

ゲームとパズルと計算量

上原 隆平 (北陸先端科学技術大学院大学)*

概 要

Some puzzles and games give reasonable characterizations to computational complexity classes. For example, “pebble game” is a classic model that gives some complexity classes in a natural way, and “constraint logic” is recent model that succeeds to solve a long standing open problem due to Martin Gardner who asks the computational complexity of sliding block puzzles. Such puzzles gives us “typical” and characterization and “intuitive” understanding for some computational complexity classes. On the other hand, there are still some puzzles and games that give nontrivial interesting aspects of computational complexity classes. For example, let us consider “14-15 puzzle” which is a classic well known sliding puzzle. By parity, we can determine if one arrangement can be slid to the other in linear time. Moreover, we can always find a way for sliding between them in quadratic time. However, interestingly, finding the optimal solution is NP-complete in general. Through such classic puzzles, the reconfiguration problems are recently well investigated as a new framework of characterization of computational complexity classes. I give some recent results for these games and puzzles on graphs from the viewpoint of theoretical computer science.

1. ゲームやパズルにおける計算(量)理論

数理パズル、より広く、いわゆるレクリエーション数学の歴史は、計算理論よりも長い。こうしたレクリエーション数学の中で楽しまれてきたパズルやゲームの中には、計算理論、あるいは計算量理論との相性がよい問題が数多くある。そうした背景から、計算理論の文脈の中で多くのパズルやゲームの研究が行われてきた。逆に、計算モデルの研究の中から生まれた数理パズルやゲームも少なからず存在する。

こうした数理パズルと計算(量)理論の研究には、2つの側面がある。1つ目は、すでにある数理パズルの「難しさ」が、ある程度理論的に評価できるという側面である。例えば($NP \neq PSPACE$ を前提とすれば)NP完全なパズルよりも、PSPACE完全なパズルの方が、一般には難しいだろう。こうした相対的な問題の難しさの比較は、計算量理論の応用の中でも王道と言える。もう1つは、こうした数理パズルの研究により、計算(量)理論における難しさの分類に、新しい指標が導入できるという側面である。例えば、数理パズルからの抽象化で作られた、ある種のグラフ上の問題は、問題設定によって多項式時間で解けたり、NP完全になったり、さらにはPSPACE完全になったりする。ある種の単純な問題によって特徴づけられた計算量クラスは、従来の Turing 機械による特徴づけよりも、直感的にわかりやすい。非決定性計算に対する直感がなくても、ある種のパズルの難しさなら、直感的にわかることもあろう。従来の計算量クラスに対する「新たな解釈」を与えることは、例えば $P \neq NP$ 予想に対するアプローチ

* 〒923-1292 石川県能美市旭台 1-1 北陸先端科学技術大学院大学 情報科学系
e-mail: uehara@jaist.ac.jp
web: <http://www.jaist.ac.jp/~uehara/>

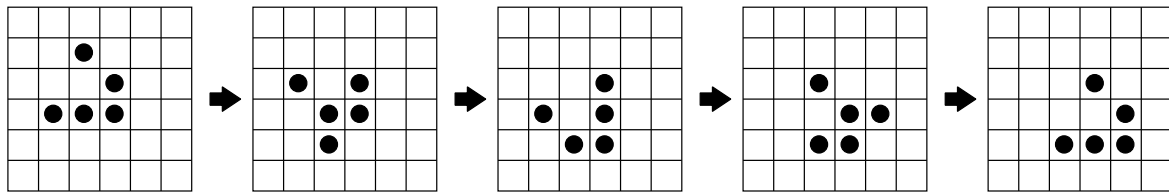


図 1: ライフゲームのグライダーと呼ばれるパターンが世代交替して移動する様子

を増やすという意味でも、価値がある。

本稿では、こうした計算理論や計算量理論的な面から見た、数理パズルやゲームの研究の歴史や今後の展望を紹介しよう。時代によって流行りすたりがあるものの、ときにはゲームやパズルの研究が理論的な枠組を提供し、ときには理論的な研究によって実際のパズルやゲームの困難性の研究が大きく発展している様子がわかる。

1.1. 黎明期

計算理論研究の黎明期から、現代まで連綿と続いている計算モデルと言えば、セル・オートマトンである。「状態」を持ったオートマトンは、自分の周囲の状況を入力として受取り、次の「状態」へと遷移する。この単純なモデルは、多くの人を引きつける魅力があるようで、例えばかの Turing も、ある種の貝殻の模様をオートマトンで模倣する研究をしており、近年も Mathematica で有名な Wolfram が信じられない厚さ (約 1200 ページ!) の本 [Wol02] を書いている。

セル・オートマトンに関連したゲームといえば、ライフゲーム [BCG04] が代表格であろう。ライフゲームは (群論の研究などで著名な) 数学者 Conway が 1970 年ごろに考案したゲームであり、ルールは単純である。正方格子上に配置された「生命 (ライフ)」は、自分に隣接する 8 つのマスの状態により、次の世代の生死が次のように決まる。

- あるマスにライフがいて、近傍に 2 つあるいは 3 つのライフがあれば、そのライフは次の世代も生き延びる。1 つ以下あるいは 4 つ以上の場合、過疎あるいは過密のために次の世代では死んでしまう。
- あるマスにライフがおらず、近傍に 3 つのライフがあれば、そのマスには次の世代でライフが誕生する。

ある世代において、すべてのマスに同時に上記の規則が適用される。単純な例を図 1 に示す。学生のプログラミング演習の課題にはちょうどよいレベルであり、自分でプログラムを作るのも楽しい。しかし今は Golly という有名なプログラム¹があり、これを利用する方が賢明だ。やや余談であるが、この Golly は、専用に開発された高度なアルゴリズムを使っており、驚くほど大きな盤面のライフゲームを信じられないほど高速に計算してくれる。

ともあれ、ライフゲームは単純なルールであるにも関わらず、非常にドラスティックに盤面が変化する興味深いゲームである。Golly を使って遊ぶと、時間がたつのを忘れてしまう。興味深い例として、図 1 に格子上を移動していく「グライダー」を示した。実は、このグライダーを無限に生成する「グライダー砲」も知られている。このグライダー砲を使えば、有限個のライフから始めて、無限にライフを増殖させることが可能

¹<http://golly.sourceforge.net>

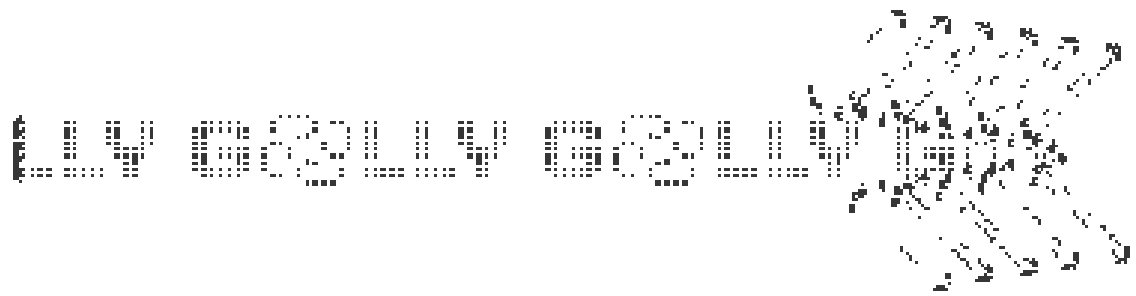


図 2: ライフゲームで GOLLY という文字列を無限に生成する様子

である。しかもそれだけではない。複数のグライダー砲をうまく組み合わせることで、「ライフゲームは Turing 完全である」ことが証明されている。本稿では「Turing 完全」の詳細は述べないが、直感的には、Turing 機械で計算できる問題は、すべてライフゲームでも計算できるという意味である。言い換えると、ライフゲームを使えば Turing 機械のあらゆる計算を模倣できる。つまり、現在のフォン・ノイマン型コンピュータで計算できることは、すべてライフゲームで計算できる。上記の Golly をもってすれば、かなり高度な計算もできそうである。実際、Golly の Web ページを見ると、ライフゲームで Golly という文字を延々と生成するデモ（図 2）や、ライフゲームで素数を計算するというデモもあり、「ライフゲームのもつ万能性」の一端を感じることができる。

1.2. 1970 年代～1980 年代

1970 年代も半ばを過ぎると、計算可能性を主に議論する計算理論というよりは、計算の効率を議論する計算量理論に関連した結果が多くなった。これは、計算の理論における「標準的な計算モデルの模索」という研究の段階に区切りがつき、単に計算できるかできないかといったことだけでなく、例えば P や NP、PSPACE といった計算効率を意識した計算量クラスが一般的な困難性の指標として定着したことによるものであると考えられる。

このころは「ペブルゲーム」という抽象化ゲームが提案され、さまざまな研究がなされた（[LT80, KAI79] など）。これは大雑把には次のような枠組である。

- グラフと、その上に置かれたいくつかの小石（ペブル）
- 小石は、辺に沿って別の頂点に移動することができ、そのときに別の小石を取り除く。この遷移ルールは入力の一部として与えられる。（図 3 に典型的な遷移の様子を示す。）
- ある小石を指定されたゴールへと運べるかどうかを判定する。

ゲームという名前では呼ぶには、やや人工的に見えるかもしれない。ともあれ、この枠組を用いて、笠井らは $NLOGSPACE/P/NP/PSPACE/EXP$ といった代表的な計算量クラスの完全問題を示した [KAI79]。これは本稿の文脈で捉え直すと、「Turing 機械によらない計算量クラスの特徴づけ」という意味からは先駆的な結果と言えよう。

こうした「単純なモデルによる各種の計算量クラスの特徴づけ」という結果は、それぞれの計算量クラスに対して、新たな解釈や視点を与えるという意味で重要である。

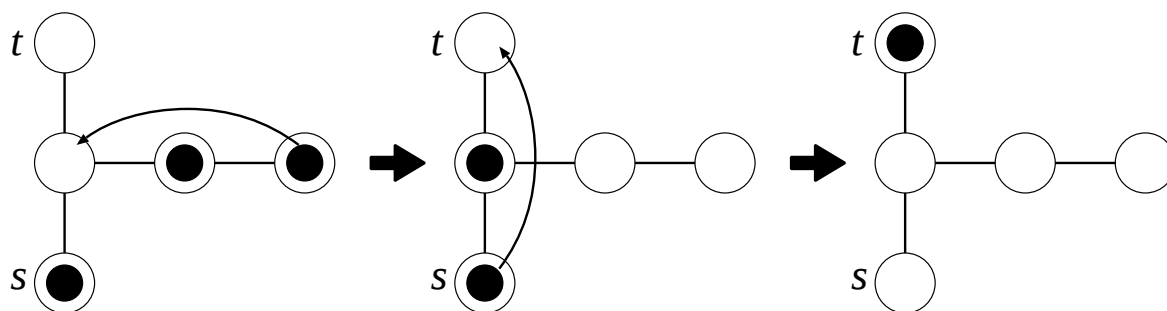


図 3: ペブルゲームで頂点 s のペブルを頂点 t に移動する様子

つまり、例えば P で解ける（多項式時間で解ける）ペブルゲームと、 NP 完全問題であるペブルゲームの違いを考えることは、クラス P とクラス NP の違いを考えることに他ならず、この本質的な違いを明らかにできれば、今なお未解決な $P \neq NP$ 予想への足掛かりになる。

ともあれ、こうした「Turing 機械によらない計算量クラスの特徴づけ」の研究が進んだことで、ある種の計算量クラスについての直観が養われ、多くのパズルやゲームの計算量理論的な観点からの困難性の解明が進んだ。

1.3. 1980 年代 ~

1980 年代以降、数多くのパズルやゲームの計算量理論的な困難性が明らかになっていった。オセロ [IK94]、将棋 [AKI87]、ペグソリテア [UI90]、ジグソーパズル ([DD07] が詳しい) など、実際に人々がプレイしている、多くの自然なゲームやパズルが計算量の観点から研究された。この時代、多くのゲームやパズルに関する困難性が示されて、その結果として、計算の困難性において、ある種の「勘」が発達してきた。つまり、あるパズルやゲームの問題が与えられたときに、それが一人でやるパズルなのか、二人でやるゲームなのか、といった点や、盤面がある種の単調性を持つかどうか（例えばペグソリテアでは、1 手ごとにペグが必ずちょうど 1 つ減るが、将棋は「千日手」に見られるように同じ盤面が 2 度以上現れることがある）といった点から、だいたいの計算量クラスが前もって見積もれるようになってきた。

1.3.1. ペンシルパズルとクラス ASP

上記の流れの中で、ペンシルパズルは、一つの大きな支流をなしている。日本の場合は特に雑誌「ニコリ」(<https://www.nikoli.co.jp>) が果たしてきた役割は大きい。数独（ナンバープレース）、ナンバーリンク、カックロ、天体ショー、シャカシャカなどなど、いわゆるペンシルパズルと呼ばれる一大ジャンルは、雑誌ニコリを通じて生まれ、普及したものが多く、² このあたりの最新状況については、2018 年の段階では文献 [IH18] が詳しいが、文献 [IH18] が発行されたあとも、いくつかのパズルの NP 完全性が示されており、ニコリの公式ページに載っているパズル (<https://www.nikoli.co.jp/en/puzzles/index.html>) はほとんどすべてが NP 完全であることが知られている。

計算量理論の観点からいえば、1 人で楽しむタイプのペンシルパズルは、 NP 完全であるものが多い。ごく直観的な説明をすれば、このタイプのペンシルパズルは、与え

² なお「数独」はニコリの登録商標であり、国内で同社が制作に関与していないものはナンバープレース（ナンプレ）と呼ばれる。海外では商標登録されておらず、sudoku の名前で親しまれており、origami と同様、日本語由来の英単語として認知され、定着している。

られたパズルの問題に対して、規則にしたがって何かを選び、それを鉛筆で書き込み、その選択が正しいければ、それ以降、書き直すことがない。そしてまた、その選択の結果、最終的に正しい解が得られたら、人間がそれを簡単に確認できる。こうした条件を考えると、少なくともNPに入ることはすぐにわかる。したがって困難性さえ示せばNP完全となる。

さて、ペンシルパズルは、ほとんどすべてがNP完全であることがわかれば、それで良いのだろうか。そこには少しパズル特有の事情がある。パズル業界では「意図解」という言葉が使われることがある。これは「問題の作者が、解き手に見つけてもらいたかった、意図した解」という意味である。意図解があるということは、意図以外の解も考えられる。これは通常「別解」と呼ばれ、作者の意図しない別解があるパズルは、場合によっては「パンクした」と言われることもある。こうした別解があるかどうかを確かめるとい問題は、パズルデザインという観点からは重要である。つまり、問題と意図解が入力として与えられたとき、別解があるかどうかを判定するという「別解問題」が、パズル的な観点からは自然な問題として定義される。この別解の有無を確認する問題のクラスとして、計算量クラスASPが瀬田によって提案されている[Set02, YS03]。

筆者の主観では、この計算量クラスASPは計算量理論の観点からも興味深いと考えている。あるNP完全な問題のインスタンスに対して解の存在が保証されているとき、そのインスタンスに別解があるかどうかを判定する問題は、元のNP完全問題の通常のインスタンスの判定問題と比較して、どちらが難しいのであろうか。計算量クラスASPは、計算量理論の観点からは、あまり研究されておらず、今後の課題と言えそうである。例えばPapamitritiouによる古典的な計算量理論の教科書[Pap94]を調べると、古典的な計算量クラスとして、「解が一つしかないかどうか」を判定する計算量クラスUPや、NPとcoNPの二つのクラスを用いて定義される計算量クラスDPなどがある。ASPが、こうした従来の計算量のクラスと、どのような関係にあるかが研究されれば、従来の計算量クラスに対して、パズルという観点からの新しい知見を与えられるかもしれない。

1.4. 2000年代

近年になると、実際のパズルの計算量を示す研究としては、よりゲーム性の高いものが好まれるようになってきた。例えばリアルタイムで遊ばれるテトリス[BDH⁺04]や、カードゲームUNO[DDU⁺14]、あるいはルービック・キューブ[DDE⁺11, DER18]の計算量の研究もあれば、果てはマリオを始めとする任天堂の古典的なビデオゲームが、どれも全般にNP困難であるという総括的な論文[ADGV15]まである。

こうした非常にポピュラーなゲームとなると、マニアのゲーマーも多く、証明の中で想定しているキャラクターの動きに対して、いろいろとコメントがつくことがあるようだ。彼らの発表を聞いていると、機械語で書かれたビデオゲームのバイナリファイルを読み出して、わざわざ逆アセンブルして仕様を調べたり、かなり周到な調査までやっていて、そこまでくると、さすがに論文を書く方も大変である。

若干のやりすぎ感は否めないが、ある種のゲームやパズルと、計算量クラスの間の関係は、近年では、かなり明らかになり、ある種の「勘」も働くようになった。例えばニコリや任天堂が新しいパズルやゲームを提案すると、その計算量的な困難性は、ある程度は予想がたつようになり、証明の技法もかなり確立されつつある。

逆に、ある種のゲームの楽しさや難しさが、例えばNP計算によるのであれば、それ



図 4: スライディングブロックパズルの例

を逆手にとったゲームが考案されても良さそうである。典型的な NP 完全問題から派生させた、面白いゲームができて良さそうなものだが、そういう例は聞いたことがない。ゲーム開発者の間に、こうした計算理論の結果が、まだまだ普及してないことが一因であろうか。

1.5. 遷移問題と制約論理

上記の歴史の中で、いろいろな形で研究が進められて、近年大きな潮流となった一群のパズルや問題群がある。最後にこれを紹介しよう。まずこの問題について、最初に指摘したのは、アメリカのレクリエーション数学の大家、マーティン・ガードナーである。1964 年の彼の記事 [Gar64] によれば「こうしたパズルには理論というものが強く求められている。与えられた状態から別の状態が得られるかどうかを判定する方法は、ちょっとした試行錯誤を除いて、まったく知られていない...」とのことであった（日本語訳は [HD09] の邦訳による）。この「こうしたパズル」とは、スライディングブロックパズルのことである。スライディングブロックパズルと言う名前は知らなくても、具体例を見れば誰でも「アレか」と思うであろう古典パズルである。ちょうど筆者の机の近辺にある具体例を図 4 に挙げよう。

最も有名なスライディングブロックパズルは、 4×4 の盤面に 1 から 15 までの数字が書かれたピースが置かれた「15 パズル」であろう。1 つだけ空いた場所があり、それを利用してピースをスライドして盤面の配置を変えていくパズルである。図 4 の左側は、これを一回り小さくした「8 パズル」のキーチェーンである。15 パズルに関する俗説として「1878 年にパズル作家のサム・ロイドが「14 と 15 を入れ替えた配置から元に戻す」という「解けない問題」を出題し、1000 ドルの賞金をかけたため、パズルは有名になり、ロイドは金持ちになった」と言われている。しかしこれは、ロイド自身が流布し

た作り話であったことが判明している。そのあたりのパズルの歴史は[Gar08]や[SS06]に詳しい。歴史的真相はともあれ、1880年代には、その後のルービック・キューブをしのぐ流行を巻き起こしたことは確かなようだ。

この15パズルを一般化しよう。盤面の大きさは $n \times n$ であり、この上に $n^2 - 1$ 枚のピースが置かれている。つまり1箇所だけ空いている。それぞれのピースには1から $n^2 - 1$ までの数字が書かれていて、空いている場所を利用してピースをスライドするわけだ。この $n^2 - 1$ パズルの問題では、2つの配置 S と T が与えられる。そして S から T まで遷移できるかどうか問われる。これはごく自然な問題である。そして、この $n^2 - 1$ パズルは、計算量理論の観点から、非常に興味深い性質を持っている。

まず、このパズルにはパリティがある。このパリティの正確な定義は、反転数が絡み、少しややこしいので本稿では割愛する。ともあれ、ある配置 S から別の配置 T に遷移できる必要十分条件は、パリティが一致することである。そしてこのパリティは、盤面の大きさに対して線形時間で計算できる。つまり、このパズルの「Yes/Noタイプの判定問題」は、線形時間で簡単に解ける。

解を判定してみたら、Yesだったとしよう。すると当然、 S から T への遷移列が求めたくなるだろう。一般の $n^2 - 1$ パズルの場合、これは実は隅から1つずつ T の配置に並べていくことが可能で、実際の手順を吟味すると、全体で $O(n^3)$ 回スライドすれば十分であることがわかる。つまり多項式長の解を多項式時間で求めることができる。

さて当然、次は最短手数で解きたくなるだろう。ところがここで驚いたことに、 $n^2 - 1$ パズルの最短手数を求める問題はNP完全なのである。(この問題のNP完全性は1990年に[RW90]で示されているが、2017年に、もっとずっとわかりやすい証明が[DR17]で与えられている。)

この $n^2 - 1$ パズルの性質は、既存のパズルやゲームの計算量の振舞いとは、少し違って見える。そしてこのあたりには、さらに続きがある。そもそもガードナーが1960年代に問題提起したパズルは、 $n^2 - 1$ パズルをさらに一般化したスライディングブロックパズルである。 $n^2 - 1$ パズルではピースはすべて単位正方形であったが、一般のスライディングブロックパズルでは、ピースの形は長方形である。このタイプのスライディングブロックパズルも、15パズルに負けず劣らずの歴史のあるパズルである。日本では「箱入り娘」という名前でも知られており、北米でもDad Puzzleと呼ばれる古典的なパズルである。図4の右に示したものは、著者が中国で入手したもので、中国でも古典的なパズルであることがわかる。

ガードナーが1960年代に指摘したにも関わらず、この種のパズルの計算量は、長い間、よくわかっていなかった。筆者の知る限り、日本発のパズルゲームである倉庫番がPSPACE完全であることを示した論文[Cul97]あたりが、かなり先駆的な結果であったと思われる。しかしこの論文では、倉庫番でTuring機械を模倣するという、王道と言えば王道だが、あまりやりたくない証明技法を使っていて、その後に結果が続かなかった。

ここにブレークスルーをもたらしたのが、Robert HearnとErik D. Demaineの提唱した「制約論理」である。彼らは2005年に[HD05]でこの枠組を提案し、この枠組を用いて、スライディングブロックパズルやそれに類する一連のパズルの計算量を解き明かした。この成果はその後、Hearnの博士論文となり、書籍化されて販売されている[HD09]。端的には、ガードナーが指摘したパズル類は、その多くが、一般にPSPACE完全となることが判明した。

制約論理は、名前からは想像しにくい、グラフ上のある種のゲームである。かつてのペブルゲームの進化形とも見なせるかもしれない。ペブルゲーム同様、条件をいろいろと変えることで、様々な計算量クラスを特徴づけることができる。特に、一般に操作がリバーシブルで、同じ状態に何度でも戻すことができる点が、典型的なNP完全問題であるペンシルパズルや、ペブルゲームとの際立った違いである。近年になり、制約論理は、ゲームやパズルに限らず、さまざまな問題の計算量的困難性を示すためのツールとして、使われつつある。

こうした中、スライディングブロックパズルのような問題をきちんと定式化した問題は、大きく「組合せ遷移問題」と名付けられ、活発な研究が進められている。組合せ遷移問題とは、基本的には次のような枠組である。

定義 1.1 (組合せ遷移問題) 問題 P と、その問題の解 S, T と、遷移規則が与えられたとき、 P の解の列 $S = X_0, X_1, \dots, X_k = T$ となる解の列があるかどうかを判定せよ。ただし (X_i, X_{i+1}) は遷移規則を満たすものとする。

この定義だけだとわかりにくいので、典型的な例として、充足可能性問題 SAT を考えよう。通常の SAT は、命題論理式 $F(x_1, x_2, \dots, x_n)$ が与えられたときに、 $F()$ を充足するブール変数 $x_1, \dots, x_n$ の割当てがあるかどうかを判定する問題である。このとき SAT の遷移問題とは次のように定義される。

例 1.1 (SAT の遷移版) 命題論理式 F と、 F を充足する2つの解 $S = (s_1, s_2, \dots, s_n)$ と $T = (t_1, t_2, \dots, t_n)$ が与えられる。(各 s_i と t_j は定数。) このとき、「1ビットだけ値を反転する」という操作を繰り返して、 F の充足可能性を満たしたまま、 S を T に遷移させられるかどうかを判定せよ。

SAT の遷移問題版についての結果は、遷移問題の研究の先駆けであり、2006 年に [GKM06] で PSPACE 完全性が示された。その後、典型的な問題を数多く扱ったマイルストーン的な論文 [IDH⁺11] も出て、枠組として定着した。こうした遷移問題は、電力会社とユーザの割当てを決める問題などに具体的な応用もあり、遷移問題に特化した国際会議も何度か開催されている (<http://www.ecei.tohoku.ac.jp/alg/core/>)。最新の結果をまとめたサーベイ論文もある [Nis18]。

パズルの文脈に戻れば、スライディングブロックパズルは、初期配置と目指す配置が問題の「解」として与えられていて、遷移規則は「空いたところにブロックをスライドできる」と考えることができ、こうした遷移問題を単純化・抽象化しているという観点から、重要な意味をもつ。そして制約論理は、その計算量を示すための有用なツールとしての地位を獲得しつつある。しかし制約論理は、まだツールとしての使用に留まっていて、「Turing 機械によらない計算量クラスの特徴づけ」の一つとしての研究は、まだそれほど進んではない。今後、ツールとしての使用実績が蓄積し、それによって制約論理自身の研究が進めば、それを通じて、ある種の計算量クラスについての直観が養われて、計算量クラスに関する新たな知見が得られることになるかもしれない。

参考文献

- [ADGV15] Greg Aloupis, Erik D. Demaine, Alan Guo, and Giovanni Viglietta. Classic Nintendo Games are (Computationally) Hard. *Theoretical Computer Science*, 586:135–160, 2015.

- [AKI87] H Adachi, H. Kamekawa, and S. Iwata. Shogi on $n \times n$ board is complete in exponential time (in Japanese) . *Trans. IEICE Japan*, J70-D(10):1843–1852, 87.
- [BCG04] Elwin R. Berlekamp, John H. Conway, and Richard K. Guy. *Winning Ways for Your Mathematical Plays*, volume 4. A K Peters/CRC Press, 2nd edition, 2004.
- [BDH⁺04] Ron Breukelaar, Erik D. Demaine, Susan Hohenberger, Hendrik Jan Hoogeboom, Walter A. Kosters, and David Liben-Nowell. Tetris is Hard, Even to Approximate. *International Journal of Computational Geometry and Applications*, 14(1-2):41–68, 2004.
- [Cul97] J. Culberson. Sokoban is PSPACE-complete. Technical Report TR97-02, Department of Computer Science, The University of Alberta, 1997. <http://web.cs.ualberta.ca/~joe/Preprints/Sokoban/index.html>.
- [DD07] Erik D. Demaine and Martin L. Demaine. Jigsaw Puzzles, Edge Matching, and Polyomino Packing: Connections and Complexity. *Graphs and Combinatorics*, 23[Suppl]:195–208, 2007.
- [DDE⁺11] Erik D. Demaine, Martin L. Demaine, Sarah Eisenstat, Anna Lubiw, and Andrew Winslow. Algorithms for Solving Rubik’s Cubes. In *Proceedings of the 19th Annual European Symposium on Algorithms (ESA 2011)*, pages 689–700. Lecture Notes in Computer Science Vol. 6942, Springer-Verlag, 2011.
- [DDU⁺14] Erik D. Demaine, Martin L. Demaine, Ryuhei Uehara, Takeaki Uno, and Yushi Uno. UNO is hard, even for a single player. *Theoretical Computer Science*, 521:51–61, 2014.
- [DER18] Erik D. Demaine, Sarah Eisenstat, and Mikhail Rudoy. Solving the Rubik’s Cube Optimally is NP-complete. In Rolf Niedermeier and Brigitte Vallée, editors, *35th Symposium on Theoretical Aspects of Computer Science (STACS 2018)*, volume 96 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 24:1–24:13, Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.
- [DR17] Erik D. Demaine and Mikhail Rudoy. A simple proof that the $(n^2 - 1)$ -puzzle is hard. arXiv:1707.03146, July 2017.
- [Gar64] Martin Gardner. The Hypnotic Fascination of Sliding-Block Puzzles. *Scientific American*, 210:122–130, 1964.
- [Gar08] Martin Gardner. *Hexaflexagons, Probability Paradoxes, and The Tower of Hanoi*. The New Martin Gardner Mathematical Library. Cambridge University Press, 2008. (邦訳 : 岩沢宏和・上原隆平監訳 (2017) 『ガードナーの数学パズル・ゲーム』 , 日本評論社 , 2015 .) .
- [GKMⁱ06] P. Gopalan, P. G. Kolaitis, E. N. Maneva, and C. H. Papadimitriou. The Connectivity of Boolean Satisfiability: Computational and Structural Dichotomies. In *Automata, Languages and Programming, 33rd International Colloquium (1) (ICALP 2006(1))*, pages 346–357. Lecture Notes in Computer Science Vol. 4051, Springer-Verlag, 2006.
- [HD05] Robert A. Hearn and Erik D. Demaine. PSPACE-completeness of sliding-block puzzles and other problems through the nondeterministic constant logic model of computation. *Theoretical Computer Science*, 343(1-2):72–96, 2005.
- [HD09] R. A. Hearn and E. D. Demaine. *Games, Puzzles, and Computation*. A K Peters Ltd., 2009. (邦訳 : 『ゲームとパズルの計算量』 上原隆平訳 , 近代科学社 , 2011 .) .
- [IDH⁺11] Takehiro Ito, Erik D. Demaine, Nicholas J.A. Harvey, Christos H. Papadimitriou, Martha Sideri, Ryuhei Uehara, and Yushi Uno. On the complexity of reconfiguration problems. *Theoretical Computer Science*, 412:1054–1065, 2011.

- [IH18] Chuzo Iwamoto and Masato Haruishi. Computational Complexity of Usowan Puzzles. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E101-A(9):1537–1540, 2018.
- [IK94] Shigaki Iwata and Takumi Kasai. The Othello game on an $n \times n$ board is PSPACE-complete. *Theoretical Computer Science*, 123(2):329–340, 1994.
- [KAI79] Takumi Kasai, Akeo Adachi, and Shigeki Iwata. Classes of Pebble Games and Complete Problems. *SIAM Journal on Computing*, 8(4):574–586, 1979.
- [LT80] R.J. Lipton and R.E. Tarjan. Applications of a Planar Separator Theorem. *SIAM Journal on Computing*, 9(3):615–627, 1980.
- [Nis18] Naomi Nishimura. Introduction to Reconfiguration. *Algorithms*, 11(4):1–25, 2018.
- [Pap94] C. H. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.
- [RW90] Daniel Ratner and Manfred Warmuth. The $(n^2 - 1)$ -puzzle and related relocation problems. *Journal of Symbolic Computation*, 10(2):111–137, 1990.
- [Set02] Takahiro Seta. *The Complexities of Puzzles, Cross Sum and Their Another Solution Problems (ASP)*. Senior thesis, University of Tokyo, 2002.
- [SS06] Jerry Slocum and Dic Sonneveld. *The 15 Puzzle Book: How it Drove the World Crazy*. Slocum Puzzle Foundation, 2006.
- [UI90] R. Uehara and S. Iwata. Generalized Hi-Q is NP-Complete. *The Transactions of the IEICE*, E73(2):270–273, 1990.
- [Wol02] Stefan Wolfram. *A New Kind of Science*. Wolfram Media, 2002.
- [YS03] Takayuki Yato and Takahiro Seta. Complexity and Completeness of Finding Another Solution and Its Application to Puzzles. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E86-A(5):1052–1060, 2003.