委員会の方々などを対象に「小学校中学校で使用する学習者端末」について，Web 上でアンケートを行ってみました．関心の高さからか，数日で 100 以上の回答があり，あらためて現状が分かってきました．たとえば，現在の学習者端末の整備状況については，「コンピュータ室に 40 台」しかないという回答が 34.6%，「コンピュータ室に 40 台＋教室で使う端末が数クラス分程度」が 33.6%，つまり 70% 近くが，コンピュータ室に行かなければ使う機会が少ない，ということです．この環境でディジタル情報を分析・理解し，活用する力を子供たちにつけさせようというのはやはり無理があり，だから 1 人 1 台学習者端末の整備が必要なのです．では 1 人 1 台端末が整備されたら，先生方はどのくらい子供たちに使わせるのでしょうか．アンケートの結果では，「ほぼ毎時間使わせたい」という回答が 29.0%，「1 日に数時間」が 46.7%，「1 日に 1 時間くらい」が 15.9% でした．つまり 90% をこえる人が，「毎日使わせたい」と思っている．先生方は「環境があれば子供たちに使わせたい」のです．

しかし，だからといって性急な整備，特に端末を入れることだけを考えた整備は，かえって使えない環境を生みます．これまでも「ネットに全員のパソコンをつないだら動かなくなった」という話は，学校ではよく聞いたものです．1 人 1 台端末を支える校内の高速ネットワーク，学校からネットへの太い回線，端末を充電するための電源確保など，学校という施設を大きく作り変えるくらいのことをしないと，本当に「授業で子供たち一人ひとりが端末を使う」時代は来ないのではないでしょうか．整備を担当する自治体の教育委員会の担当者がそういった俯瞰的な環境整備を構想することが必要であるとともに，外部からの適切なパッケージ提案が行われることが今一番大切なことであると思います．

望月陽一郎（大分県立芸術文化短期大学）

LOGOTYPE DESIGN...Megumi Nakata, ILLUSTRATION&PAGE LAYOUT DESIGN...Miyu Kuno

ARTICLE

実践的演習を伴うサイバーセキュリティ公開講座の取り組み

丸山一貴　明星大学
佐々木伸彦　ストーンビートセキュリティ（株）
高谷宏幸　マカフィー（株）

興味の入口としての公開講座

一般の方にとってはサイバーセキュリティというと高度に複雑であり，理解が難しく恐いものであると思われがちである．また，報道される際には事件としての側面から，個人情報漏洩の規模や被害額，責任の追及といった部分が大きく取り上げられ，技術的な側面は置いてきぼりになることが多い．そうした中，2014 年にサイバーセキュリティ基本法が成立し，セキュリティ分野の人材育成に一層注目が集まるようになった．企業では社内の人材を教育するため，サイバーセキュリティに関する演習付きの研修を外部に委託するケースが増えた．その一方で，大学では，学生向けに同様の機会を設けたり教材を準備したりするのは難しいこと☆1や，CTF（Capture The Flag）☆2のようなイベントは専門性が高く未経験者が初めに学ぶ題材としてはハードルが高いという問題があった．

セキュリティではプログラムや OS，ネットワークといった幅広い分野の理解が必要なことから，学生が自身の専門分野を活かせる可能性があることと，専門分野以外にも興味を持つことの必要性を感じてもらうことが重要である．そこで我々は，PC の操作は問題なくできるが，セキュリティについては素人である 10 代の生徒・学生を主な対象として，実践的な演習形式でその一端に触れる機会を設けようと考えた．2015 年に第 1 回を開催して以降，毎年 1 回，明星大学情報学部主催の無償の公開講座としてこれを運営している（図 -1）．2019 年のアンケートでは参加者の約 90% が「非常に満足」「満足」と回答し，後述する演習参加者に限定すると「満足」以上は 100% であった．本稿ではこの取り組みの概要と課題について述べる．

☆1　一部の学科にカリキュラムとして導入している大学は存在する．
☆2　サイバーセキュリティ技術に関する競技であり，ネットワークやバイナリ解析等の幅広い分野の技術が必要とされる．

講座の構成と参加形態

公開講座は休日午後の約 4 時間を利用して開催しており，大きく分けて 3 部からなる．第 1 部は講演として，サイバーセキュリティに関する基礎と最新情報を，その年のテーマに合わせて簡単に紹介する．第 2 部は公開講座のメインイベントである演習を行う．前半は前提知識やツール類の紹介をハンズオン形式で行い，後半は用意した課題に取り組んでチームごとの成果を競う．前半で扱う内容は，後半の課題に取り組むためのヒントを兼ねており，参加者は教材に例示されたコマンドの使い方等を参考にしながら課題に挑戦していく．第 3 部は演習内容の講評と全体の総括，質疑応答を行

図 -1　公開講座の様子

うとともに，優秀な成果を上げたチームを表彰する．

参加形態は演習参加と聴講参加の 2 種類を設定している．演習参加は，第 2 部で実際に 1 人 1 台のノート PC を使って課題に取り組むことができる．生徒・学生の優先枠を設け，2019 年は 36 名の演習参加者を募集した．演習参加者は 3 人で 1 チームを構成し，協力して課題に取り組む．聴講参加は演習を行わず，第 2 部では演習参加の様子を見守る形になる．

講座の運営には 2 つの課題があり，1 つは参加者の前提知識やスキルの違いである．演習参加者は高校生と大学生（大学院生を含む），社会人からなるが，間口を広くするという意図から，参加募集の際に専門的な知識や経験を要求していない．結果として，(1) 教材に出てくる用語等が分からない参加者や，(2) キー入力により起動するコマンドラインツールの取り扱いに不慣れで，第 2 部前半のハンズオンをスムーズに進められない参加者が存在している．

(1) の対策として，用語等の必要な知識を事前に学習できる簡易な予習教材を準備し，明星大学の学習管理システム☆3 を通じて演習参加者に配布している．2019 年は，事前登録した演習参加者の約 84% が予習課題にアクセスしていた．(2) に対しては，いわゆるティーチングアシスタントのようなサポートスタッフを数名配置して，個別の質問に対応しながら演習を進めることで解決を図っている．

講座運営のもう 1 つの課題は，聴講参加者の支援である．聴講参加者は，演習中は室内を自由に移動して演習参加者の PC 画面を見ながら，進捗状況を見守る形になる（図 -2）．しかし，画面内のコマンドの表示などは見づらく，また，各チームの進捗状況を把握することは難しい．この対策については後述する．

演習のテーマ

第 2 部の演習で出題する課題のテーマは年により異なり，サイバー攻撃を扱う回とディジタルフォレンジックスを扱う回に分けられる．ここではそれぞれの課題の概要と，いずれの回でも共通して課しているレポーティングについて述べる．

❑ サイバー攻撃

サイバー攻撃を扱う回では，演習参加者は攻撃者の立場になり，運営側である我々が用意した攻撃対象のシステムを外部から調査する．システムには典型的な脆弱性☆4 が残されており，そこからシステム内部に侵入して重要情報を取得する，というシナリオである．攻撃者の視点を学ぶことで，防御側，すなわち攻撃を受ける側としてどのような対策が必要か，何を学ぶべきかを考えるきっかけを提供している．

2015 年の第 1 回は，当時のマカフィー(株)がトレーニングプログラムのために用意していた課題をベースにしており，演習参加者は攻撃側と防御側に分かれていた．しかし，防御側の参加希望者が少ないため，現在は攻撃側のみを募集している．防御側の視点は，後述するフォレンジックスの回で取り上げている．

❑ フォレンジックス

フォレンジックスを扱う回では，演習参加者は企業の情報システム部門の立場になる．運営側である我々は，攻撃を受けて情報漏洩が発生した状態の情報システムを準備しておく．演習参加者には社の重

図 -2　演習中の聴講参加者

☆3 講義資料の配布やレポートの提出を管理できる Web システム．いわゆる LMS（Learning Management System）．

☆4 もろくて弱い状態のこと．セキュリティホールのこと．

役から，社外からの通報により個人情報の漏洩が発覚したため，原因と漏洩の経緯を調べるよう業務指示があった，というシナリオである．

現実には，フォレンジックスの作業そのものは社内スタッフでは行わず，専門的な外部企業に委託するケースも多い．しかし，フォレンジックスの作業を理解しておくことで，情報システムの企画や設計，運用において何に注意が必要であるか，意識を高めるきっかけになると考えている．

❑ レポーティング

演習の最後は，どちらのテーマを扱った回でも，必ずチームごとに簡単なレポートを提出する．レポートには，いつ，どんな原因で何が起こったか，どんな対策が考えられるかを，簡潔に記載してもらう（図 -3）．客観的な事実に基づいて，具体的な対策まで含めて報告することで，演習で取り組んだ内容を効果的に振り返ることができる．第 3 部の講評では，レポートの内容について簡単に触れ，課題の進捗がよかったチームや適切な対策を報告したチームを紹介して表彰する．

IoT サイバーセキュリティ演習

ここではより具体的な内容の紹介として，2019 年に実施した IoT サイバーセキュリティ演習について，テーマ設定の背景と当日の進行や参加者の様子を紹介する．

図 -3　レポートの記入

❑ テーマ設定の背景

IoT（Internet of Things，モノのインターネット）は，あらゆる機器がインターネットに接続することで，我々の生活や仕事などを豊かにしてくれることが期待されている一方で，サイバーセキュリティの脅威が懸念されている．2018 年に公開された映画『名探偵コナン ゼロの執行人』でも IoT デバイスが登場し，話題になった．

実際，IoT デバイスに対するサイバー攻撃は年々増加傾向にあり，2016 年には Mirai[☆5] と呼ばれる IoT デバイスをターゲットとしたマルウェアが出現している．Mirai は脆弱な IoT デバイスへ不正侵入し，ボットネット[☆6] と化して，かつてないほどの大規模なサイバー攻撃（DDoS 攻撃[☆7]）を発生させた．その後，Mirai のソースコードがインターネットで公開されたこともあり，Mirai の亜種が次々と発生し，今もインターネット上で活動を続けている．

Mirai の攻撃は，典型的な ID とパスワードの組合せで次々とログインを試みるという非常に古典的な手法だったが，世界中で何十万台もの IoT デバイスがこの方法で不正侵入されてしまった．より身近になる IoT デバイスへの侵害は，これまで以上に，我々の生活に直結する甚大な被害を与えかねない脅威となり得るが，IoT デバイスにおけるセキュリティ対策やその理解は，まだまだ十分な状況とは言えない．デバイスが多様化しても，いつの時代も，セキュリティ対策は基本が重要であることを理解してもらうため，2019 年は IoT に関するサイバーセキュリティをテーマとして演習を実施した．

❑ 演習の題材

第 2 部の演習は，架空の鉄道事業者が運用する

☆5 Mirai は，脆弱な IoT デバイスへ不正侵入し，それらデバイスをボットネットとして悪用するマルウェア．ターゲットとするシステムや Web サイトに対して大量の通信トラフィックを送信し，サービス妨害攻撃（DDoS 攻撃）を行う．

☆6 マルウェア感染や不正侵入などによって，攻撃者の指令によって動く端末のことをボットと呼ぶ．ボットが大量に構成されたものをボットネットという．

☆7 Distributed Denial of Service attack，分散型サービス妨害攻撃．

「鉄道の運行管理システム」をテーマとした．この事業者では，離れた事務所からも電車の走行を把握できるようにして監視業務の運用を効率化したいという要望があり，監視用の IoT カメラが新設された，という状況を設定した☆8．

演習では，走行する鉄道模型（N ゲージ）と信号機に見立てた LED ライト，これらを制御する運行管理システム（Raspberry Pi）からなる演習セットを各チームに提供し，シナリオに沿って課題に取り組んでもらった．シナリオは，IoT カメラを調査して脆弱性を発見し，それを利用することで内部の運行管理システムへアクセスするという流れである．最終的に運行管理システムの認証を突破することができれば，鉄道模型を動かしたり，LED ライトを自由に点灯させたりすることが可能になる（図 -4）．

例年の演習とは異なり，鉄道模型や LED ライトといった目に見える変化を生む機器があることで，より積極的に課題に取り組む様子が見られた．特に，ほかのチームが鉄道模型を動かし始めると，「うちのチームも頑張らないと」という反応があり，会場全体が盛り上がっていた印象を受けた．

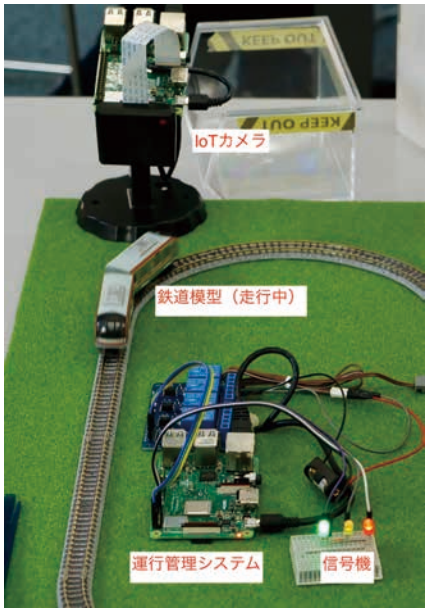


図 -4　鉄道模型を使用した演習セット

☆8 実際の鉄道運行システムはインターネットから隔離されているため，今回のテーマのような事態は発生しない．あくまで，IoT の脅威を勉強するための架空の設定である．

課題と今後の計画

講座の運営については大きく 2 つの課題があり，1 つは演習参加者の前提知識やスキルの違い，もう 1 つは演習中の聴講参加者の支援である．前者の対策についてはすでに述べたが，後者の対策として演習の進捗状況の可視化と解説を検討している．

可視化については Facebook CTF☆9 の利用を試みた回もあったが，単に課題の到達度を表示するだけでなく，チーム内のコミュニケーションの状況や試行錯誤の様子を見せられるとより効果的であると考えている．演習中の講師やサポートスタッフがそれらの表示を指し示しながら解説することで，聴講参加者の理解度向上を期待している．

2020 年度はハードニング（hardening，堅牢化）に焦点を当てた内容を検討しており，これらの改善案を盛り込んで実施することを目指している．

（2020 年 2 月 15 日受付）

丸山一貴（正会員） kazutaka@acm.org

2004 年東京大学大学院修了．博士（情報理工学）．同大情報基盤センター助教等を経て，2013 年明星大学情報学部情報学科准教授．プログラム開発環境やユーザインタフェースの研究，大学における ICT サービスの設計と運用に従事．

佐々木伸彦 sasaki@stonebeat.co.jp

2015 年にストーンビートセキュリティを設立，代表取締役．2016 年から外務省最高情報セキュリティ責任者（CISO）補佐官を務める．脆弱性診断や情報セキュリティコンサルティング，トレーニング講師など幅広く活躍中．CISSP，CISA，GCFA，LPIC-3 Security.

高谷宏幸 Hiroyuki_Takatani@McAfee.com

2005 年マカフィーに入社し，セールスエンジニアとして営業活動の技術的支援業務に従事．2014 年以降，プロフェッショナルサービス本部のシニアトレーナーとしてセキュリティ研修サービスを展開している．

☆9 CTF の運営を支援する Web アプリケーション．現在はオープンソース化されている．

ARTICLE

再帰的思考のすすめ

中川正樹
東京農工大学

今，教員生活を終わろうとしているこの時点で，これまでに「美しい」と思った再帰的な考え方をまとめてみたいと思う．難しいと思われるところは読み飛ばしていただきたい．

数学的帰納法

今から約 50 年前，高校生のときに数学で一番美しいと思ったのは数学的帰納法であった．n=1 のときに成り立つことと，n=k–1（あるいは n<=k–1）のときに成り立つと仮定し，n=k でも成り立つことを示して，したがって，すべての自然数 n で成り立つという証明法である．例として，次の等式を証明してみよう．

$$1+2+3+4+\cdots+n=\frac{1}{2}n(n+1)$$

n=1 のときは，$1=\frac{1}{2}1(1+1)$ なので，明らかに成り立つ．次に，n=k–1 のときに成り立つと仮定する．つまり，n に k–1 を代入して，

$$1+2+3+4+\cdots+(k-1)=\frac{1}{2}k(k-1)$$

が成り立つと仮定とする．これを用いて，次のように n=k でも成り立つことを示す．

$$\begin{aligned}&1+2+3+4+\cdots+k\\&=1+2+3+4+\cdots+(k-1)+k\\&=\frac{1}{2}k(k-1)+k=\frac{1}{2}k(k+1)\end{aligned}$$

したがって，n=1 で成り立って，2 で成り立って，. . . ，k–1 で成り立って，k で成り立つというやり方である．

数式微分

大学 3 年生のときに，LISP という言語で数式微分のプログラムを書いた．数値微分ではなく，入力に x^2 を与えると $2x$ を返す微分である．変数がアルファベット 1 文字だけで，数も 1 桁，かつ，四則演算だけの数式は次のように定義できる．下で，「｜」は「または」の意味である．

式＝項＋式｜項－式,
項＝因子×項｜因子／項,
因子＝(式)｜変数｜数,
変数＝ a|b|…|x，数 =0|1|…|9.

ある数式の x による微分 D は次に従う．

D(項＋式) = D(項) + D(式),
D(項－式) = D(項) − D(式),
D(因子×項) = D(因子) ×項＋因子× D(項),
D(因子／項)
　= (D(因子) ×項－因子× D(項)) ／ (項)^2,
D((式)) = (D(式)),
D(変数) = if 変数 = x, then 1 else 0,
D(数) = 0.

数式微分のプログラムは，表記の違いはあるが，原理的にほとんどこの通りである．そして，実行は，たとえば次のようになる．

D(x^2+3x+1)=D(x^2)+D(3x+1)
=D(x)×x+x×D(x)+D(3x)+D(1)
=1×x+x×1+D(3)×x+3×D(x)+0

$=x+x+0\times x+3\times 1+0=2x+0+3+0$

$=2x+3$

プログラムを書いてみると，数式の微分そのものよりも，結果を整理する（たとえば，$1\times x=x$，$x\times 1=x$，$x+x=2x$，$0+3+0=3$，無駄な括弧を外すなどのための）プログラムの方がはるかに長くなったのを記憶している．詳しく学びたい読者は文献 1)，2) を参照されたい．

典型的な再帰プログラム

再帰のプログラムで典型的なのは「ハノイの塔」の解法手順を出力するプログラムである．ベトナムのハノイにある寺院には，3 本の柱（A, B, C）と，真ん中に穴の開いた大きさがすべて異なる 100 枚の石でできた円盤があるという．元々はすべての円盤が柱 A に刺さっていて，僧侶たちが 1 枚ずつ移動し，すべての円盤を柱 C に移動しようとしている．僧侶が 100 枚の円盤を移動し終わった瞬間，この世は消滅するという．図 -1 は円盤が 5 枚のときの課題を示す．円盤は重ねられて柱 A に刺さっており，それらをすべて柱 C に移動する．ただし，1 枚ずつしか移動できず，しかも，小さな円盤の上にそれよりも大きな円盤を重ねてはならない．どの円盤も移動後は，いずれかの柱に刺しておかなければならない．そのために柱 B を補助として利用する．

さて，n 枚の円盤を柱 A から柱 C へ，柱 B を補助に使って移動する手順を示す言語 C のプログラム 1 を完成してみよう．1 枚のときは自明である．n 枚のときは，上の n–1 枚を柱 B に移し，一番下の円盤を柱 C に移してから，柱 B の n–1 枚を柱 A による補助で柱 C に移せばよい．プログラムの□の中には，a, b, c いずれかの変数が入るので，適切なもので埋めて実行してみよう．

円盤 1 枚を 1 分で動かせるとして，移動を始めてからどれぐらいの時間で終了するか考えてみてほしい．安心して眠れると思う（なかなか計算できなくて眠れないかもしれないが……）．ちなみに，ハノイには 10 回ぐらい行ったが，いまだにこの寺院には行き当らない．

コンパイラ

筆者が若いころは計算機科学がまだ黎明期で，日本ではまだその教育が確立しておらず，大学院に入ってから，英国の Essex 大学の修士課程に留学する機会を得た．そこでは，初期の人工知能の論文もたくさん読んだが，学部科目で，言語 C などの高水準言語から機械語に翻訳するコンパイラというプログラムを読んだ．プログラムが上例の数式と同

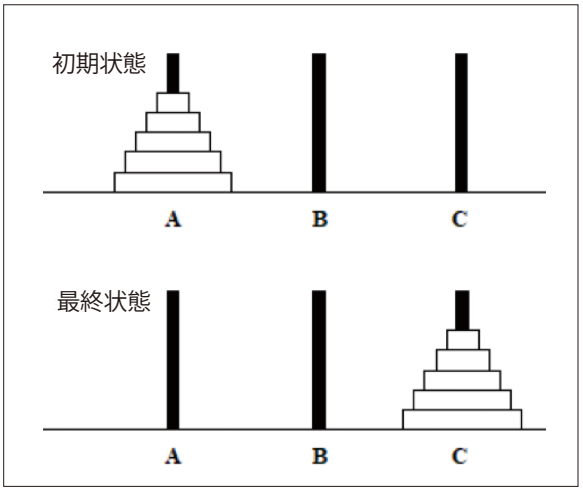


図 -1　ハノイの塔（5 枚のとき）

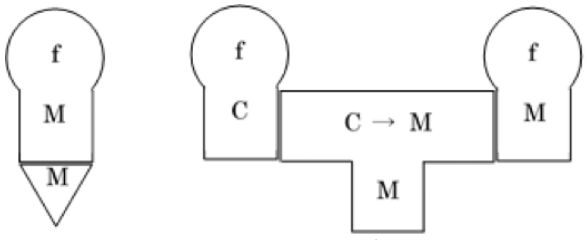
```
#include <stdio.h>
void hanoi(char a, char b, char c, int n)
/* move n disks from a to c using b */
{
   if(n==1) printf("Move a disk from %c
      to %c.¥n", a, c);
   else {
      hanoi(□, □, □, n - 1);
      printf("Move a disk from %c to %c.
      ¥n", □, □);
      hanoi(□, □, □, n - 1);
    }
}
int main()
{
   int n;
   printf("Tower of Hanoi¥n");
   printf("How many disks? ");
   scanf("%d", &n);
   hanoi('A', 'B', 'C', n);
}
```
プログラム 1　解法手順を表示するプログラム

じように厳密な再帰的文法で定義されることを利用して，高水準言語から機械語への変換を再帰法に従って行う．そのプログラムの行数は予想よりはるかに少なかった．この分野の成果は計算機科学の金字塔の１つと言えるだろう．

そして，小さい言語からより大きい言語に拡大していく方法（bootstrapping）にも感嘆した．図 -2（a）に示すように，機械語 M で動作するコンピュータを▽で中に M，機械語 M で記述された機能 f を実行するプログラムを鍵穴形で下に M，上に f で書く．プログラムを実行させることで，f の機能が発揮する．

コンパイラは機能として，言語 C から機械語 M に変換するので，図 -2（b）で示すように，鍵穴の上の半円を四角に変形し，中に C → M と書く．言語 C で書かれた任意の機能 f のプログラムを同じ機能を持つ機械語 M のプログラムに変換する．

最初のコンパイラは小さい言語（μ C と書こう）のために作り，その言語 μ C でフルスペックの言語 C のコンパイラを書けば（図 -3 の左端），そのコンパイラをコンパイルすることで，フルスペック C のコンパイラが得られる（図 -3 中央）．これを動作させれば，任意の機能 f のプログラムをコンパイルできる（図 -3 右）．言語 C だけでなく，別の言語でもよい．興味のある人は Earley [3] を読んでほしい．



(a) 機能 f の実行　　(b) コンパイラの機能

図 -2　記述言語と機能の実行

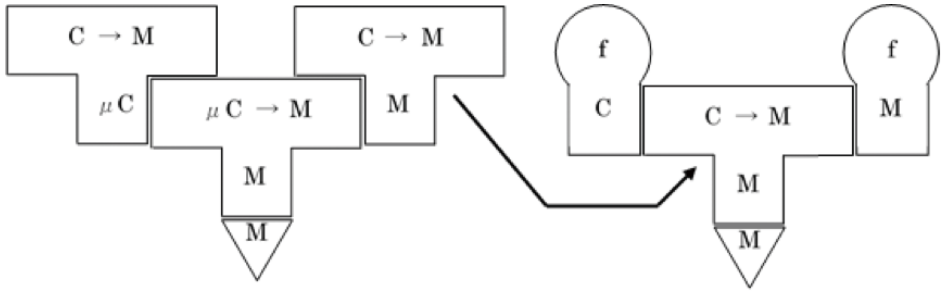


図 -3　小さい言語のコンパイラを作って大きい言語のコンパイラを作る

分割統治法

自然数は１から始まって，それに１を加えたものも自然数という一番単純な構造を持つ．木構造は幹から次々に枝に分かれてどこかで葉になる構造を持つ（そうでない変な植物もあるが）．プログラムは枝や葉に種類がある木構造である．こうした構造を処理するプログラムを書くとき，構造を分解して部分ごとに解法を適用し，それらの結果をまとめることで構造全体を処理する方法を分割統治法と呼ぶ．要素数が n のときに，たとえば，分割しないと n^2 の処理時間になる方法を，上手に分割することで $n \log n$ で処理できるようになる．数学やプログラムあるいはデータのように明確な構造を持つ対象だけでなく，さまざまな対象で分割統治法は採用されている．学校のクラス分けやグループ分け，企業や組織の部や課への分割とそれらの管理である．大規模な問題を効率的に処理できるプログラムを書ける人は大組織の管理や運営にも能力を発揮するはずである．したがって，誰もがプログラムを書いてみることに賛成である．逆に，プログラムを書いたこともない経営者や管理者が，情報システムの企画責任者になることにはリスクを感じざるを得ない．メガ銀行が統合するときに起きたトラブル，大手コンビニの電子決済システムの穴など，枚挙にいとまがない．

落語

修士課程を出て，経済的な理由から博士課程に行かずに農工大の助手として採用してもらった．講座の教授は高橋延匡（のぶまさ）先生だった．計算機システムで多くの業績を残された先生なのだが，落語好きで「えんきょうさん」と呼ばれていた．その先生から落語には再帰の概念も登場するという話を聞いた．それは「あたまやま」である．

あるケチな男がサクランボの種を捨てるのが惜しいと飲み込んだら，頭のてっぺんから桜が生えてきて，春になると満開になる．町人が花見に押しかけて宴会をする．しかし，そのうるささに耐えかねて，桜を抜いてしまう．すると，その穴に雨が溜まって池になり，魚が増える．今度は，町人が昼夜を構わずに釣りに来る．うるさくて眠ることもできない．その男は「いっそ死んでしまえ」と，その池に身を投げる．無理矢理に図にすると図 -4 のようになる．

別の例は「のっぺらぼう」である．ある商人の男が身投げをしようとする若い女を助けてみると，顔に目も鼻も口もないのっぺらぼうである．男は「世の中，なまじ目や鼻がついているために苦労している女は五万といる」となだめつつも，内心では血が引く思いで，蕎麦屋の屋台に駆け込む．息も切れ切れにその話をすると，蕎麦屋は「こんな顔ですかい」とのっぺらぼうの顔で振り向く．男は慌てて家に逃げ帰り，奥さんにその話をする．すると，奥さんが「こんな顔？」と振り向く．男は気を失う．しばらくして，奥さんに起こされて男は目を覚ます．「変な夢でも見てたんじゃない」と言われて，夢だったのかと，その夢の話をする．そうすると奥さんが「こんな顔？」と振り向く．男は気を失う，…….

これらの話は繰り返し（loop）ではない．再帰（recursion）なのだ．

図 -4　あたまやま

振り返って

再帰が美しいのは，一番単純な場合の処理を定義し，単純でない場合には問題を分解して部分の処理に自分自身を呼んで結果をまとめる記述をすればよいことである．何回ループするなどの記述は不要である．先人は再帰プログラムを実行できるようにするために，計算を途中で保留して，次の計算を始め（その計算をまた途中で保留し，次の計算を始める，……），結果が返ってきたときに計算を再開するためのスタックというデータ構造を生み出した．

当方は農工大の助手に採用してもらって 41 年間，手書きパターン認識とタブレットのユーザインタフェースの研究をし，実用化もできた．

この原稿とあまり関係のない研究のように思われるかもしれないが，課題解決のさまざまな局面で再帰が登場した．まさに，自分の思考法の柱の 1 つが再帰であった．

参考文献

1) 平野拓一：記号数式処理，http://www.takuichi.net/hobby/symbolic/
2) 渡邉慶一：LIST でやる記号微分，https://www.slideshare.net/KeiichiWatanabe/lisp-102975121
3) Earley, J. and Sturgis, H.: A Formalism for Translator Interactions, Communications of the ACM, 13, 10, pp.607-617 (1970).

（2020 年 1 月 12 日受付）

中川正樹（正会員） nakagawa@cc.tuat.ac.jp

1979 年東京農工大学工学部助手．1997 年教授．現在，副学長．手書きパターン認識とユーザインタフェースなどの研究・教育に従事．理学博士．2016 年文部科学大臣表彰，東京都功労者表彰など受賞．