

Computational Thinking に関する言説の動向

A Review of the State of Computational Thinking Discourse

林 向達
Kotatsu Rin

徳島文理大学 人間生活学部
Faculty of Human Life Science, Tokushima Bunri University

＜あらまし＞ 初等教育段階へのプログラミング教育導入に端を発して日本では「プログラミング的思考」という言葉が使われ始めた。この言葉が用いられる際に踏まえられたとされる「Computational Thinking」は、諸外国のコンピュータ科学教育あるいはプログラミング教育推進の際の重要な鍵概念とされるものの、その定義は各様で、様々議論されてきたところである。本発表では、Computational Thinking への注目が生じた経緯からその後の言説の動向について先行文献を調査し、概説を試みた。

＜キーワード＞ コンピュータ・シンキング 計算論的思考
プログラミング的思考 コンピュータ科学

1. はじめに

平成 29 年度小学校学習指導要領総則では、各教科の指導に当たって配慮する事項として、情報活用能力の育成に関わって「プログラミングを体験しながら論理的思考力を身に付ける」という文言が盛り込まれた。

学習指導要領改訂に先立ち設置された「小学校段階における論理的思考力や創造性、問題解決能力等の育成とプログラミング教育に関する有識者会議」は、「小学校段階におけるプログラミング教育の在り方について（議論のまとめ）」（文部科学省 2016）で、プログラミング教育を「時代を超えて普遍的に求められる『プログラミング的思考』などを育むこと」と示した。小学校学習指導要領解説は、これを根拠として小学校段階のプログラミングの取り組みを解説している。

プログラミング的思考は「自分が意図する一連の活動を実現するために、どのような動きの組合せが必要であり、一つ一つの動きに対応した記号を、どのように組み合わせたらいいのか、記号の組合せをどのように改善していけば、より意図した活動に近づくのか、といったことを論理的に考えていく力」と定義され、「いわゆる『コンピュータ・シンキング』の考え方を踏まえつつ、プログラミングと論理的思考との関係を整理しながら提言された定義である」と補足された。

しかし、有識者会議においてコンピュータ・シンキング（Computational Thinking: 以下 CT）に対する詳細な検討が行なわれた開示記録はなく、プログラミングの思考という用語がどのような踏まえによって導出されたのかは明示されていない。

太田（2016）は諸外国のプログラミング教育カリキュラムの比較を介して、坂東ほか（2017）は我が国の情報教育の史的展開におけるプログラミング教育の位置付けの整理を介して、CT の概念や考え方の言説を検討した。阪東らは CT に基づく「プログラミング的思考はプログラミング（コーディング）に限定されており、矮小化されてとらえられているもの」と解し、プログラミング的思考における CT の考え方の踏まえ方が、発展的あるいは深まりを得る形とは言い難いことを示唆した。

太田や阪東らの CT 概念・考え方の検討以外に、小田ほか（2018）はプログラミング教育関連の研究文献を調査分析するにあたって「computational thinking」を検索語として傾向を整理したが、2006 年に公表された Wing（2006）以降の CT 等に関する言説の状況は、日本国内での検討作業が十分とはいえない結果を示していた。

本発表は、CT 言説の動向を調査し、議論を見通す補助線の提供を試みる。

2. 研究目標

Computational Thinking に関する海外の主要な研究文献を調査し、CT 概念が登場した背景と CT に関する議論の動向について概観することを目標とする。

3. 調査研究方法

米国計算機学会 (ACM) の ACM Digital Library および Google Scholar の検索機能を利用して、「Computational Thinking」の語と関連性の高いものとして出力された文献を順次参照しながら、参照件数の多い論文および言説動向を記してある論文を対象として探索的に抽出し、CT 定義に関する記述について調べた。

4. 関連文献調査結果

4.1. Computational Thinking の生起

Computational Thinking という用語は、Papert によって最初に示されたと指摘する文献が多く、Papert (1980) で用語が初出し (Denning 2017, Korkmaz ほか 2017, Corradini ら 2017, 阪東ほか 2017)、同書で提起された「Procedural Thinking」と関連性があると指摘するもの (NRC 2010, Czerkawski ほか 2015, Atmatzidou & Demetriadis 2015, Tedre & Denning 2016, Hacker 2018) もあれば、Papert (1996) で問題解決の新しい方法として初めて言及されたと指摘するもの (Weinberg 2013, Shodiev 2015, Kalelioglu ほか 2016) もあった。そもそも、CT という言葉が最も早くに提示されたのは 1962 年で、プログラミング課程について紹介した Alan

Perlis による記述との指摘 (Czerkawski ほか 2015) もある。

CT 概念に Papert らが少なくない貢献をしていることは確実であるが、今日議論されている CT は Wing (2006) において注目されたものであることに異論を唱える言説は見当たらない。

4.2. CT 言説の構図

今日展開している CT 言説の多くは Wing (2006) に端を発している。一方で、これら言説に関わるムーブメントに対して批判的な立場をとる主張も存在する。Tedre & Denning (2016) は、Wing (2006) とそれに触発された CT 言説および CS 喧伝の動きを「新しい CT ムーブメント」(New CT movement) と称した。同時に、それらが従来の CT (Traditional CT) 言説の歴史や蓄積を踏まえていない点について批判的見解を示し、過度な CT 言説によって失敗の歴史を繰り返さぬよう警鐘を鳴らしている。

4.2.1. 新しい CT ムーブメント

Wing (2006) の 10 年後に公表されたブログ記事 (Wing 2016) で、CT 概念提唱当時の述懐と過去 10 年の成果に関する Wing 自身の見解が記された。

2005 年当時のコンピュータ科学 (CS) を取り巻く状況は決して芳しいものではなく、大学学部存続もが危ぶまれる空気であった。そうした中で Wing 本人は CS の明るい未来を見通していた。社会のあらゆる場でコンピューティングの概念と方法が活用され、こ

表 1 Wing による Computational Thinking 定義記述の変遷

| | |
|--------------------------|--|
| Wing (2006 2008) | Computational thinking is taking an approach to solving problems, designing systems and understanding human behaviour that draws on concepts fundamental to computing. |
| CunySnyderWing (2010) | Computational thinking is the thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by an information-processing agent. |
| Wing (2014) | Computational thinking is the thought processes involved in formulating a problem and expressing its solution(s) in such a way that a computer – human or machine – can effectively carry out. |

の力を利用できる人が有利となる。将来世代に対してコンピュータ科学者の思考の仕方を伝える良い機会であると Wing は考え、CT 概念を主張する論稿が書かれたと述懐する。

Wing 自身の述懐から分かるように Wing (2006) は、CS のプロモーション (CS for All) 的な色彩の強い論稿であった。歴史的には、インターネット Web2.0 の黎明期であり、スマートフォン登場以前という時代状況である。CT は、CS に対する世間的な誤解 (CS とはコンピュータプログラミングであり、CS を専攻しても職業選択の幅が狭い等の認識) を変えようと努力している CS 教育者や研究者、実践者のグランドビジョンとなるべく提起された概念であると主張された。そのためか Wing (2006) の CT に関する記述は、定義というよりも様々なコンピューティング概念を回覧したに近く、「CT は何であって何でないのか」を列記した外延指定をするに留められていた。Wing (2006) で CT 概念の端的な定義記述は行なわれなかったが、その後の定義はフィードバックを受けながら修正されていった (表 1)。Wing (2017) 講演で、Wing (2014) 定義記述が現時点で最適であると述べている。

翻訳者である中島 (2015) は、日本において Wing (2006) は好意的に受け止められた旨を記している。しかし、邦訳公表までの時間差や CT の邦文関連文献の数を考えると、当時の日本に新しい CTムーブメントが及んでいたとはいえない。

4.2.2. 歴史的蓄積の重視

Özçınar (2017) の文献計量学的な分析によれば、Wing (2006) などと並び、CT 関連文献の中心的位置にあるのが Denning (2003) であった。

Denning は、ACM 会長の経歴をもち、Denning (1989) の論文主査として計算機分野のカリキュラム案を作成した。当時の計算機分野を 9 つの副領域として取り出し、各副領域を 3 つの基本プロセス (理論、抽象化、設計) で定義したマトリックスは「デニング図」と呼ばれている。CS の学問的存在意義に関する問いに携わり続ける (Denning 2005 ほか)

か) 立場から、新しい CT ムーブメントに対して発言を繰り返してきた。

Denning (2009) では、新しい CT ムーブメントが、主に 4 つの関心事のもとで動いていると指摘している。

- ・ 科学の場で CS が力を発揮すること (プログラマーとしてではなくパートナーとして)
- ・ CS がもっと魅力的になり、学生には専攻対象として、他分野の科学者には協同対象としてもらえるよう模索すること
- ・ 取り組んでいる研究をその分野の深い問いに立ち返らせること
- ・ コンピューテーションが基礎的なもので、しばしば不可避なものだと、ほとんどの取り組みで示すこと、つまり布教願望である

しかし、こうした新しい CT では他分野への CS 売り込みはうまくいかず、魅力的に見てもらうことも出来ないだろうと懸念する。かつてコンピューショナル科学とコンピュータ科学が注意深く棲み分けを行なおうとしていた歴史を踏まえぬまま、あたかも CS 独特な特徴として CT を推している。1960 から 70 年代に広げてしまった「CS = programming」という認識が今日の誤解に繋がっているが、CT は「CS = computational thinking」への衣替えを安易に想起させるだけではないのか、我々はそれを望んでいるのか。以上のように Denning (2009) で論じている。

さらに Tedre & Denning (2016) では、従来の CT における歴史等をふまえないければ、次の 7 つのリスクに襲われてしまうと警鐘を鳴らす。

- 1) 大志が失われること (Lack of Ambition)
- 2) 独断的になること (Dogmatism)
- 3) 知識と実践の対立 (Knowing Versus Doing)
- 4) 誇張された主張 (Exaggerated Claims)
- 5) コンピューティングに対する狭い理解 (Narrow Views of Computing)
- 6) 定式化の過度な強調 (Overemphasis on Formulation)
- 7) 計算モデルの考え方を失うこと (Losing Sight of Computational Models)

大意を以下要約する。(1)歴史的な洞察なく「車輪の再発明」をしたところで、CT への取り組みは 1980 年代における先達の資産 (その大志) を希釈したものになってしまう。

表 2 関連文献における CT 概念等の構成要素 [Bocconi ほか(2016) p17 を改変し作表]

| Barr & Stephenson, 2011 | ISTE & CSTA, 2011 | Grover & Pea, 2013 | Selby & Woollard, 2013 | CAS Barefoot, 2014 (注 1) | CAS, 2015 | ISTE Standard, 2016 (注 2) | Angeli et al., 2016 | Grover & Pea, 2018 |
|-------------------------|--|---|------------------------|--------------------------|---------------------------|-------------------------------|---|--|
| Abstraction | Abstractions | Abstractions and pattern generalizations | Abstraction | Abstraction | Abstraction | Abstraction | Abstraction | Abstraction and generalization |
| Algorithms & procedures | Algorithmic thinking (a series of ordered steps) | Algorithmic notions of flow of control | Algorithmic thinking | Algorithms | Algorithmic thinking | Algorithms | Algorithms (including Sequencing and Flow of control) | Algorithms and algorithmic thinking |
| Automation | Automating solutions | | | | | | Automation | Automation |
| Data analysis | Analyzing | | | | | Analyze data | | |
| Problem Decomposition | Formulating problems | Structured problem decomposition (modularizing) | Decomposition | Decomposition | Decomposition | Formulate problem definitions | Decomposition | Problem decomposition |
| | | Debugging and systematic error detection | | Debugging | | Refine prototypes | Debugging | Testing and debugging |
| | | Efficiency and performance constraints | Evaluation | Evaluation | Evaluation | Test prototypes | | Evaluation |
| | Generalizing | | Generalizations | | Generalisation (Patterns) | | Generalization | |
| | The ability to deal with open ended problems | Iterative, recursive, and parallel thinking | | Tinkering | | Work with open-ended problems | | Iterative refinement (incremental development) |
| Parallelization | | | | | | | | |
| Simulation | Models and Simulations | | | | | Simulation | | |
| Data representation | | Symbol systems and representations | | | | Represent data | | |
| | | Systematic processing of information | | | | | | |
| | Logically organizing | | | Logic | | | | Logic and logical thinking |
| | | | | Patterns | | | | Patterns and pattern recognition |
| | | | | Creating | | | | Creating computational artefacts |
| | The ability to communicate and work with others | | | Collaborating | | | | Collaboration & Creativity (part of broader twenty-first century skills) |
| | Persistence | | | Persevering | | Perseverance | | |
| | Tolerance for ambiguity | | | | | Ambiguity | | |
| | Confidence in dealing with complexity | | | | | | | |
| | Transferring | | | | | | | |
| | Identifying | | | | | Identify data | | |
| | Implementing | | | | | | | |
| Data collection | | | | | | Collect data | | |

注 1: Computational Thinker の概念とアプローチから 注 2: ISTE Standards for Students の Innovative Designer と Computational Thinker から

(2)あたかも CT が最善の思考法であるかのよう
に考えるべきではなく、逆に視野を狭めて
独断的になることを避けなければならない。
(3)知識を得ることと実践できることの違い
について、概念フレームワーク上で十分吟味
されているかどうか、評価をする際に気をつ
けなければならない。(4)いくつもの知見で転
移は確かめられていないにも関わらず、CT
のもとで問題解決能力が他領域の問題にも
適用できると主張することは誇張となりう
る。(5)昨今の様々なプログラミング推進活動
はともすれば、「コーディング」という専門用
語に対する混乱を生み、CS をプログラミング
という古い枠組みに押し込め、CT に同じ
轍を踏ませてしまう罠を潜ませている。(6)CT
定義で頻繁に登場する定式化 (Formulation)
という言葉は不明瞭なまま使われることが多
い。(7)CS 各領域には固有の計算モデルが存
在するのであって、CT とは単に既存の計算
モデルを制御することだけでなく、新しい計
算モデルを設計することでもある。何を計算
しようとするかを理解するためにも、CT と
機械動作 (理論的あるいは物理的なもの) と
の強い関係性について意識すべきである。

Denning (2017) では、Aho (2012) 定義に
目を向けるよう提案した。Aho は「計算モデ
ルとよばれる抽象化は、コンピューテーション
と CT の核心である。コンピューテーションと
は、計算モデルの基礎的観点から定義される
プロセスであり、CT とは解決策が計算ス
テップとアルゴリズムとして表現可能な問題
の定式化を伴う思考プロセスである。」と計
算モデルに触れているが、一方、2006 年以降
の新しい CT に計算モデルへの言及がないこ
とを誤りだと Denning は評している。

4.3. 新しい CT の様々な定義言説

様々な批判的言説も議論の中へ含み込んで
いく形で、新しい CTムーブメントは各国の
情報教育・CS 教育に影響を与えていった。
Haseski ら(2018)は、CT 関連文献 (2016 年以
前のもの) を検索し、CT 定義を含んだ論文数
を集計したところ、59 タイプの定義が見出さ
れたとした。こうした状況が CT 概念の理解
を難しくしていると考察している。

表 2 は、Bocconi ほか (2016) が CT 概念と
スキルについて整理したものを改変して、主
な文献と教育カリキュラムで示された CT 概
念等の構成要素を追加したものである。

米国の International Society for
Technology in Education (ISTA) と the
Computer Science Teachers Association
(CSTA) は、2009 年夏から K-12 のための CT
に関する操作的定義をつくるため、CT の概
念と能力の整理に取り組み (Barr &
Stephenson 2011)、その成果が ISTE &
CSTA (2011) として公表された。CSTA
K-12 Computer Science Standards Revised
2011 は、この定義を踏まえて作成された。な
お、CSTA K-12 Computer Science Standards
Revised 2017 では、CT の語が直接フレーム
ワークに使われなくなった。

NRC (2010)で報告されたワークショップ
参加者だった Pea は、Grover & Pea (2013)
で K-12 における CT 言説をレビューし、CT に
含む要素として広く受け入れられているもの
を挙げた。Grover & Pea (2018) では英国
CAS の定義を踏まえて見直しをしている。

英国の CAS による定義は、CAS Barefoot
(2014) と CAS (2015) の 2 つあるように見
受けられる。学習教材リソースの提供が目的
である CAS Barefoot (2014) では
Computational Thinker を支える 6 つの概念
と 5 つのアプローチを示す形で CT 概念を示
した。CAS (2015) では 5 つの要素を含む論
理的思考であると CT を解説しているが、同
じ文献内に両者が示されているにも関わらず
厳密な対応づけがないため多少の混乱を生ん
でいる。

Angeli ほか (2016) は K-6 の CT カリキュ
ラムフレームワークを作成するために CT 言
説をレビューした結果を報告した。

各所の定義を一つの表にまとめた結果、表
2 は、概念要素に留まらず、能力や行為レベ
ルの要素も含み込んだものとなっている。

5. まとめ

本発表は「コンピューティショナル・シンキン
グ」(Computational Thinking) に関して、そ
の言説動向を先行文献を通して概観した。

Wing (2006) 以降の CT ムーブメントは, Denning を始めとした批判的検討の要請に応えつつ, 文脈に応じた議論へと歩を進めているように見受けられる. たとえば CSTA K-12 Computer Science Standards Revised 2017 では, CT に関して Lee(2016) の定義を掲げた上で, 「CS は CT を発達させるユニークな機会を提供するものの, CT を適用する機会 は CS の文脈を超えている」という認識を QA で示している. フレームワークでは, CT を CT 実践(CT Practices)の形で反映させたことも, CT 議論の混乱から抜け出る前進の形であろう.

しかし, 海外で展開してきた CT 言説は, 日本に紹介される機会がほとんど無かっただけでなく, 新しい CT ムーブメントはもちろん, 従来の CT に対する歴史的な理解についても乏しいのが日本の現状ではなかろうか. 海外の CT 言説がさらに前進しつつある中で, CT に対する理解や議論水準が十分でない日本で示された「プログラミング的思考」は, 果たしてどれだけ CT を踏まえたといえるのか. 疑問の余地は少なくない. このままプログラミング教育の文脈だけが取り上げられて議論が進めば, Denning が懸念した畏に陥ることは明白である.

今後は, 日本の情報教育, 情報処理教育, プログラミング体験・学習等が, 体系的な教育内容として学校教育で扱われるための議論がより必要になってくる. そのためにも, 世界のコンピュータ科学に対する基礎的知識を学ぶ機会を増やし, 加えて, 日本の情報処理分野の歴史についても参照しやすくすることで議論の土壌を作っていく必要があるだろう. さらに実践や評価における CT の議論についても視野を広げて知見を整理したい.

参考文献

- Aho, A. V. (2012) Computation and Computational Thinking. *The Computer Journal*, 55(7): 832-835.
- Allan, W. & Coulter, B. & Denner, J. & Erickson, J. & Lee, I. & Malyn-Smith, J. & Martin, F. (2010) Computational Thinking for Youth. ITEST Small Group on Computational Thinking. <http://stelar.edc.org/publications/computational-thinking-youth/> (accessed 2018.04.23)
- Angeli, C. & Voogt, J. & Fluck, A. & Webb, M. & Cox, M. & Malyn-Smith, J. & Zagami, J. (2016) A K-6 Computational Thinking Curriculum Framework: Implications for Teacher Knowledge. *Educational Technology & Society*, 19 (3): 47-57.
- Atmatzidou, S. & Demetriadis, S. (2015) Advancing students' computational thinking skills through educational robotics: A study on age and gender relevant differences. *Robotics and Autonomous Systems*, 75B: 661-670.
- 阪東哲也, 黒田昌克, 福井昌則, 森山潤 (2017) 我が国の初等中等教育におけるプログラミング教育の制度化に関する批判的検討. *学校教育学研究*, 30: 173-184.
- Barr, V., & Stephenson, C. (2011) Bringing computational thinking to K-12: What is involved and what is the role of the computer science education community? *ACM Inroads*, 2(1): 48-54.
- Bocconi, S. & Chiocciariello, A. & Dettori, G. & Ferrari, A. & Engelhardt, K. (2016) Developing computational thinking in compulsory education - Implications for policy and practice. European Commission, Joint Research Centre (JRC). <http://publications.jrc.ec.europa.eu/repository/handle/JRC104188> (accessed 2018.04.23)
- CAS Barefoot (2014) Computational Thinking. <https://barefootcas.org.uk/barefoot-primary-computing-resources/concepts/computational-thinking/> (accessed 2018.04.23)
- Computer Science Teachers Association (2011) CSTA K-12 Computer Science Standards Revised 2011. <http://c.ymcdn.com/sites/www.csteachers>.

- org/resource/resmgr/Docs/Standards/CS
TA_K-12_CSS.pdf (accessed 2018.05.06)
- Computer Science Teachers Association
(2017) CSTA K-12 Computer Science
Standards, Revised 2017.
<https://drive.google.com/file/d/0B0TlX1G3mywqbXpydGdIVk00Y1U/>
(accessed 2018.05.06)
- Corradini, I. & Lodi, M. & Nardelli, E. (2017)
Conceptions and Misconceptions about
Computational Thinking among Italian
Primary School Teachers. In
Proceedings of the ICER '17, August,
2017, pp.136-144.
- Csizmadia, A. & Curzon, P. & Dorling, M. &
Humphreys, S. & Thomas Ng, T. &
Selby, C. & Woollard, J. (2015)
Computational thinking - A Guide for
teachers. Computing At School (CAS
2015), UK.
<https://community.computingschool.org.uk/resources/2324/>
(accessed 2018.04.23)
- Czerkawski, B. C. & Lyman, E. W. (2015)
Exploring issues about computational
thinking in higher education.
TechTrends, 59(2): 57-65.
- Denning, P. (Chairman) & Comer, D. E. &
Gries, D. & Mulder, M. C. & Tucker, A.
& Turner, A. J. & Young, P. R. (1989)
Computing as a Discipline. *Commun. ACM*, 32(1): 9-23.
- Denning, P. J. (2003) Great principles of
computing. *Commun. ACM*, 46(11): 15-20.
- Denning, P. J. (2005) Is computer science
science? *Commun. ACM*, 48(4): 27-31.
- Denning, P. J. (2009) The profession of IT:
Beyond computational thinking.
Commun. ACM, 52(6): 28-30.
- Denning, P. J. (2017) Remaining trouble
spots with computational thinking.
Commun. ACM, 60(6): 33-39.
- Grover, S. & Pea, R. (2013) Computational
thinking in K-12: A review of the state
of the field. *Educational Researcher*, 42
(1): 38-43.
- Grover, S. & Pea, R. (2018) Computational
Thinking: A Competency Whose Time
Has Come. *Computer Science Education: Perspectives on Teaching and Learning in School*, pp.19-38.
- Guzdial, M. (2015) Learner-Centered Design
of Computing Education: Research on
Computing for Everyone. *Synthesis Lectures on Human-Centered Informatics*, 8(6):1-165.
- Hacker, M. (2018) Integrating
Computational Thinking into
Technology and Engineering Education.
Technology and Engineering Teacher,
77: 8-14
- Haseski, H. & Ilic, U. & Tugtekin, U. (2018)
Defining a New 21st Century Skill-
Computational Thinking: Concepts and
Trends. *International Education Studies*, 11(4): 29-42.
- International Society for Technology in
Education & Computer Science Teachers
Association (2011) Operational
Definition of Computational Thinking
for K-12 Education. <http://www.iste.org/docs/ct-documents/computational-thinking-operational-definition-flyer.pdf>
(accessed 2018.04.23)
- International Society for Technology in
Education. ISTE Standards for
Students, 2016;
<http://www.iste.org/standards/for-students> (accessed 2018.05.06)
- Korkmaz, Ö. & Çakir, R. & Ozden, M. (2017)
A validity and reliability study of the
Computational Thinking Scales (CTS).
Computers in Human Behavior, 72:
558-569.
- Lee, I. & Martin, F. & Denner, J. & Coulter,
B. & Allan, W. & Erickson, J. & Malyn-Smith, J. & Werner, L. (2011)
Computational thinking for youth in
practice. *ACM Inroads*, 2(1): 32-37.
- Lee, I. (2016) Reclaiming the roots of CT.
CSTA Voice: The Voice of K-12 Computer Science Education and Its Educators, 12(1): 3-4.
http://www.csteachers.org/resource/resmgr/Voice/csta_voice_03_2016.pdf
(accessed 2018.05.06)
- National Research Council (2010) Report of a

- workshop on the scope and nature of computational thinking. The National Academies Press, Washington, DC
- National Research Council (2011) Report of a workshop of pedagogical aspects of computational thinking. The National Academies Press, Washington, DC
- 文部科学省 (2016) 小学校段階におけるプログラミング教育の在り方について (議論の取りまとめ). 小学校段階における論理的思考力や創造性、問題解決能力等の育成とプログラミング教育に関する有識者会議, 2016 年 6 月 16 日.
http://www.mext.go.jp/b_menu/shingi/chousa/shotou/122/attach/1372525.htm (参照日 2018.04.23)
- 小田理代, 堀田達也 (2018) 海外と日本における初等中等教育段階のプログラミング教育の研究に関する傾向の分析, 日本教育工学研究報告集, 18(1): 41-46.
- 太田剛, 森本容介, 加藤浩 (2016) 諸外国のプログラミング教育を含む情報教育カリキュラムに関する調査, 日本教育工学会論文誌, 40(3): 197-208.
- Özçınar, H. (2017) Bibliometric analysis of computational thinking research. *Educational technology theory and practice*, Turkey, 7(2): 149-171.
- Papert, S. (1980) *Mindstorms: Children, computers, and powerful ideas*. New York, NY: Basic Books.
- Papert, S. (1996) An exploration in the space of mathematics education. *International Journal of Computers for Mathematical Learning*, 1(1): 95-123.
- Selby, C. & Woollard, J. (2013) Computational thinking: the developing definition. University of Southampton (E-prints).
<https://eprints.soton.ac.uk/356481/> (accessed 2018.04.23)
- Tedre, M. & Denning, P. J. (2016) The Long Quest for Computational Thinking. In *Proceedings of the 16th Koli Calling Conference on Computing Education Research 2016*, pp.120–129.
- Wing, J. M. (2006) Computational Thinking. *Commun. ACM*, 49(3): 33–35.
- (邦訳 中島秀之 (2015) 計算論的思考. 情報処理, 56(6): 584-587.)
- Wing, J. M. (2008) Computational thinking and thinking about computing. *Philosophical Transactions of The Royal Society A*, 366: 3717–3725.
- Wing, J. M. (2010) Research Notebook: Computational thinking: What and Why? *Link Magazine*.
<https://www.cs.cmu.edu/link/research-notebook-computational-thinking-what-and-why/> (accessed 2018.04.23)
- Wing, J. M. (2014) Computational thinking benefits society. *40th Anniversary Blog of Social Issues in Computing*.
<http://socialissues.cs.toronto.edu/2014/01/computational-thinking/> (accessed 2018.04.23)
- Wing, J. M. (2016) Computational thinking, 10 years later. *Microsoft Research Blog*.
<https://www.microsoft.com/en-us/research/blog/computational-thinking-10-years-later/> (accessed 2018.04.23)
- Wing, J. M. (2017) Opening keynote: Computational Thinking. The Trippel Helix Conference on Computational Thinking and Digital Competences in Primary and Secondary Education, Stockholm, September, 2017.
https://www.youtube.com/watch?v=f_cOtBzi2Oo&t (accessed 2018.04.23)