Jeannette M. Wing: Computational Thinking, Communications of the ACM, Vol.49, No.3, pp.33-35 (Mar. 2006)

計算論的思考〔学習用翻訳〕

周以真 Jeannette M. Wing
(Microsoft Research and Carnegie Mellon University)

コンピュテーショナル・シンキング（計算論的思考）

それは，コンピュータ科学者だけでなく誰にでも例外なく応用できる態度と技能であり，是非とも学んで利用すべきものです。

計算論的思考とは，コンピューティング・プロセスの能力と限界にもとづくもので，それが人間によるものか，機械によるものかは関係ありません。計算論的な方法とモデルは，私たちが単独で挑むには困難が伴う問題の解決やシステムの設計に勇気を与えてくれます。計算論的思考が対峙している機械知能の謎には：人類がコンピュータよりも優れているのは何か？そして，コンピュータが人類よりも優れているのは何か？といったものがありますが，これらが問うている最も根本的な問いとは：計算可能（computable）であるとは何か？ということなのです。今日，私たちにわかっているのはそうした問いに対する部分的な答えだけです。

計算論的思考は，すべての人にとっての基本的な技能であって，コンピュータ科学者だけのものではありません。「読み，書き，そろばん（算術）」（3R's）と合わせ，「計算論的思考」もあらゆる子どもの分析的能力の一つに加えるべきです。まさに印刷技術が3R'sの浸透を促したことに引き付けて言うなら，このビジョンにおいてコンピューティングとコンピュータが計算論的思考の浸透を促すのだと言えます。

計算論的思考は，問題を解決し，システムを設計し，人間行動を理解することに，コンピュータ科学の基礎概念を利用します。計算論的思考が含んでいるのは，コンピュータ科学分野の幅広さを反映した思考ツール（mental tools）の集まりなのです。

ある特定の問題を解決しようとするとき，私たちは：解決するのはどれくらい難しいだろう？とか，ベストな解決策は？とか問うでしょう。コンピュータ科学はしっかりとした理論的基盤で成り立っていて，それらの問いに正確に答えようとします。たとえば，解決策を実行しようとするコンピュータ装置，その装置の基本能力を使って問題の難しさを説明するには，装置の命令セットやリソース制約，操作環境についてまで考慮しなければならないのです。

効率的な解決のため，私たちはさらに，近似解で十分なのか，乱数化を効果的に使えるのか，また偽陽性や偽陰性は許されるのか，などを考慮するかも知れません。計算論的思考は，たとえば簡略化，埋込み，変換，あるいはシミュレーションといった手段を使って，難しそうな問題を私たちが解き方を知っている問題に変換してくれます。

計算論的思考とは再帰的思考です。並列処理であり，コードをデータとして，データをコードとして解釈することです。次元解析の一般化における型検査のことであり，ヒトやモノに対して２つ以上の名前を付けてしまう，いわゆるエイリアシングの利便性と危険性を理解することでもあります。間接アドレッシングとプロシージャ呼び出しのコストとパワーの両方を認識することであり，プログラムを正確さと効率だけでなくその審美性とシステム設計の単純さ，洗練さで評価することなのです。

計算論的思考は，巨大で複雑なタスクに取り組んだり巨大で複雑なシステムを設計するときに抽象化と分解を用います。つまり関心の分離です。問題の適切な表現を選んだり，問題を扱いやすくするため関連部分をモデル化することです。また，不変項を用いて，システムを簡潔かつ宣言的に記述することでもあります。巨大で複雑なシステムに対し，細部のあらゆることを知ることなしに私たちが変更や影響を与えることができるという確信をもつことなのです。複数のユーザーを想定して何かしらモジュール化することであり，将来的な利用を見越してプリフェッチとキャッシングをすることでもあります。

計算論的思考は，冗長性や障害抑制，誤り訂正によって，最悪ケースのシナリオを防止したり，そこからの保護や回復について考えることです。交通渋滞（gridlock）をdeadlock（膠着停止）と，契約（contracts）をinterfaces（境界面）と呼ぶことでもあります。さらには，互いに連動してしまった会議の競合状態の回避を学ぶことでもあるのです。

計算論的思考は，ヒューリスティックな推論を使って解を見つけ出すことです。不確定な状況においてプランニング，学習，スケジューリングをすることだといってよいでしょう。検索して，検索して，さらに検索を重ね，Webページのリストや，ゲームに勝つ方略，あるいは範例を見つけだすようなものです。計算論的思考は，膨大なデータを用いて計算を高速化することです。そこでは，時間と空間，処理能力と記憶容量とがトレードオフ関係にあるのです。

次のような日常の例を考えてみることにしましょう：あなたの娘が朝学校へ行くとき，その日に必要なものをバックパックに詰め込むこと；これがプリフェッチとキャッシングです。あなたの息子が手袋をなくしたとき，あなたに促されて来た道を辿り直すこと；これがバックトラッキングです。どの時点でスキーレンタルの利用をやめて自分用スキーを買うのか；これはオンラインアルゴリズムです。スーパーマーケットでどのレジ行列に並ぶのか；これはマルチサーバーシステムにおける性能モデリングです。なぜあなたの電話は停電中でも通ずるのか；これは障害からの分離であり，設計の冗長性です。完全自動公開チューリングテスト（CAPTCHA）は人間であることをどう立証するのか；これはコンピューティングエージェントの裏をかくため，AIにとって解決困難な問題を利用したものです。

計算論的思考が人々の暮らしに根付いたと言えるのは，アルゴリズムや前提条件といった単語がすべての人の語彙の一つとなり，非決定性とガベージ・コレクションがコンピュータ科学者の用いている意味となり，ツリーといえば上下逆さに描かれるようになったときでしょう。

私たちは他の研究領域で計算論的思考が影響を及ぼしているのを目の当たりにしてきました。たとえば，機械学習は統計学を変えました。統計的学習は，数年前には考えられなかったような規模のデータ量と分類次元を有する問題に適用されています。あらゆる組織の統計部門がコンピュータ科学者を採用しています。コンピュータ

科学の学部・研究科は，すでに統計部門が存在しているか，あるいは新しい統計部門を設立することに取り組んでいます。

コンピュータ科学者達の最近の関心が生物学に向いたのは，生物学者にとって計算論的思考が有益であると考えているからです。生物学に対するコンピュータ科学の貢献は，膨大な量のシーケンスデータからパターンを探し出す能力に留まりません。データ構造とアルゴリズム，私たちがいうところの計算論的抽象化と方法論を使ってタンパク質の構造を表現し，その機能を解明することが期待できるのです。計算論的生物学は生物学者の思考法を変えていってます。同様に，計算論的ゲーム理論は経済学者の思考を；ナノコンピューティングは化学者の思考を；量子コンピューティングは物理学者の思考を変えつつあるのです。

こうした思考は，他分野の科学者だけでなく，すべての人々のスキルセットの一部となり得ます。ユビキタスコンピューティングが今日もたらしているものを，計算論的思考は明日もたらします。ユビキタスコンピューティングが今日の現実となった昨日の夢であったのですから，つまり，計算論的思考は明日の現実なのです。

コンピュータ科学者のように考えるとは，
コンピュータをプログラムできること以上に，
複数の抽象的レベルで考えられることを意味します。

何であって，何でないのか

コンピュータ科学では，計算に関して，何が計算できて，それをどう計算するのか，を研究しています。したがって，計算論的思考は次のような特徴を有しています：

⟨概念化であり，プログラミングではない⟩

コンピュータ科学とはすなわちコンピュータプログラミング，ではありません。コンピュータ科学者のように考えることは，コンピュータをプログラムできること以上の意味を持ちます。それは複数の抽象的レベルで考えることを必要としているのです。

⟨基礎的であり，暗唱的技能ではない⟩

基礎的技能とは，現代社会で活動するために人間なら誰もが知っているべきものです。一方，暗唱的とは，機械的なルーチンのことです。皮肉なことがあるとすれば，コンピュータが人間のように考えることを目指すAIグランドチャレンジ課題をコンピュータ科学が解決してしまうと，人間の思考は暗唱的なもの（機械的ルーチン）だと見なさざるを得なくなることです。

⟨人間の思考法であり，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ありません。コンピュータは鈍感で退屈

であり，人間は賢くて創造性が豊かなのです。私たち人間がコンピュータを刺激的なものにするのです。コンピュータ機器を用いることで，コンピュータ時代より前には扱えなかった問題への取り組みや，私たちの想像力だけが機能性の制約だったシステムの構築にも，私たち自身の賢さを使うことができるようになります。

⟨数学的思考と工学的思考が補い合い組み合わさる⟩

コンピュータ科学は，すべての科学がそうであるように，その形式的基礎が数学にあることから，本質的に数学的思考を内包しています。コンピュータ科学は，現実世界と相互作用するシステムを構築していることから，本質的に工学的思考を内包しています。コンピュータ機器がもっている制約は，コンピュータ科学者を数学的にだけでなく，計算論的に思考するよう仕向けます。自由自在に仮想世界を構築できるため，物理世界を超えたシステムの建設が可能なのです。

⟨アイデアであり，モノではない⟩

私たちが生み出すソフトウェアやハードウェアといった成果物は，単に物理的にいろんな場所に存在して生活の中でいつでも触れられるというだけではありません。計算論的な概念として，私たちが問題にアプローチしたり解決したり，日常生活を管理したり，他人とコミュニケートしたり交流するために用いるものなのです。そして，

⟨すべての人に，どこででも⟩

計算論的思考が，実際的なものとなるのは，人間の努力と一体化して，自明な考え方になったときでしょう。

多くの人がコンピュータ科学とコンピュータプログラミングを同じとみなしています。一部の保護者はコンピュータ科学を専攻している子供達の就職機会が狭く限られていると見ています。多くの人はコンピュータ科学における基礎的研究は完了していて，工学的な課題だけが残されていると考えています。計算論的思考は，この分野におけるこうした社会通念を変えようと行動している私たちコンピュータ科学の教育者，研究者，実践者を導いてくれるグランドビジョンなのです。私たちにとって特に必要なのは，大学前段階の先生や保護者や生徒に向けて，２つのメッセージを送り届けることです：

⟨理解し解決すべき知的な挑戦と魅力的な科学的問題が待っている⟩

問題の範囲と解決策の範囲を制限するのは私たち自身の好奇心と創造力だけだということです。そして，

⟨コンピュータ科学を専門にした人は何にでも携われる⟩

英語や数学を専門にした人は多数の異なるキャリアを歩んでいます。コンピュータ科学も同じです。コンピュータ科学を専門にした人は，医学，法律，経営，政治，いかなるタイプの科学や工学，あるいは芸術のキャリアにさえ進むことができるのです。

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」といった科目を大学の新入生向けに，コンピュータ科学専攻だけでなく他専攻の学生たちにも提供して教えるべきです。大学前段階の生徒にも計算論的な方法やモデルに触れさせるべきでしょう。コンピュータ科学への関心が低下していたり，コンピュータ科学への研究予算配分が減っていることを嘆くよりも，世間一般の関心をこの分野の知的冒険へと導くことにこそ力を割くべきです。そうやってコンピュータ科学のもつ楽しさや畏敬，パワーを広げていき，計算論的思考が当たり前のものになることを目指していきましょう。