《講演》

ACM's Celebration of 50 Years of the A. M. Turing Award
Day2 Saturday 24 June 2017

Computer Science as a Major Body of Accumulated Knowledge
知識の集積としての計算機科学

Donald ("Don") Ervin Knuth（1974 Turing laureate）
翻訳：鳥澤健太郎　Julien Kloetzer（情報通信研究機構）　協力：国立情報学研究所

＜講演者の紹介＞

Donald E. Knuth 博士は計算機科学の黎明期より活躍してきたこの分野を代表する研究者であり，まず何より，日本語訳も出版されている "The Art of Computer Programming" という本の著者として知られている．この著書は，数多くのアルゴリズムとその計算量を含む分析をカバーしていて，計算機科学における金字塔となっており，その内容に関して Knuth 博士は 1974 年にチューリング賞を受賞している．また，おそらく，計算機科学者に限らず，多くの研究者は LaTeX というシステムなしでは論文が書けず仕事にならないと思われるが，そのベースとなった組版システムである TeX の開発者としても知られている．ちなみにこの開発のきっかけは先ほどの著書の印刷の品質に不満があったからとのことであるが，だからといってオリジナルの組版システムを自ら作ってしまうというところが黎明期の計算機科学者のある意味とんでもないバイタリティが感じられるところであり，また，現在の IT 社会はそうしたとんでもないバイタリティの発露が多々あって初めて成立していることを実感させられるところである．Knuth 博士は，この TeX の開発のきっかけや，彼の書いたプログラムはバグが異常なほど少ないことなど数多くの有名な逸話に事欠かず，計算機科学における「生きた伝説」と言ってよいであろう．本講演では学術的に新たなことを話しているわけではないが，計算機科学と数学等の他の研究分野との比較，昨今の AI ブームをチクリと刺す皮肉や「計算機科学史」の研究に関する注文などをユーモアに包みつつ述べており，会場はたびたび爆笑の渦に包まれている．若い読者はまだ未経験の人も多いかもしれないが，往々にしてハイレベルな国際会議での招待講演はこのように笑いを取りつつ，含蓄の多い話がなされることが多く，本講演はある意味その典型的な例と言えよう．また，読者の中には Knuth 博士の名前を知らない方もおられるかもしれないが，これをきっかけに彼の著書も眺めてみるとよいかもしれない．なお，本講演の動画は，Web ページ☆1 で視聴することができる．

MC：私が最初に，Donald E. Knuth さんの名前を聞いたのは，私がまだ数学者のときでした．その後，計算機科学に移るとき，サウサンプトン大学の David Barron 教授の個人授業を受けました．Donald は，彼のことをソフトウェア運用，またエディタとして活躍した人物として覚えているそうです．Donald は，David と実際に会ったことはないそうですが，定期的に連絡をとっていたそうです．本日はここに Donald の著書☆2 が置いてありますね．興味深い小道具になりそうです．

☆1 https://www.acm.org/turing-award-50/video
☆2 "The Art of Computer Programming"

Donald は，文芸を愛した人物です．計算機科学者であり，また歴史家，音楽家でもあり，その人生は科学とアートの枠を超え，幅広く活躍してきました．今も，舞台袖で非常に若々しい姿でスタンバイしています．彼が ACM チューリング賞を受賞したのは 1974 年です．ですから，受賞の際はおそらく半ズボンを履いていたでしょう（会場笑）．

受賞したときは，彼は非常に若かったですね．今であれば，チューリング賞受賞の前に ACM Prize を受賞したでしょう．明らかに，私たちの世界では最も重要な人物の 1 人です．それではステージにお迎えいたします．1974 年 ACM チューリング賞受賞者，Donald E. Knuth ！

Knuth：（大きな拍手に対して）分かったよ，もう十分だよ．大げさな拍手は十分なので，それより私の本をぜひ読んでください（笑）．

今朝，主催者から本日の私の講演タイトルを聞いたのですが，正確には覚えていなくて，計算機科学は，偉大な成果を数多く出した，立派な研究分野だと，皆さんに“証明”してほしい，という趣旨だったかと思います．実際，その点を（科学的に）“証明”する必要はないかと思っております．ですから本日は私の経験，エピソードをいくつかお話しし，その後多くの時間を皆さんからの質疑応答に当てたいと思っています．

中央の通路にマイクを 2 本用意していますので，私の話が終わった後に，質問 1 つに 30 秒から 1 分程度でお答えしたいと思っています．質問が 1 つもなかったら，ぜひ休憩時間としてお楽しみください（会場笑）．

1960 年代，数学専攻の学友から「何の研究をしてるんだい？」と聞かれ「計算機科学だ」と答えると，「計算機科学？　なんだそりゃ？」と言われたものでした．当時は，人工知能（AI），数値解析，プログラミング言語の分野で構成されていました．また別な例を挙げますと，プリンストン大学にいた私の同僚が，「計算機科学は，1,000 個ほど“深い”定理（deep theorems）が証明されたら，本当の科学になるよ」と言ったんですよね．“深い”定理とは，一体どういう意味か分かりませんが，おそらく深層学習（deep learning）で発見されるものとは違うものだろうと思います（会場爆笑）．

いずれにしても，当時，私は“深い”成果を 1,000 件も主張できませんでした．それでも“deep：深層化（多層化）”の概念はその頃を発端に，70 年代に引き継がれ，そして現在までの間に，何千件もの研究に使われるようになりました．計算機科学には，数学と同じように，私たちが研究課題を創案できるという特権があります．自然界が決めた方向に進む必要はなく，自分で研究分野を作り出し，独自の言語，公理などをデザインできるのです．一方で，計算機科学が数学の部分集合なのか，数学が計算機科学の部分集合なのか，私は分かりませんが（会場笑），私が以前からずっと信じてきたように，どちらが上位かではなく，並行する 2 つの学問の分野と言えるでしょう．共通点が多くありながら，明確な違いもあります．大学に計算機科学部が設立されて間もない頃，明らかだったのは，数学では決して終身在職権を得られない人たちが，計算機科学では得られたのです（会場笑）．反対もありましたけどね．実際には，反対の方が多かったかもしれません．

いずれにしても学部として発足した頃は，そのような状況でした．学問分野として数学と共通点が多く，しかも計算機科学部では，研究テーマに関して私たちが主導権を持つことができたのです．また，結果が出たときに，それが確実に分かる点もいいですね．たとえば物理学者は，成果に完全な確証が持てません．実際に星や太陽まで行って，実際の数値を測定することはできませんから．推測しかできず，自分がしたことが本当に正しいのか，死ぬまで確実に分かることはありません．ですが数学と計算機科学には，その機会があるのです．

それでは，何か質問がある方は，ぜひマイクの方へ向かってください．お答えしたいと思いま

す．では，質問する方がマイクに向かうまでの間，私が ACM チューリング賞を受賞したときのエピソードをお話ししようと思います．1974 年，あの当時は，ACM チューリング賞受賞で手にする賞金は，今より 100 万ドル少なかったんです（会場爆笑）☆3．ですが，非常に素敵なティファニー製のシルバーのボウルをいただきまして，妻と私はイチゴを食すときにいつも使っています（会場笑）．実際にイチゴの味がずっとおいしくなりましてね（会場笑）．本日は，私が賞を受賞した日に，出版社からいただいたプレゼントをお見せしたいと思います．ここにあるのがそのプレゼントですが，私の著書全 3 巻☆4 をこのように皮のカバーで美しく綴込んだ形でいただきました．実に見事な製本技術で，今でも非常に良い状態です．中の文字を見ると，組版が古い時代のものだと分かりますね☆5．当時は，著書をこのような形でもらえることが，何よりも価値あることだったんです．だから，受け取ったあとすぐに著作権のページを探して，どこの組版を使っているのか調べましたよ（会場笑）．では，質問の準備ができたようなので，質疑応答に移りましょう．

質問者①：少しバカげた質問かもしれませんが，尊敬する教授に質問ができる貴重な機会ですので，ぜひお願いします．教授のアルゴリズムの著書の最初で，演習問題の難易度を設定されましたね．そこで伺いますが，同じ尺度を，あなたの生活の，研究以外の面に応用したことはあるでしょうか？

　たとえば，研究の助成金申請を書くというタスクに応用するとしたら，その難易度は 11111 か，もしくは 00 のタスクどちらと言えるでしょうか（会場笑）．

Knuth 氏：うーん，自分で（自分の著書の）演習問題を見るたびに違う難易度を与えなきゃと思うんだけどね．私の人生では，（研究資金を得るために）ウォールストリート含め，その他お金があるところに対して文書を書くことが多かったですね．ですから，今引退して一番嬉しいのは，もう研究資金の心配をしなくていいということです．実際，私はコンピュータが何のために買われているのか，使われているのか，まったく知りませんでした．私個人の場合はたまたま，使えるコンピュータが周りにあったというだけなんです．経済的な面に関してアドバイスをするには，私はまったくみなさんのお役に立てないと思います．研究資金を得るというのは一種のゲームで，自分がそこそこうまくやったというのは認めますが．では，次の質問に移りましょう．

質問者②：あなたが TeX を発表されたとき，（TeX のバグが少ないことを言うために）大変大胆な賭けをしましたよね．新しいバグが見つかるたびに，前回のバグ発見者にあげた懸賞金の 2 倍の懸賞金を与えるまでになりました．TeX のように便利なシステムで，そんな大胆な賭けで宣伝されたものはほかに聞いたことがないんですが，そんなやり方を一体どうやって思いついたのですか？

Knuth 氏：懸賞金が 327.68 ドルに到達したところで，懸賞金を倍にするのはやめましたけどね（会場笑）．

　とてもありがたかったのは，バグを発見した人たちが自分たちの仕事を中断してまで初期の段階でバグを報告してくれたことです．次にバグの確認をするのは 2021 年になるはずです．ボランティアの人たちがバグの報告を整理してくれたので，私はスワップイン，スワップアウトよりも信用しているバッチ☆6 でデバッグをすることにしました．当初は毎年，のちに 2 ～ 3 年ごとに修正プログラムを出していました．最後に出したのが，2014 年だったと思います．その 7 年後が 2021 年です．そこで私がバグの報告を見るこ

☆3 現在賞金は 100 万ドル．
☆4 1974 年受賞の段階では “The Art of Computer Programming” は全 3 巻だった．
☆5 自身が TeX で新たな組版を開発した．
☆6 プログラムを一気にまとめて実行するバッチ処理という方法にかけて，バグの報告がたまったところでまとめてデバッグをしていることを言っている洒落．

とになると予想しています（会場笑）．懸賞金は326.78 ドルのままですけれどね．

質問者②：本当にお伺いしたいのは，このように非常に複雑なシステムでありながら，賭けをしてしまえるほど純粋な[☆7]ものをどうやってデザインできたのか，という点です．

Knuth 氏：文芸的プログラミングはかなり役立ちましたね[☆8]．また，The Errors of TeX という論文を書き，最初に除去した 1,500 個のバグを調べ，“goto 文は有害である ” などと種類別に分類しました（会場笑）．問題を把握しようと．結果は，その通り goto 文は有害でした．それでも全体の 2% ほどしかありませんでしたけど（会場笑）．その論文はのちにジャーナルで公表されましたが，役立つ経験になりました．論文を探してみてください．

質問者③：ジョークは用意していませんが，２つ質問があります．まず１つ目，コンピュータは優れた作曲家になる日が来ると思いますか？

Knuth 氏：コンピュータは，作曲の優秀な助手になってくれますよね．ですが，アルゴリズムを信頼して作曲を全部させたいと思っている人たちもいますね．でも，私が言いたいのはそういう話ではありません．私が望むのは，コンピュータの手助けは（自分で作曲をすべてする場合よりは）少数の選択肢を見せるところまでで，そこから先は自分で考えて選択をするような方法です．実際，この方法は（私が作曲する場合には）うまくいっています．

質問者③：それに関係して，２つ目の質問です．あなたは人生で膨大な数の定理を証明しましたが，今後あなたの定理の証明をコンピュータが手助けすることはあるでしょうか？

Knuth 氏：イエス．実際，現在すでに手伝ってもらっています（会場爆笑）．自宅のコンピュータで走ってるプログラムが，必要なときに証明のプロセスを手伝ってくれると期待してます．

質問者④：数学と計算機科学の違いについて話したときに，“1,000 ほどの定理 ” とおっしゃいましたが…….

Knuth 氏：私が言いたかったのは，（数学と計算機科学の）違いではないんです．計算機科学がいつか成熟したときに，独立した学科として考慮されるに値する，という意味の話です．

質問者④：明確にしていただいて，ありがとうございます．社会科学者が，物理学の方が研究結果の再現性が高いという理由で，物理学者を妬んでいるという話を聞いたことがあります．その再現性に関して計算機科学の中でも同じように物理学，数学などへの妬みのようなものというのはあるのでしょうか？　また，そうした問題に対してどう対処すべきでしょうか？

Knuth 氏：いい質問ですが，明確な答えはできないかと思います．確かにコンピュータの計算結果の再現性について言えば，私は文芸的プログラミングの提唱者ですので，ぜひインターネット上で文芸的プログラミングについて調べていただければと思います．統計学の領域では，文芸的プログラミングを喜んで受け入れています．論文の主張を検証する際に，プログラムが分かりやすいほうがいいですからね．文芸的プログラミングの概念は，Edsger Wybe Dijkstra，Charles Antony Richard Hoare，Ole-Johan Dahl で知られる構造化プログラムの概念からアイディアを得たものです．そのチューリング賞受賞者の３人で，“ 構造化プログラミング ” という素晴らしい本も出していますしね．その数年後，私は，“ よし，私は文芸的[☆9]プログラミングなるものに取り組むことにしよう ” と心に決め，“ そうしないと，みんなが読み書き不能のプログラミング[☆10]しか書かないと批判されてしまう ” と，この概念を発案しました（会場笑）．

☆7 つまり，バグが少ない．
☆8 文芸的プログラミングは Knuth 博士が提唱したプログラミングのスタイルである．
☆9 literate ＝ 読み書きができる人．
☆10 illiterate programing：literate programing にかけたジョーク．

いずれにしても，文芸的プログラミングのアイディアは，プログラムをそもそも実行してもらうためにコンピュータに提示するものと考えるのではなく，人間に提示してしかも理解してもらうものだと考えるということです．ですから，その点で文芸的プログラミングは，再現性の実現を強く後押しするはずです．いずれにしても行き着くところは，プログラムをどうやって理解できるか，ちゃんと知っておかないといけないということです．でも，そうですね，説明が難しいのですが，私の思考プロセスは，他の分野にいる人のものとはおそらく違うでしょう．私たちの分野の人は，皆同じ風変わりな点があり，そのせいで他の領域をうらやましいと感じている人が多いですよね．その風変わりな点を皆持っているために，お互いを見つけ，こういう場で一緒になり，話をし，仲良くなる．しかし，我々（計算機科学者）がそうした風変わりな考え方とは別の考え方をしているならば，違う分野の人からシステムのデザインに協力を頼まれても，そのデザインをするのに我々は適任ではないと言わなければないでしょう☆11．

質問者⑤：数々の素晴らしい著書を出版してくれたことに感謝しています．将来また新たなものが生まれるかもしれませんね（笑）．私の質問は，計算機科学で最も重要な定理は何だと思いますか？理論的，実践的両方の観点からお願いします．

Knuth 氏：私が最もイヤな質問の 1 つですね（会場笑）．誰にでも苦手な質問はあるでしょう．今の質問は，親御さんに“子供たちの中で誰が一番好きか？”と聞くのと似たような質問ですが，一番好きな定理であれば，おそらく自分のではないと答えるでしょうけど（会場笑）．

一方で，一番好きなアルゴリズムと聞かれたら，それはあるんですね．“Robert Endre Tarjan の強連結成分アルゴリズム”です．非常にゴージャスで，美しく品のあるアルゴリズムで，私が見てきた中では一番ですね．いつか超えるものが出るかもしれませんけれど．短く，深いアルゴリズムで，私はいろんなものに使っています．

質問者⑥：未来を見据えた質問がしたいのですが．

Knuth 氏：ノー！（会場爆笑）次の方！

質問者⑥：いや，気にしないでいきましょう．昨今，AI などの到来で，人類の存在が脅かされる可能性があると話題になっています．人によって意見はさまざまだと思いますが，あなたの見解はいかがでしょうか？

Knuth 氏：その点については，楽観的，悲観的，両方の見方ができますね．昨日も Stuart Russel が，現在ほとんどの時間をその点に費やしていると言っていました．彼は，誰よりも思慮深い考えをする人物ですが，怖いと感じるのは，彼の主張を見てみると，人間は合理的な存在だという仮説に基いているのです（会場笑）．

しかもその後に，選挙結果☆12 を見たりすると……（会場爆笑）．これは真剣に言っているんですよ．“機械（コンピュータ）に乗っ取られないようにするための手段”というようなシナリオの数々を読んでみると，人間が知性的な判断をするという仮定に基づいて論じられているのです．ですから，我々のように考えていない☆13 人々をコントロールするために，まずはそういう人たちを探し，その人たちと協力するのがいいのではないでしょうか．ロボットのコントロールより，彼らのコントロールを先にした方がいいでしょう．

質問者⑦：将来の計算機科学者のために質問したいと思います．今持っているすべての知識を基に，もし再びキャリアを一からやり直すとしたら，どの部分をこれまでと違う形でやり直しますか？

Knuth 氏：計算機の中で 2 進法ではなく 10 進法を使うと思います（会場爆笑）．

☆11 つまり，我々計算機科学者は風変わりな考え方をするので，システムのデザインができている，と言いたいものと思われる．

☆12 おそらく大統領選のことを言っているものと思われる．

☆13 つまり，合理的でない．

質問者⑧：あなたは，この分野の歴史に関しても素晴らしい研究業績をあげてこられました．ですが，あなたは最近の科学史家たちが我々の分野について書く方法について懸念を持っておられます．お訊きしたいのは，我々の分野の歴史を分析して文献に残すとしたら，何をトピックにするのがよいでしょうか？　将来の世代である大学院の学生で計算機科学の歴史を専攻したい人にはどのようなガイダンスを与えるでしょうか？

Knuth 氏：それに関して今まで私が話したことのほとんどは，インターネット上で公開している動画，“計算機科学史を簡素化して伝えるのはやめよう [Let's not dumb down the history of computer science]” で見ることができます．これは，Tom Kailath 教授に敬意を表するためにスタンフォード大で年に一度開催しているプログラム☆14 にて私が行ったレクチャーです．まずは，計算機科学を研究している人たちにとって，なぜ計算機科学史が大事なのか，という点から入っています．その理由は，これまでの先輩たちが，どこから，どのようにそのアイディアを思いついたのかを学んでおくと，自分たちが進める研究の創造力の点で役立つのです．しかし，計算機科学史の研究トレンドは，目覚ましく変化してきました．60 年代，計算機科学史に関する論文は，アルゴリズムやプログラミングが中心でした．それが今，計算機科学史に関する論文の内容は，“このような形で誰かが研究資金を得た”，“こういうやり方で誰かがどこかとパートナーシップを締結することになった”，“こう言った具合に誰かが両親との間に問題を抱えていたらしい” といった感じです．計算機科学者でない人たちが理解しないような話には何ひとつ言及しないと．

科学史の分野全体で，研究対象が（科学の内容，つまり，科学的な理論，知見，実践等の歴史を主として扱う）内在史と呼ばれるものから（科学的研究を行う業界の栄枯盛衰から研究資金の調達や科学的研究を推進あるいは阻害する政治的状況までを含め，科学を取り巻く社会的状況全般の歴史も扱う）外在史と呼ばれるものへと変わってきたからです．内在史が軽視されるようになったのです．その理由の 1 つは，現在，科学史家がジャーナルに論文を掲載するには，外在史を題材にしないといけないという点です．実際に，ENIAC の歴史についての素晴らしい本があるのですが，著者は，その内容に関する論文は，科学史のジャーナルには掲載してもらえなかったと話していました．また，大学も落ちぶれてしまい，いま米国には，計算機科学史を教えて次世代を育てようとする計算機科学部／科を支援する大学が一校もありません．だから私は，スタンフォード大で取り入れてもらおうと説得しているところです．

最近私は，まだ発表前のある博士論文を読んだのですが，Edsger Wybe Dijkstra の ALGOL 60 コンパイラを研究した人物です．ご存知のように，この ALGOL 60 コンパイラは，アセンブラではなく機械語と八進法で書かれていますが，彼は，非常に入念に分析，考察を行い，そこに見られるイノベーション，その構造，すべての問題，構成についても述べています．こういう論文こそ，計算機科学史と呼ぶにふさわしいと思いましたね．

質問者⑨：ご気分はいかがですか？　ステージ上で☆15 立派にやってるじゃないですか．うまいこと臨機応変にお話しされてます（笑）．さて，あなたも懸念されていると思いますが，私は計算機科学の教育に対する大学側のフォーカスの欠如を危惧しています．あなたが達成した成果の多くは私も大変好きなのですが，しかし何よりも心に強く残るのは，その成果にいたるまでの道のり，つまり強靭な集中力なんです．あなたは常に研究に集中しているので，私たちはなかなかあなたにお会いできません．1 つが終わってもすぐに次の巻，次の定理，次のアルゴリズムへ進んでいきます．あなたの計算機科学に対する心構えを，この

☆14 2014 Kailath Lecture.

☆15 おそらく年齢の割に．

場にもいる若い世代にどうお伝えするか，ぜひお聞きしたいです．

Knuth 氏：私は，次の 2 つの仮説のどちらが正しいのかで悩んでいます．1 つ目の仮説は，どんな人でも，計算機科学を教えてやれば，計算機科学者になれるというもの．もう 1 つの仮説は，世界の人口のうち，わずか 2% がギークになるべくして生まれてきていて，98% はそうではないというもの[☆16]．どちらの仮説が正しいのか，まだ自分の答えは出ていません．しかし，2 つ目の仮説が正しいと仮定しましょう．ところが，ギークは自分がギークであることを判断できない．一方で，誰かのパーソナリティを特定しようとする人は，自分と同じタイプのパーソナリティしかうまく特定できない．つまり，誰かがギークであることを証明するのは難しい．それでも，2% の方を正しいと仮定してみましょう．すると，世界の人口の 2% だけしか，計算機科学を教えることができません．しかしそうなれば，教える内容を理解している人が教えていることになります．

最近の『Communications（of the ACM)』でも記事にありましたが，“ 計算的思考：computational thinking” を教えることになるけれど，計算的思考とはなんだろう？という内容です．おそらく現場の先生（教授）たちも，“ 私自身，計算的思考でどう物事を考えるのか分からないのに，どうやって教えればいいんだい？ ” と戸惑っているかもしれません．そこが問題の本質で，もっと力を入れて議論すべきです．現段階ではこれ以上答えられないですね．さて，私はあと 1 つジョークを用意してきたので，いい質問を期待したいですね（会場爆笑）．

質問者⑨：そのあなたのジョークで答えられる一般的な質問になるとよいのですが．ACM の広範なコミュニティにも役に立つ質問にはなると思います．あなたは，計算機科学の数学的な研究における権威で，アルゴリズム解析という分野を作り出しました．また，あなたの著書は，“The Art of Computer Programing” というタイトルですが，計算機科学は基本的に大学の工学部の中に属しています．さらに，この分野は従来の枠を超えて経験的手法に頼ることが増え，データサイエンス，機械学習などさまざまなものを使って，我々の周辺の世界や人々の反応からさまざまなことを学んでいます．あなたは計算機科学を数学と比較していましたが，（数学とは異なって）世界に関してデータから学ぶということもありますし，それ以外の自然科学の分野と比較すべきではないでしょうか？　また，我々計算機科学者は，抽象化，デザインやその他多くのツールを使って非常に複雑な体系を作っているのですから，工学とも比較すべきではないでしょうか？

Knuth 氏：私がチューリング賞受賞講演を行った際，“The Art of Computer Programing” のタイトルの意味を説明したのですが，それは，美しいというだけでなく，自然界にないもののアートという意味なのです．質問の趣旨は，私は計算機科学を数学と常に比較しているが，他の分野とは比較していないという点ですが，実は，その答えが，私が用意したジョークにつながるんです（会場笑）．

本日はこの場の全員が，『Communications of the ACM』の最新版を受け取ると思いますから，ぜひ Peter J. Denning の記事を読んでみてください．とてもよく書かれた記事で，タイトルも素敵で，“Deep Earning[☆17]：深層（深い）金儲け ” です（会場爆笑）．

☆16 つまり，2% しか計算機科学者にはなれないということ．

☆17 タイトルは ”Deep Earning” ではないが，おそらく，以下の記事で ”deep earning” という表現が使われていることを指しているものと思われる．https://cacm.acm.org/magazines/2017/6/217742-remaining-trouble-spots-with-computational-thinking/fulltext