Computational Thinking

計算論的思考

基般

Jeannette M. Wing（Microsoft Research and Carnegie Mellon University）
翻訳：中島秀之（公立はこだて未来大学）

［原文］Jeannette M. Wing：Computational Thinking, Communications of the ACM, Vol.49, No.3, pp.33-35（Mar. 2006）より許可を得て翻訳.

これは Wing の 2006 年のエッセイである．これが出た当時，我々日本の研究者仲間も似たような感覚を持っていたので，このエッセイを歓迎した．すぐに誰かが翻訳するものだと思っていたら，2014 年の現在に至るまでその気配はない．書いてあることが我々研究者には当たり前だった（のでわざわざ翻訳しようと思わなかった）し，このエッセイを読んでほしい世間の人は CACM のこの記事の存在すら知らなかった（のでわざわざ翻訳してほしいと思わなかった）のだろう．

2014 年の今になって読み返してみると，その主張は少しも古びていない．一部はすでに実現してしまっているし，逆に間違いと分かったような記述は 1 つもない．その意味では 8 年を経てかえって重みが増したのではあるまいか．

2014 年 10 月の Microsoft Research Asia のシンポジウムで北京に招待された折に Wing が講演していて，この主張に再会した．早速翻訳を申し出て快諾してもらった（版権は ACM にあるので，その許諾が必要とのコメント付きで）．実は自分で途中まで訳しかけてそのまま忘れていたのだ．出版社からの催促のない翻訳というのは完成しないものだ（催促があっても完成するかどうか怪しいのに）と思った．今回とり急ぎその続きの作業を行ったのが本稿である．

なお，英語の “compute” の語感は日本語の計算よりは広範囲を指し示すように感じている．日本語としては情報を操作する感覚に近いものである．しかしながら，“computer” には計算機という定訳があり，いまさら情報処理装置と呼ぶわけにもいかない．悩んだ末，“compute” は「計算する」，“computational thinking” は「計算論的思考」と訳すことにしたが，それ以外の “computer” は「コンピュータ」とした．ゆえに，“computer science” は「コンピュータ科学」，“computer scientist” は「コンピュータ科学者」とした．（訳者）

このエッセイはコンピュータ科学者だけではなく，すべての人が学び，そして使いたいと考えるに違いない一般的な態度とスキルに関するものである．

計算論的思考は計算プロセスの能力と限界の上に成立しているもので，計算の主体が人間であるか機械であるかは問わない．計算手法と計算モデルは，我々個人が単独では決して達成できないであろう問題を解決したリ，システムをデザインしたりする勇気を与えてくれる．計算論的思考は機械知能の謎に挑む：人間がコンピュータより優れているのは何か？　コンピュータが人間より優れているのは何か？　さらに最も根本的な問題がある：何が計算可能か？　これらの問いに対して，我々はまだ部分的な回答しか知らない．

計算論的思考は，コンピュータ科学者だけではなく，すべての人にとって基本的な技術である．すべての子供の分析的思考能力として，「読み，書き，そろばん（算術）☆1」のほかに計算論的思考を加えるべきである．印刷，出版技術が 3R の普及を進めたように，コンピュータ科学と計算装置が計算論的思考を普及させることを忘れてはならない．

計算論的思考は問題解決，システムのデザイン，そして基本的なコンピュータ科学の概念に基づく人間の理解などを必要とする．そして，コンピュータ科学の広がりを反映したさまざまな思考の道具も含

☆1 英語では Reading, wRiting, aRithmetic の 3R と呼ばれている

まれる．特定の問題を解くときに私たちは，どれくらい難しいか？　最善の解決の道筋は何か？　などを考える．コンピュータ科学はこれらの問いに正確に答えるための確固とした理論の上に構築されている．問題の難しさを述べるためにはその土台となる機械—問題の解決策を探る計算装置—の能力を知らねばならない．機械の命令セット，計算資源の制約，動作環境などを考慮しなければならない．

問題を効率良く解くためにはさらに，近似解で良いのか，乱数化をうまく利用できるのか，また解の判定に偽陽性や偽陰性が許されるのか，などを考慮せねばならない．計算論的思考は一見難しそうな問題を我々がすでに解き方を知っている問題に変換する．これには簡略化，埋込，変換，シミュレーションなどが使えるだろう．

計算論的思考とは再帰的に考えることであり，並列処理であり，命令をデータとし，データを命令とすることである．それは次元解析の一般化としての型検査である．それは人やモノに 2 つ以上の名前を付けること（エーリアシング）の利便性と危険性を理解することである．それは間接アドレスや手続き呼出しのコストと威力を認識することである．それはプログラムを正しさと効率からだけではなく，美学的基準や，システムデザインの単純さと洗練度からも判断することである．

計算論的思考とは巨大で複雑なタスクに挑戦したり，巨大で複雑なシステムをデザインしたりするときに，抽象化と分割統治を用いることである．それは問題点の分割である．それは問題の適切な表現法を選ぶことであり，問題を解きやすくするために問題の適切な側面だけをモデル化することである．それは不変項を見つけてシステムの振舞いを簡潔かつ宣言的に記述することである．それは，すべての細部にわたり理解することなく巨大複雑系を使いこなし，変更し，影響を与えることが自信を持ってできることである．それは複数のユーザに備えてサブシステムをモジュール化したり，将来の利用に備えてデータをプリフェッチしたりキャッシュしたりすることである．

計算論的思考とは予防，防御，そして最悪のシナリオからの復帰という観点を持ち，そのために冗長性，故障封じ込め，誤り訂正などを用いることである．それはグリッドロック（超渋滞），デッドロックを判定し，コントラクトインタフェースを起動することである．それは会議を設定するときに競合条件の回避を行うことを学ぶことである．

計算論的思考はヒューリスティックな推論により解を発見することである．それは不確定な状況でのプランニング，学習，スケジューリングのことである．それは探索して，探索して，そしてさらに探索して Web ページのリストや，ゲームに勝つ戦略や，あるいは反例を見つけることである．計算論的思考は超大量のデータを使って計算を高速化することである．それは時間と空間のトレードオフ，あるいは計算パワーと記憶容量のトレードオフをすることである．

以下のような日常の例を考えてみよう：あなたの娘さんが朝学校に行くとき，その日必要なものをカバンに詰める—これはプリフェッチとキャッシュである．あなたの息子さんが手袋を失くしたとき，来た道を逆戻りすることを勧める—これはバックトラックである．どの時点でスキーのレンタルを止めて自分用のを買うか？—これはオンラインアルゴリズムである．スーパーマーケットのレジでどの列に並ぶだろうか？—これはマルチサーバシステムの効率モデリングである．あなたの電話はどうして停電中も通じるのか？—これは故障からの隔離であり，デザインの冗長性である．完全自動チューリングテストはどのようにしてコンピュータと人間を見分けるのだろうか，あるいは人間を認証するのだろうか？—これは AI の解決困難な課題を利用して計算エージェントに箔を付けることである．

アルゴリズムや前提条件といった用語が人々の日常的語彙となり，非決定性やゴミ集めの意味がコンピュータ科学者の使うものに変化し，木が上下逆に描かれるようになったとき，計算論的思考は生活の必須要素となる．

私たちは計算論的思考が他の研究領域に与える影

響を目撃してきた．たとえば，機械学習は統計学を変えた．統計的学習は，ほんの数年前には考えられなかったような，データ量と次元の巨大な問題に適用された． すべての組織の統計部門はコンピュータ科学者を採用し始めた．コンピュータ科学の学部では統計学科をすでに擁立していない場合には，新しく設立している．

コンピュータ科学者の生物学に対する最近の興味は，生物学者が計算論的思考から恩恵を受けると信じていることに後押しされている．コンピュータ科学者の生物学に対する貢献は，単に大量のゲノムシーケンスデータから特定のパターンを見つけ出すことにとどまらない．データ構造とアルゴリズムという，我々の持つ計算的抽象化と方法論が，タンパク質の構造を，構造から機能が明らかになるような形で表現できることが期待されている．計算生物学は生物学者の思考法を変えつつある．同様に，計算ゲーム理論は経済学者の思考法を，ナノコンピューティングは化学者の思考法を，そして量子計算は物理学者の思考法をそれぞれ変えつつある

このような思考法は他分野の科学者だけでなくすべての人に必要な技量の 1 つである．ユビキタスコンピューティングが今日にもたらした影響と同様のものを，計算論的思考が明日にもたらす．ユビキタスコンピューティングは昨日の夢が今日の現実となったものであり，計算論的思考は明日の現実である．

＜コンピュータ科学者のように考えるということは，コンピュータをプログラムできるということ以上の意味を持つ．複数のレベルの抽象思考が必要である＞

それは何であり，何でないか

コンピュータ科学とは計算に関する，すなわち計算可能性と計算方式の，学問である．

したがって計算論的思考は以下の特徴を持つ：

―概念化のことであり，プログラミングではない．コンピュータ科学というのはコンピュータをプログラムすることではない．コンピュータ科学者のように考えるということは，コンピュータをプログラムできるということ以上のものである．それは複数の抽象レベルで考えることを要求する．

―基礎的な技能であり，機械的なものではない．この基礎的な技能は，現代社会で活動するためにすべての人が知らねばならないものである．機械的というのはルーチンワークのことである．皮肉なことに，コンピュータが人間のように考えるという AI のグランドチャレンジをコンピュータ科学が解決するまでは，思考は機械的である．

―人間の思考法のことであり，コンピュータのそれではない．計算論的思考は人間の問題解決法であり，人間がコンピュータのように考えることを目指すものではない．コンピュータは単調で退屈であるが，人間は賢くて想像力豊かである．人間がコンピュータを刺激的なものにする．コンピュータという計算装置を持つことにより，我々は計算の時代以前には挑戦できなかったような問題を解くのに自らの叡智を使うことができ，新しいシステムを構築することができる．限界は我々の想像力だけである．

―数学的思考と工学的思考を組み合わせ，補完することである．コンピュータ科学は本質的に数学的思考の上に成立している．そのため，すべての科学同様，コンピュータ科学の形式的基礎は数学にある． コンピュータ科学は，実世界と相互作用するシステムを構築する場合，本質的に工学的思考の上に成立している．それらを司る計算装置の制約が，コンピュータ科学者に数学的だけではなく計算論的な思考を要求する．仮想世界を自由に構築できるため，物理世界の制約を超えたシステムの構成が可能である．

―概念であり，モノではない．我々が創造するものは単なるソフトウェアやハードウェアという，物理的にどこにでも存在し，いつでも触れることのできるモノではなく，問題に迫り解決するための計算論的な概念で，我々の日常生活を助け，他の人々とコミュニケーションをとり交流するためのものである；そして

―それは，すべての人にどこでも．計算論的思考

は，人間の努力と一体化してしたときに現実となり，明示的に哲学する必要性は消えてしまう．

多くの人がコンピュータ科学をコンピュータのプログラミングのことだと思っている．コンピュータ科学を専門とする子供たちの就職先の可能性を狭く捉える親がいる．またコンピュータ科学の基礎的研究は完了していて技術的問題だけが残っていると，多くの人が考えている．計算論的思考は，この分野に対する社会通念を変えようとする

コンピュータ科学の教育者，研究者，そして実務家を導く主要な観点である．特に，大学入学前の学生とその教師や親たちを含む人々に対し，以下の 2 つのメッセージを送る必要がある：

—知的に挑戦的で魅力的な科学的問題が多く残されている．問題領域と解決策領域を限定しているのは我々の好奇心と創造性だけである；そして

—コンピュータ科学を専攻した学生は何を専門にしてもよい．英語や数学を専攻した学生は異なる分野で複数のキャリアを追求しているではないか．コンピュータ科学もしかり．コンピュータ科学を専攻した後に医学，法律，経営，政治，そしてあらゆる種類の科学や工学，さらには芸術の分野に進むことができる．

コンピュータ科学の教授は「コンピュータ科学者のように考える方法」と名付けた科目を，大学の新入生に教えるべきである．そしてそれはコンピュータ科学専門の学生だけでなく他学科の学生たちにも開放すべきである．大学以前の学生にも計算手法やモデルに触れる機会を作るべきである．コンピュータ科学に不満を述べたり，それに対する興味を否定するのではなく，あるいはコンピュータ科学の研究費を却下したりしないで，一般の人々の興味をこの分野の知的冒険へと導くべきである．そのようにしてコンピュータ科学の喜び，恐怖，威力を広め，計算論的思考を一般的なものにしたい．

Jeannette M. Wing（wing@cs.carnegie.edu）はカーネギーメロン大学（ピッツバーグ）の「学長」教授で，コンピュータ科学科の学科長である．

（2015 年 1 月 14 日受付）

訂　正

本誌 56 巻 6 号（2015 年 6 月号）の解説「Computational Thinking 計算論的思考」に一部誤りがありました．お詫びして訂正いたします．

P.585　右段 4 行目
（誤）それはグリッドロック（超渋滞），デッドロックを判定し，コントラクトインタフェースを起動することである．
（正）それは（世の中で超渋滞と呼ばれている）グリッドロックをデッドロックと呼び，（世の中で契約と呼ばれている）コントラクトをインタフェースと呼ぶことである．
P.585　右段 32 行目
（誤）これは AI の解決困難な課題を利用して計算エージェントに箔を付けることである．
（正）これは AI の解決困難な問題（の性質）を利用して，（悪意のある）計算エージェントを妨害することである．
P.586　右段 6 行目
（誤）皮肉なことに，コンピュータが人間のように考えるという AI のグランドチャレンジをコンピュータ科学が解決するまでは，思考は機械的である．
（正）皮肉なことに，コンピュータが人間のように考えるという AI のグランドチャレンジをコンピュータ科学が解決すると，思考は機械的であることになってしまうが．