

# 不動点定理と計算量理論

九州大学大学院 数理学府 数理学専攻  
石塚天 (Takashi ISHIZUKA)

## 概要

計算量クラス TFNP は、解の存在が保証された計算量クラスである。TFNP の難しさを議論するとき、数学における存在定理（鳩の巣原理、握手補題、不動点定理など）が活用される。Nash 均衡と Brouwer の不動点一致するという事実から、不動点計算問題の難しさは社会科学・計算機科学の双方から注目されている。本発表では、講演者の結果を交えつつ、不動点定理と計算量理論の関係を紹介する。

## 1 はじめに

アルゴリズム (algorithm) とは、計算問題を解く手法をひとつひとつ示したものである。多くの場合、アルゴリズムの効率性は、『P vs. NP 問題』の文脈で議論される。これは、ミレニアム懸賞問題のひとつであり、数学・理論計算機科学双方の分野における最重要未解決問題である。

計算量理論とは、計算問題の難しさを定量的に評価し、効率的なアルゴリズムの有無を明確にする研究分野である。計算問題には、大きく分けてふたつの形式がある。ひとつは、Yes/No で答えることが可能な判定問題である。もうひとつは、具体的な解を出力する探索問題である。以下に具体例を示す：

判定問題 自然数  $N$  は素数であるか？

探索問題 自然数  $N$  を割り切る素数  $p$  は何か？

前者の問題は、Yes/No で答えることができる；自然数  $N$  が素数であるか否かは、一意に定まる。一方で、後者の問題を Yes/No で答えることは不適切である。具体的な自然数（より正確には素数）を提示する必要がある。

計算量クラス P と NP は、判定問題に関するクラスである。Papadimitriou [20] は、『P vs. NP 問題』を探索問題の文脈で、等価な『FP vs. FNP 問題』へと拡張した。

探索問題は、『与えられた入力に対する適切な解を見つける問題』として定式化される計算問題である。本稿では、以下のふたつの条件を満たす探索問題を扱う：

1. 解のサイズは、入力長の多項式サイズである；
2. 解の妥当性は、多項式時間で判定できる。

上記のふたつの条件を満たす探索問題を多項式時間限定探索問題と呼ぶ。本稿では、多項式時間限定探索問題しか扱わないため、これらを単に探索問題と呼ぶことにする。計算量クラス FNP は、探索問

題全体のクラスである。探索問題が完備性 (totality) を持つとは、次の条件を満たすときをいう：

- 任意の入力に対して、少なくともひとつの解が存在する。

完備性を満たす探索問題のことを完備探索問題と呼ぶ。計算量クラス TFNP は、完備探索問題全体のクラスである。

探索問題の計算量理論では、一般的に TFNP の部分クラスの複雑さに焦点をあてる。Papadimitriou [20] は、完備性の証明方法に基づき、探索問題をいくつかのクラスへ分類した。代表的な計算量クラスは、PPAD, PPA, PPP などである (詳細については、[15] を参照すること)。

多くの場合、探索問題の計算量理論の研究動機は、応用的な側面 — 実社会に現れる計算問題の解決 — から来ることが多い。局所最適解を求める問題 [17] や、経済モデルの均衡点を求める問題 [7, 25], 公平な配分・割り当てを求める問題 [14, 18, 22] などである。これらの問題は、いずれも解の妥当性検証が容易であり、常に解を有するような問題である。一方で、効率的な計算手法が確立されていない問題となる。

本稿の目的は、不動点定理と計算量理論の関係を紹介することにある。特に、『不動点定理によって存在が保証された不動点を計算する問題 (不動点計算問題)』の難しさに関する一連の研究の流れを紹介する。

## 2 探索問題の計算量理論

### 2.1 準備

ここでは、本稿で使用する記号や用語を導入する。なお、計算量理論に関する基本的な事項は、標準的な教科書を参照されたい (たとえば、[1, 21, 23] など)。

本稿では、自然数は 0 を含まないものとし、自然数全体の集合を  $\mathbb{N}$  で表す。さらに、非負整数全体の集合を  $\mathbb{Z}_{\geq 0}$  で表し、非負実数全体の集合を  $\mathbb{R}_{\geq 0}$  で表す。また、実数全体の集合を  $\mathbb{R}$  でかく。自然数  $n$  に対して、 $[n] = \{1, 2, \dots, n\}$  と定める。ふたつの相異なる実数  $a < b$  に対して、 $[a, b] = \{x \in \mathbb{R} ; a \leq x \leq b\}$  とする。

有限集合  $\Sigma$  を『文字』の集合とする。直感的には、計算機で扱うことが可能な『文字』の集まりと捉えればよい。一般性を失うことなく、 $\Sigma = \{0, 1\}$  を仮定してよい (証明は、標準的な計算理論の教科書を参照すること)。文字列とは、『文字』の有限列である。つまり、任意の文字列  $s$  に対して、適切な自然数  $n$  が存在して、 $s = s_1 s_2 \cdots s_n$  と書くことができる；ここで、 $s_i \in \Sigma \forall i \in [n]$ 。このとき、 $n$  を文字列  $s$  の長さ (あるいは、文字列長) と呼ぶ。文字列  $s$  の長さを  $|s|$  で表す。自然数  $n$  に対して、 $\Sigma^n$  で、長さ  $n$  の文字列全体の集合を表す。さらに、 $\Sigma^*$  で、文字列全体の集合を表す。

### 2.2 探索問題とは？

関係  $R \subseteq \Sigma^* \times \Sigma^*$  が、 $|y| \leq \text{poly}(|x|) \forall (x, y) \in R$  を満たすとき、多項式バランスをもつという。また、 $R$  が多項式時間検証可能であるとは、 $(x, y) \in R$  を多項式時間で判定できるときをいう。関係  $R$  が完備であるとは、すべての入力  $x \in \Sigma^*$  に対して、少なくともひとつの解  $y \in \Sigma^*$  such that

$(x, y) \in R$  が存在するときをいう。以下、本稿では、多項式バランスかつ多項式時間検証可能な関係  $R \subseteq \Sigma^* \times \Sigma^*$  のみを考える。

いま、 $R, S$  を探索問題とする。探索問題  $R$  から  $S$  への多項式時間帰着  $R \leq_P S$  は、ふたつの多項式時間計算可能関数  $f, g$  で次のように定められる： $R$  の各入力  $I_R$  に対して、 $f(I_R)$  は  $S$  の入力である；さらに、 $f(I_R)$  に対応する適切な解  $y$  について、 $g(y, I_R)$  は、 $I_R$  に対する適切な解である。

計算量クラス  $C$  に対して、探索問題  $R$  が  $C$  困難であるとは、 $C$  に属するすべての探索問題  $S$  から、 $R$  への多項式時間帰着が存在するときをいう。さらに、 $C$  困難である探索問題  $R$  が、計算量クラス  $C$  に属するとき、 $R$  は  $C$  完全であるという。

関係  $R \subseteq \Sigma^* \times \Sigma^*$  に関する探索問題を次のように定める：

**定義 2.1** (Papadimitriou [20]). Given a string  $x \in \Sigma^*$ , find a string  $y \in \Sigma^*$  such that  $(x, y) \in R$  if such a  $y$  exists, and return the string “no” otherwise.

計算量クラス FNP は、多項式時間検証可能かつ多項式バランスをもつ関係  $R \subseteq \Sigma^* \times \Sigma^*$  に関する探索問題で構成されるクラスである。計算量クラス TFNP は、完備な関係  $R$  で構成される計算量クラス FNP の部分クラスである。計算量クラス FP は、多項式時間で探索問題を解くことが可能な計算量クラスである。定義から明らかに次の関係が成り立つ。

**命題 2.2.**  $FP \subseteq TFNP \subseteq FNP$ .

これらの包含関係が、真であるか、等号が成立するか、は重要な未解決問題である。特に、 $FP \stackrel{?}{=} FNP$  は、ミレニアム懸賞問題のひとつである  $P \stackrel{?}{=} NP$  と等価な問いである。

**定理 2.3** (Megiddo and Papadimitriou [19]).  $FP = FNP$  if and only if  $P = NP$ .

### 3 不動点定理と計算量理論

本章では、連続探索問題の複雑さを扱う。つまり、実数値関数が入力として与えられ、望みの実数値解を出力する問題である。この問題を定式化するため、算術回路 (Arithmetic Circuit) を導入する。本稿における算術回路の定義は、Daskalakis and Papadimitriou [8] と同じである。

#### 3.1 Arithmetic Circuit

関数  $f : [0, 1]^n \rightarrow [0, 1]^m$  を計算する算術回路は、変数でラベル付された  $n$  個の入力ゲート、有理数でラベル付された幾つかの入力ゲートと、 $m$  個の出力ゲートを持つ非巡回有効グラフである。全ての内部ノードは、二部ゲートであり、次の演算のいずれかでラベル付されている： $\{+, -, *, \max, \min, >\}$ ；ここで、演算  $>$  は、入力  $(x, y) \in \mathbb{R} \times \mathbb{R}$  に対して、 $x > y$  のとき 1 を出力し、それ以外は 0 を出力する。

算術回路の入力と出力は、 $[0, 1]$  区間上に制限する。出力が負値をとるとき、算術回路の出力を 0 で再定義する。一方で、出力が 1 を超えるとき、算術回路の出力を 1 で再定義する。算術回路  $C$  の

サイズ<sup>\*1</sup>を  $\text{size}(C)$  でかく。

上記の算術回路の定義は, [8] に基づく。しかし, これには若干の問題が含まれている。それは, 算術回路が多項式時間で計算できない可能性である。それどころか, 多項式時間階層に含まれているかすら定かではない。

計算量クラス FNP に属する探索問題を考えるためには, 算術回路に幾つかの制約を課す必要がある。Daskalakis and Papadimitriou [9] は, この問題を解決する訂正手法を提案した。本稿では, 彼らによる制約ではなく, Fearnley et al. [12] によって提案された手法を用いる。彼らの定式化は, Daskalakis and Papadimitriou [9] による定式化と等価であるだけでなく, 幾つかのアドバンテージを有している。

算術回路  $C$  が, *well-behaved* であるとは, 次の条件を満たすときをいう:

- 出力に到達するすべての有向パスは, 高々  $\log(\text{size}(C))$  個の真乗算ゲート (*true multiplication gate*) を含む;

ここで, 真乗算ゲートは, ふたつの入力と共に定数ノードでない乗算ゲートのことである。Fearnley et al. [12] は, 連続探索問題の入力を *well-behaved arithmetic circuit* に制限することで, 計算量クラス FNP に属する計算問題を定式化した。前述したとおり, Daskalakis and Papadimitriou [9] による定式化に比べて, *well-behaved arithmetic circuit* を用いた定式化は, 次のふたつのアドバンテージを有する:

1. 余分な入力・出力を追加する必要がない;
2. 与えられた算術回路が, *well-behaved* であるかどうかは容易に検証できる。

## 3.2 不動点計算問題の難しさ

算術回路を用いた計算問題で最も興味深い問題は, 不動点計算問題 — 不動点定理で存在が保証された不動点を求める問題 — である。基本的に, 不動点計算問題は, 遷移関数  $f: [0, 1]^3 \rightarrow [0, 1]^3$  を計算する算術回路と近似パラメータ  $\varepsilon > 0$  を入力として受け取り,  $x \in [0, 1]^3$  such that  $\|f(x) - x\|_\infty \leq \varepsilon$  を出力する問題として定式化される。ただし, 完備性を満たすために幾つかの『違反』も会として認めている。たとえば, 定義 3.1 で紹介する計算問題 BROWER では, 連続性に反する点を回として認めている。

本章では, 不動点計算問題の難しさに関する研究の流れを紹介する。

### Brouwer の不動点定理

最も有名な不動点計算問題は, Brouwer の不動点計算問題である。これは, Brouwer の不動点定理 [4] によって存在が保証された不動点を実際に計算する問題である。形式的には次のように定式化される:

---

<sup>\*1</sup> サイズとは, 算術回路  $C$  を表現するために必要なビット長を意味する。このビット長には,  $C$  で使用される定数を表現するためのビットも含まれる。

### 定義 3.1. BROWER

#### Input:

- two parameters  $\varepsilon, \lambda > 0$ ;
- an arithmetic well-behaved arithmetic circuit computing  $f : [0, 1]^3 \rightarrow [0, 1]^3$ .

#### Task: Find one of the following:

- a point  $x \in [0, 1]^3$  such that  $\|f(x) - x\|_\infty \leq \varepsilon$ ;
- two points  $x, y \in [0, 1]^3$  such that  $\|f(x) - f(y)\|_\infty > \lambda \|x - y\|_\infty$ .

ここで、関数の連続性は Lipschitz 連続を用いて保証している。これは完備性を満たすために要求している条件である。Brouwer の不動点定理の主張と、解が満たすべき条件を比較すると計算問題 BROWER が完備探索問題であることが分かる。

Papadimitriou [20] は、計算問題 BROWER が PPAD 完全問題であることを示した。この計算問題は 1 次元では、容易に — 多項式時間で — 解くことができるが、2 次元では PPAD 完全問題となる [6].

Brouwer の不動点定理の難しさは、様々な問題の PPAD 完全性の証明に応用されている。たとえば、Nash 均衡計算問題の難しさが PPAD 完全であることは、Brouwer の PPAD 完全性に基づいて証明されている [6, 20].

### Banach の不動点定理（縮小写像の不動点定理）

続いて、Banach の不動点定理 [2] に関連する計算問題の難しさを紹介する。計算問題 BANACH は、計算問題 BROWER において、パラメータ（Lipschitz 定数） $\lambda$  が、 $0 < \lambda < 1$  を満たすことを要求した計算問題である。形式的な定義は下記のとおり：

### 定義 3.2. BANACH

#### Input:

- a parameter  $\varepsilon >$ ;
- a parameter  $0 < \lambda < 1$ ;
- an arithmetic well-behaved arithmetic circuit computing  $f : [0, 1]^3 \rightarrow [0, 1]^3$ .

#### Task: Find one of the following:

- a point  $x \in [0, 1]^3$  such that  $\|f(x) - x\|_\infty \leq \varepsilon$ ;
- two points  $x, y \in [0, 1]^3$  such that  $\|f(x) - f(y)\|_\infty > \lambda \|x - y\|_\infty$ .

Daskalakis and Papadimitriou [8] は、計算問題 BANACH が計算量クラス  $\text{PPAD} \cap \text{PLS}$  に属することを示した。しかし、この計算問題が完全問題であるかどうかは未解決である。より正確には、 $\ell_p$ -norm について、計算問題 BANACH が  $\text{PPAD} \cap \text{PLS}$  困難であるか、が未解決である [12]. 前述で、

$\ell_p$ -norm を持ち出したことには訳がある。計算問題 BANACH の入力において、距離を計算する算術回路を付加することで、この問題は  $\text{PPAD} \cap \text{PLS}$  完全問題となる [10, 13].

### Caristi の不動点定理

Caristi の不動点定理 [5] は、Banach の不動点定理の一般化として知られている。

#### 定義 3.3. CARISTI

##### Input:

- three parameters  $\varepsilon, \eta, \lambda > 0$ ;
- two arithmetic well-behaved arithmetic circuits computing  $f : [0, 1]^3 \rightarrow [0, 1]^3$  and  $\varphi : [0, 1]^3 \rightarrow [0, 1]$ .

##### Task: Find one of the following:

- a point  $x \in [0, 1]^3$  such that  $\|f(x) - x\|_\infty \leq \varepsilon$ ;
- a point  $x \in [0, 1]^3$  such that  $\eta\|x - f(x)\|_\infty > \varphi(x) - \varphi(f(x))$ ;
- two points  $x, y \in [0, 1]^3$  such that  $|\varphi(x) - \varphi(y)| > \lambda\|x - y\|_\infty$ .

計算問題 CARISTI では、遷移関数  $f : [0, 1]^3 \rightarrow [0, 1]^3$  に加えて、ポテンシャル関数  $\varphi : [0, 1]^3 \rightarrow [0, 1]$  も入力として受け取る。また、遷移関数  $f$  が Lipschitz 連続であることは、要求していない。一方で、ポテンシャル関数  $\varphi$  が Lipschitz 連続であることを要求する。Caristi の不動点定理 [5] から、上記の問題は完備探索問題である\*2。

Ishizuka [16] は、計算問題 CARISTI が PLS 完全問題であることを示した。計算量クラス PLS は、局所探索アルゴリズムで解くことが可能な計算問題全体の集合であり、Johnson et al. [17] によって導入された。[16] で示されるまで、不動点定理による計算量クラス PLS の特徴付けは知られていなかった。

### Brøndsted の不動点定理

Brøndsted の不動点定理 [3] は、Caristi の不動点定理の類似であり、同様に Banach の不動点定理の一般化である。Caristi の不動点定理では、ポテンシャル関数が Lipschitz 連続であることを要求した。Brøndsted の不動点定理では、ポテンシャル関数が連続であることは要求しない。一方で、遷移関数は連続であることを要求する。形式的には、次のように定式化する：

\*2 厳密には、Caristi の不動点定理のオリジナルの主張 [5] では、ポテンシャル関数が下半連続であればよい。Brouwer の不動点定理同様に、入力が要求に反していることを多項式時間で検証できることを保証するために、Lipschitz 連続であることを要求している

### 定義 3.4. BRØNDSTED

#### Input:

- two parameter  $\varepsilon, \eta, \lambda > 0$ ;
- two arithmetic well-behaved arithmetics circuit computing  $f : [0, 1]^3 \rightarrow [0, 1]^3$  and  $\varphi : [0, 1]^3 \rightarrow [0, 1]$ .

#### Task: Find one of the following:

- a point  $x \in [0, 1]^3$  such that  $\|f(x) - x\|_\infty \leq \varepsilon$ ;
- a point  $x \in [0, 1]^3$  such that  $\eta\|x - f(x)\|_\infty > \varphi(x) - \varphi(f(x))$ ;
- two points  $x, y \in [0, 1]^3$  such that  $\|f(x) - f(y)\|_\infty > \lambda\|x - y\|_\infty$ .

定義から明らかに、計算問題 BRØNDSTED は計算量クラス PPAD に属する。なぜなら、計算問題 BRØNDSTED は、計算問題 BROUWER を用いて解くことができる。また、 $\text{PPAD} \cap \text{PLS}$  困難問題であることも知られている [16]。しかし、どちらのクラスの完全問題となるかは未解決問題である。

### Tarski の不動点定理

最後に Tarski の不動点定理 [24] に関する計算問題の難しさを紹介する。Tarski の不動点定理も、前述の Caristi の不動点定理や Brøndsted の不動点定理と同様に、順序集合上の不動点定理である。これまで、算術回路に基づく不動点定理が続いたが、Tarski の不動点定理は、通常の Boolean 回路に基づく定義ができる。形式的には、次のように定める：

### 定義 3.5. TARSKI

#### Input:

- a Boolean circuit computing  $f : [N]^d \rightarrow [N]^d$ .

#### Task: Find one of the following:

- a point  $x \in [N]^d$  such that  $f(x) = x$ ;
- two points  $x, y \in [N]^d$  such that  $x \preceq y$  and  $f(x) \not\preceq f(y)$ .

ここで、 $x \preceq y$  は、すべての座標  $i \in [d]$  において、 $x_i \leq y_i$  が成り立つことを意味する。

Etessami et al. [11] は、計算問題 TARSKI が計算量クラス  $\text{PPAD} \cap \text{PLS}$  に属することを示した。しかし、 $\text{PPAD} \cap \text{PLS}$  困難であるかどうかは分かっていない。

## 4 おわりに

本稿では、不動点計算問題の難しさに関する最新の研究状況を紹介した。不動点計算問題に関する一連の研究の流れで、筆者の貢献は次のふたつである：

- 計算問題 CARISTI が PLS 完全問題であることを示した；
- 計算問題 BRØNDSTED が，計算量クラス PPAD に含まれ，計算量クラス PPAD  $\cap$  PLS 困難であることを示した。

これらの成果は，[16] に掲載されている。

現在（2021 年 12 月中旬），不動点計算問題の研究で抱えている未解決問題は以下の通りである：

- 計算問題 BANACH は，PPAD  $\cap$  PLS 完全問題であるか？
- 計算問題 BRØNDSTED は，PPAD 完全問題であるか，それとも PPAD  $\cap$  PLS 完全問題であるか？
- 計算問題 TARSKI は，PPAD  $\cap$  PLS 完全問題であるか，それとも計算量クラス EOPL に含まれるか？

最後に，本稿で紹介した不動点計算問題はいずれも『弱近似』を求める計算問題である。これらの計算問題とは別に，『強近似』を求める計算問題の難しさも考えられている。函数  $f : [0, 1]^3 \rightarrow [0, 1]^3$  に対する  $\varepsilon$ -弱近似とは， $\|f(x) - x\| \leq \varepsilon$  を満たす  $x \in [0, 1]^3$  である。一方で， $\varepsilon$ -強近似とは， $\|x^* - x\| \leq \varepsilon$  を満たす  $x \in [0, 1]^3$  である；ここで， $x^* \in [0, 1]^3$  は函数  $f$  の不動点である。すなわち， $f(x^*) = x^*$  を満たす。弱近似と強近似の違いについては，Yannakakis のサーベイ論文 [25] を参照することをおすすめする。

## 謝辞

筆者の研究の一部は，JSPS 科研費 21J10845 および JST，ACT-X，JPMJAX2101 の助成を受けたものである。

## 参考文献

- [1] S. Arora and B. Barak. Computational Complexity: A Mordern Approach. *Cambridge University Press*, 2009.
- [2] S. Banach: Sur les operations dans les ensembles abstraits et leur application aux equations integrales. *Fund. Math.*, Vol. 3, pp. 133—181, 1922.
- [3] A. Brøndsted. on a lemma of Bishop and Phelps. *Pac. J. Math.* Vol. 55(2), pp. 335–341, 1974.
- [4] L. E. J. Brouwer. Uber Abbildung von Mannigfaltikeiten. *Mathematicshe Annalen*, Vol. 71, pp. 97–115, 1912.
- [5] J. Caristi. Fixed point theorems for mapping satisfying inwardness conditions. *Trans. Am. Math. Soc.* Vol. 215, pp. 241–251, 1976.
- [6] C. Chen and X. Deng. On the complexity of 2D discrete fixed point problem. *Theor. Comput. Sci.* Vol. 410(44), pp. 4448–4456, 2009.
- [7] C. Daskalakis. Nash equilibria: Complexity, symmetries, and approximation. *Comput. Sci. Rev.*, Vol. 3, pp. 87–100, 2009.



- [8] C. Daskalakis and C. Papadimitriou. Continuous local search. In Proc. of *SODA*, ACM, pp. 790-804, 2011.
- [9] C. Daskalakis and C. Papadimitriou. Continuous local search — corrigendum. 2020.
- [10] C. Daskalakis, C. Tzamos, and M. Zampetakis. A converse to Banach’s fixed point theorem and its CLS-completeness. In Proc. of *STOC*, AMC, pp. 447–530, 2018.
- [11] K. Ettesami, C. Papadimitriou, A. Rubinstein, and M. Yannakakis. Tarski’s theorem, supermodular games, and the complexity of equilibria. In Proc. of *ITCS*, Vol. 151, pp. 18:1–18:19. 2020.
- [12] J. Fearnley, P. W. Goldberg, A. Hollender, R. Savani. The complexity of gradient descent:  $\text{CLS} = \text{PPAD} \cap \text{PLS}$ . In Proc. of *STOC*, ACM, pp. 46–59, 2021.
- [13] J. Fearnley, S. Gordon, R. Mehta, and R. Savani. CLS: new problems and completeness. *arXiv preprint*, 2017.
- [14] A. Filos-Ratsikas and P. W. Goldberg. The Complexity of Splitting Necklaces and Bisecting Ham Sandwiches. In Proc. of *STOC*, pp. 638–649, 2019.
- [15] P. W. Goldberg and P. H. Papadimitriou. TFNP: An Update. In Proc. of *CIAC*, pp. 3–9, 2021.
- [16] T. Ishizuka. On the complexity of finding a Caristi’s fixed point. *Inf. Process. Lett.*, Vol. 170, 2021.
- [17] D. S. Johnson, C. H. Papadimitriou, and M. Yannakakis. How easy is local search? *J. Comput Syst. Sci.*, Vol. 37(1), pp. 79–100, 1988.
- [18] S. Kintali, L. J. Poplawski, R. Rajaraman, R. Sundaram, and S.-H. Teng. Reducibility among Fractional Stability Problems. *SIAM J. Comput.*, Vol. 42(6), pp. 2063–2113, 2013.
- [19] N. Megiddo and C. H. Papadimitriou. On total fuctions, existence theorems and computational complexity. *Theor. Comput. Sci.* Vol. 81(2), pp. 317–324, 1991.
- [20] C. H. Papadimitriou. On the complexity of the parity argument and inefficient proofs of existence. *J. Comput. Syst. Sci.*, Vol. 48(3), pp. 498–532, 1994.
- [21] C. H. Papadimitriou. Computational Complexity. *Addison-Wesley*, 1994.
- [22] P. Schnider. The Complexity of Sharing a Pizza. In Proc. of *ISAAC*, 2021.
- [23] M. Sipser. Introduction to the theory of computation. *PWS Publishing Company*, 1997.
- [24] A. Tarski. A lattice-theoretical fixpoint theorem and its applictions. *Pac. J. Math.*, Vol. 5(2), pp. 285–309, 1955.
- [25] M. Yannakakis. Equilibria, fixed points, and complexity classes. *Comput. Sci. Rev.*, Vol. 3, pp. 71–85, 2009.