PCP定理の証明と応用

清水 伸高 (東京科学大学)

Contents

序文

このノートは, 京都大学大学院理学研究科数学・数理解析専攻数理解析系の数理解析特別講義Ⅰにおける集中講義「PCP 定理の証明と応用」の講義ノートである. 計算量 (computational complexity) とは, 問題を解くために必要な計算リソースの量 (例えば計算時間, 記憶領域のサイズ, 乱択や量子性の有無や量) を意味し, 計算量理論 (computational complexity theory) とはそれぞれの問題の計算量を明らかにするための理論である. PCP 定理とは, 判定問題 (Yes か No で答える問題) の検証に要する計算量に関する結果であり, 端的に言うと, ある命題が真であると主張する証明が文字列として与えられたとき, その証明を検証するためには, 通常, 全ての文字を見て確認する必要があるが, PCP 定理によれば, その証明の一部だけを見ることで, その命題が真であるか否かを確率的に検証することができるという驚くべき結果である. 例えば, ある実行列 A と実ベクトル b に対して線形方程式系 $Ax = b$ は解を持つ, という命題を考えてみよう. この命題が真であるならば実際に解の一つ x を証明として提示することができるが, その証明が正しいかどうかを検証するためには検証者は Ax を実際に計算し, その各成分が b と一致するかを確認する必要がある. ところが PCP 定理によれば, 巧妙に構成された証明 π を提示することにより, その証明 π 全ての文字を見ることなく, 99%の確率で正しく検証できるのである (ここでは確率的な検証, つまり検証者はランダムネスを用いた検証を行う設定を考える).

このように, 局所的な情報だけを使って全体の構造を推測できるという PCP 定理の性質は, 単に理論的に興味深いだけでなく, 誤り訂正符号の構成, 確率論的手法の脱乱択化, 最適化問題の近似率限界の導出など, 理論計算機科学において広大な応用を持ち, 現在でも計算量理論の中心的な研究対象の一つである. このような重要性を持つにもかかわらず, PCP 定理の証明に包括的に日本語でアクセスできる文献は (筆者が知る限り本資料の執筆時点では) 存在しない. 本講義では PCP 定理の証明を与えるとともに, その証明に用いられた計算量理論の基本的な概念や技法を解説し, PCP 定理の応用についても触れる. 僭越ながらもこれは PCP 定理の証明に日本語でアクセスする (おそらく国内初の) 資料であるといえる. 今回のような挑戦的な機会を与えてくださった河村彰星先生に感謝を申し上げる.

Chapter 1

導入

この集中講義では PCP 定理と呼ばれる計算量理論の基本的な結果について解説し, その証明を与える. PCP 定理は 1998 年に Arora and Safra [AS98] and Arora, Lund, Motwani, Sudan, and Szegedy [ALMSS98] によって証明された. この証明は代数的な手法に基づく誤り訂正符号を技巧的に組合せたものであり, 難解なものであったが, その後 Dinur [Din07] によってより簡潔な証明が与えられた. この講義では Dinur [Din07] による比較的簡単な証明を紹介する. ちなみに Dinur はのちにこの業績によりゲーデル賞を受賞している.

1.1 計算量理論の復習

まずは計算量理論のどの教科書にも載っているような基礎的な用語の定義を与える. これらの用語に明るい読者はセクション 1.2.2 から読み始めても良い. なお, このノートではアルゴリズムの定義 (チューリング機械の定義) は省略し, アルゴリズムについて述べる際は具体的な計算の手続きを述べる[1].

まずは基本的な記号の定義を与える:

- オーダー記法: 二つの関数 $f, g\colon \mathbb{N} \to \mathbb{N}$ に対し, $f(n) = O(g(n))$ であるとは, ある定数 $c > 0$ が存在して, 十分大きな全ての $n \in \mathbb{N}$ に対して $f(n) \leq cg(n)$ が成り立つことをいう. また, $f(n) = \Omega(g(n)), f(n) = o(g(n)), f(n) = \omega(g(n))$ なども同様に ($n \to \infty$ として) 定義する.

- 自然数 $n \in \mathbb{N}$ に対して $[n] = \{1, \ldots, n\}$ とする.

- 有限集合 S に対し, $x \sim S$ と書いたとき, x は S から一様ランダムに選ばれた元であることを意味する.

- 集合 S 上のベクトル $x \in S^n$ および $I \subseteq [n]$ に対し, $x_I \in S^I$ を, x の I への制限, すなわち, $x_I(i) = x(i)$ $(i \in I)$ と定義する.

[1] ひとまず Python や C 言語などで実装されたプログラムを考えれば良い. ただし, 計算機内では全ての数値は有限桁の二進数で表記されており, その読み書きや演算には少なくとも桁数に比例した計算時間がかかる. なお, $\sqrt{2}$ といった無理数は本来は有限桁で打ち切った近似値を扱うが, そのような小数はこの講義では扱わず, 特に断りのない限りは整数値のみを考える. また, 記憶領域へのアクセスは定数時間で行えると仮定する (チューリング機械であればテープの移動にかかる時間も考慮する).

- 本資料で登場する全ての有限集合 $S = \{s_1, \ldots, s_m\}$ に対し $s_1 < s_2 < \cdots < s_m$ という順序が暗に定まっているとする. 非空な部分集合 $T = \{t_1, \ldots, t_k\} \subseteq S$ を考える際は常にこの順序に従って $t_1 < \cdots < t_k$ を仮定する. 例えばグラフの頂点集合にも順序が一つ決まっており, 無向辺 $e = \{u, v\}$ を考える際も常に $u < v$ を仮定する.

- $\{0,1\}^* = \bigcup_{n\in\mathbb{N}} \{0,1\}^n$ を有限長の二進文字列全体とする.

- $x \in \{0,1\}^*$ に対して $|x|$ を x の文字数とする.

- アルゴリズム A に対し, $A(x)$ を入力 $x \in \{0,1\}^*$ に対するアルゴリズム A の出力とする. ここで, 単に「アルゴリズム」と言った場合は決定的アルゴリズムを指し, 乱択アルゴリズムについては明示的に言及する.

- 関数 $T(n)\colon \mathbb{N} \to \mathbb{N}$ を考える. 十分大きな全ての $n \in \mathbb{N}$ と全ての $x \in \{0,1\}^n$ に対して, A が $A(x)$ を出力するまでにかかる計算ステップ数の最大値が高々 $T(n)$ であるとき, アルゴリズム A の計算量は $T(n)$ であるという. 特に, ある (n に依らない) 定数 $c > 0$ が存在して計算量が $O(n^c)$ で抑えられるアルゴリズムを**多項式時間アルゴリズム**という[2]

- 計算量理論において慣例的に用いられる記法だが, 関数 $f(n)$ が n に関する多項式である, すなわち n に依存しないある定数 $c > 0$ に対して $f(n) = O(n^c)$ が成り立つことを $f(n) = n^{O(1)}$ または $f(n) = \mathsf{poly}(n)$ と表す. 例えば多項式時間アルゴリズムとは, 十分大きな全ての $n \in \mathbb{N}$ に対して, 長さ n の任意の文字列 $x \in \{0,1\}^n$ を受け取ったときの時間計算量が $n^{O(1)}$ で抑えられるアルゴリズムである.

- 二つの文字列 $x, y \in \{0,1\}^*$ を入力として受け取るアルゴリズムは $A(x, y)$ と表す. 三つ以上の場合も $A(x, y, z)$ などと表す.

計算量理論で最も基本的な問題群として判定問題と呼ばれる問題群がある.

定義 1.1.1 (判定問題)

部分集合 $L \subseteq \{0,1\}^*$ を**判定問題 (または言語)** といい, アルゴリズムに与える L の入力を**インスタンス**という. また, インスタンス $x \in \{0,1\}^*$ は $x \in L$ であるとき, 判定問題 L の **Yes インスタンス**といい, そうでない場合は **No インスタンス**という.
文字列 $x \in \{0,1\}^*$ に対し $L(x) \in \{0,1\}$ を, $x \in L$ かどうかの指示関数, すなわち $x \in L$ ならば $L(x) = 1$, そうでなければ $L(x) = 0$ と定義する.
アルゴリズム A は, 任意の $x \in \{0,1\}^*$ に対して $A(x) = L(x)$ が成り立つとき, A は L を解くという.

例 1.1.2 (グラフ連結性判定問題)

グラフ $G = (V, E)$ の隣接行列を $A \in \{0,1\}^{|V|\times|V|}$ とする. この行列を長さ $|V|^2$ の二進文字列として表現したものを $\mathsf{str}(A)$ とする. すなわち, $\mathsf{str}(A)$ の第 k ビットは,

[2] 計算モデルによって一つ一つの計算ステップの定義は異なるが, 原理的には 1bit の演算や記憶領域への読み書きの回数と思えばよい. 多項式時間で動くかどうかの議論であれば, 多くの古典的な計算モデルは等価である.

$i = \lceil k/|V| \rceil + 1, j = k \mod |V| + 1$ に対して $A(i,j)$ の値として与えられる. このとき,

$$L = \left\{ \mathsf{str}(A) \in \{0,1\}^{|V|^2} : A \text{ は連結グラフ } G \text{ の隣接行列} \right\}$$

は与えられたグラフが連結であるかどうかを判定する判定問題である.

この講義では「効率的に解ける」といった場合, 多項式時間アルゴリズムによって解けることを指す. そのような判定問題の集合をクラス P という.

定義 1.1.3 （クラス P）

判定問題 L は, それを解く多項式時間アルゴリズムが存在するとき, P に属するといい, $L \in \mathsf{P}$ と表す.

注釈 1.1.4 （入力のフォーマット）

例 1.1.2 では入力としてグラフの隣接行列を二進文字列として表現したものを考えたが, 一般にグラフの表現方法は他にも隣接リストなどが考えられる. しかし, 例えばグラフを隣接行列で表現するか隣接リストで表現するかは, それぞれのフォーマット間の変換が多項式時間で行えるため, **多項式時間で解けるかどうか**という点では等価である.
そのため, 厳密には問題を定義する際はその入力のフォーマット (グラフを隣接行列で表現するか, 隣接リストで表現するか, など) も指定する必要があるが, フォーマット間の変換が自明に多項式時間で行える場合に限ってはこの講義ではそのようなフォーマットの指定を省略し, 例えばグラフ連結性判定問題を「グラフが与えられたときにそれが連結であるかどうかを判定する問題」と表現する.

次に乱択アルゴリズムについて定義する. 端的に言えばアルゴリズムの内部でコイントスを行うものを乱択アルゴリズムという. ここでは明示的にランダムシードを受け取るアルゴリズムを乱択アルゴリズムと呼ぶことにする.

定義 1.1.5 （乱択アルゴリズム）

入力 $x \in \{0,1\}^*$ とは別にランダムシード (乱数表) と呼ばれる別の文字列 $s \in \{0,1\}^*$ を受け取るアルゴリズムを**乱択アルゴリズム**といい, ランダムシードであることを強調するために $A(x;s)$ などと表す. なお, 任意の $x, s \in \{0,1\}^*$ に対して $A(x;s)$ は有限時間で停止するとし, その計算量は $|x|$ のみに依存する関数で表せるとする. このとき, ランダムシード s の長さを常に A の計算量で上から抑える. すなわち, A の計算量が $T(n)$ であるとき, 十分大きな全ての $n \in \mathbb{N}$ と全ての $x \in \{0,1\}^n$ に対して A が読み込む s の文字数は高々 $T(n)$ であるため, $s \in \{0,1\}^{T(n)}$ であると仮定する. しばし, ランダムシード s を明記する必要が特にない場合は $A(x)$ と表す.
乱択アルゴリズムのランダムシードに関する確率, 期待値, 分散を議論する際は記号と

して $\Pr_A[\cdot], \mathbb{E}_A[\cdot], \mathbf{Var}_A[\cdot]$ を用いる. 乱択アルゴリズム A が判定問題 L を解くとは,

$$\Pr_A[A(x) = L(x)] \geq 2/3$$

が成り立つことをいう.

また, 入力とは別に文字列へのオラクルアクセスを受け取るアルゴリズムを考える.

定義 1.1.6 （オラクルアルゴリズム）

文字列 $\pi \in \{0,1\}^*$ に対し, π への**オラクルアクセス**を持つアルゴリズム $A^\pi(x)$ とは, 計算途中で π の指定された位置の文字を読むことができるアルゴリズムである. すなわち, π の i 番目の文字を読む操作を $A^\pi(x)$ の計算過程中に $O(\log|\pi|)$ 時間で行うことができるアルゴリズムである.[a] 同様に乱択オラクルアルゴリズムについても定義できる.

[a] 自然数 $i \in [|\pi|]$ を指定するために $O(\log|\pi|)$ ビットを定めなければならないため, $O(\log|\pi|)$ 時間を仮定している.

1.2 検証の計算量

数学全般における検証とは, ある命題が真であると主張する証明が与えられたとき, その証明が実際にその命題を正しく証明しているかどうかを確認することを意味する. 論文や記述試験の証明の査読や採点をイメージしてもらうとわかりやすいだろう. 計算量理論では検証やその計算量の議論は重要な研究テーマであり, その検証に要する計算量が議論される.

1.2.1 効率的な検証とクラス NP

判定問題 L と入力 $x \in \{0,1\}^*$ を与えられたとき, $x \in L$ かどうかを審議したい. ここで $x \in L$ を主張する証明が文字列 $\pi \in \{0,1\}^*$ で与えられたとする. このとき, **検証者**と呼ばれるアルゴリズムは x と π を読み込んで $x \in L$ かどうかを判定する. この判定を多項式時間で行えるとき, その判定問題 L の集合を NP という.

定義 1.2.1 （クラス NP）

判定問題 L は, 以下を満たす多項式時間アルゴリズム V と多項式 $p\colon \mathbb{N} \to \mathbb{N}$ が存在するとき, L は NP に属するという: アルゴリズム V は入力として $x, \pi \in \{0,1\}^*$ を受け取り, 0 または 1 を出力する.

1. もし $x \in L$ ならば, ある $\pi \in \{0,1\}^{p(|x|)}$ が存在して $V(x,\pi) = 1$ となる.
2. もし $x \notin L$ ならば, 全ての $\pi \in \{0,1\}^{p(|x|)}$ に対して $V(x,\pi) = 0$ となる.

また, このようなアルゴリズム V を **NP 検証者**といい, π を **NP 証明**という.

注釈 1.2.2 （クラス P と NP の関係）

判定問題 L が P に属するならば $L \in \mathsf{NP}$ である．実際, 受け取った $x \in \{0,1\}^*$ に対して $L(x)$ を計算してそれを出力する検証者を考えばよい．すなわち $\mathsf{P} \subseteq \mathsf{NP}$ である．一方, 逆側の包含関係 $\mathsf{NP} \subseteq \mathsf{P}$ が成り立つかどうかは P vs NP 問題と呼ばれる計算量理論における最も重要な未解決問題であり, 多くの研究者は $\mathsf{NP} \subseteq \mathsf{P}$ が成り立たないと信じている．

検証者 $V(x,\pi)$ が 1 を出力したとき, V は証明 π を**受理する**といい, 検証者が 0 を出力したとき, V は証明 π を**拒否する**という．

例 1.2.3 （合成数判定問題）

判定問題 $L = \{x \in \{0,1\}^* : x \text{ は合成数 }\}$ を考える．このとき, 検証者は x と π を読み込んで, $\pi \notin \{1, x\}$ かつ π が x を割り切るかどうかを判定する．もしも $x \in L$ である場合, 合成数なので非自明な約数を証明 π として与えれば $V(x,\pi) = 1$ となる．そうでない場合, 非自明な約数は存在しないため必ず $V(x,\pi) = 0$ となる．このアルゴリズム V は入力長 (つまり数値の二進表現したときのビット長) に関する多項式時間で動作するため, L は NP に属する．

例 1.2.4 （グラフ彩色問題）

自然数 $k \geq 2$ とグラフ $G = (V, E)$ に対し, 関数 $c : V \to [k]$ が全ての辺 $\{u, v\} \in E$ に対して $c(u) \neq c(v)$ を満たすとき, c を G の k-彩色といい, k-彩色が存在するようなグラフは k-彩色可能であるという．任意の $k \geq 2$ に対し, 判定問題

$$L = \{G \in \{0,1\}^* : G \text{ は } k\text{-彩色可能 }\}$$

は NP に属する．検証者は G と π を読み込んで, π が G の k-彩色であるかどうかを判定する．もしも $G \in L$ である場合, G は k-彩色可能であるため, k-彩色の証明 π を与えれば $V(G,\pi) = 1$ となる．そうでない場合, G は k-彩色可能でないため必ず $V(G,\pi) = 0$ となる．このアルゴリズム V は多項式時間で動作するため, L は NP に属する．

演習問題 1 （素数判定問題）

自然数 $n \in \mathbb{N}$ に対しその二進表記を $(n)_2 \in \{0,1\}^*$ と表す．判定問題 $\textsc{Primes} = \{(a)_2 \in \mathbb{N} : a \text{ は素数 }\}$ を考える．この問題は P に属することが知られている [AKS04] が, その複雑なアルゴリズムを用いずに初等的に $\textsc{Primes} \in \mathsf{NP}$ を示したい．そのために, 以下の事実を用いる:
任意の自然数 $a \in \mathbb{N}$ と $\gamma \in \{1, \ldots, a-1\}$ に対し, $\gamma^0, \gamma^1, \ldots \pmod a$ は周期的である．さらに, 以下が成り立つ:

- a が素数であるならば, ある $\gamma \in \{1, \ldots, a-1\}$ が存在して $\gamma^0, \gamma^1, \ldots \pmod a$ の周期が $a-1$ である[a].
- 一方, a が素数でないならば, 全ての $\gamma \in \{1, \ldots, a-1\}$ に対して $\gamma^0, \gamma^1, \ldots \pmod a$

の周期は $a-1$ 未満である. 特に, その周期 L は $a-1$ を割り切る.

これらの事実を用いて, 以下の小問に答えよ.

1. 次の検証者 V_1 を考える: 入力 $a \in \mathbb{N}$ と証明 $\gamma \in \{1,\dots,a-1\}$ に対し, $\gamma^0, \gamma^1, \dots, \gamma^{a-2} \pmod a$ を全て検証し, これら全て相異なるかどうかを判定する. この検証者 V_1 が多項式時間アルゴリズム**でない**理由を簡潔に説明せよ.
2. 入力 $a \in \mathbb{N}$ に対し, a が素数であることの証明として, 原始元[a] $\gamma \in \{1,\dots,a-1\}$ および $a-1$ の素因数分解 $a-1 = p_1^{\alpha_1}\cdots p_k^{\alpha_k}$ および各 p_i が素数であることの証明を再帰的に与える. この証明を用いて, 検証者 V_2 は a が素数であるかどうかを多項式時間で判定できることを示せ.

[a] このような γ を原始元という.

1.2.2 局所的な検証とクラス PCP

一般に $x \in L$ かどうかの検証では, 証明 π の全ての文字を読む必要がある. しかし, 証明 π のうちの一部の文字を読むだけで $x \in L$ かどうかを**確率的に**判定できる場合がある. そのような性質を持つ証明を**確率的検証可能な証明** (Probabilistically Checkable Proof, PCP) という.

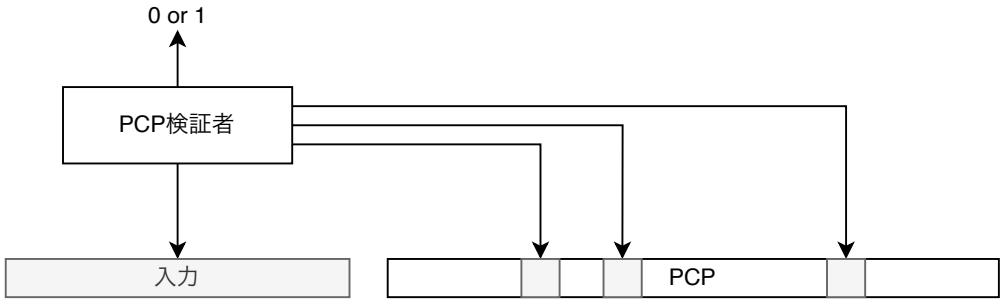

図 1.1: 確率的検証可能な証明の概念図. 検証者は乱数を用いて証明 π のうちの一部の文字のみを読み, それに基づいて判定を行う.

定義 1.2.5 (確率的検証可能な証明)

二つの関数 $r, q\colon \mathbb{N} \to \mathbb{N}$ に対し, $\mathsf{PCP}(r,q)$ を以下の性質を持つ判定集合 L の集合とする: ある多項式時間オラクル乱択アルゴリズム V が存在して, 任意の $x \in \{0,1\}^*$ に対し,

1. もし $x \in L$ ならば, ある $\pi \in \{0,1\}^*$ が存在して, (V の乱択に関して) 確率 1 で $V^\pi(x) = 1$ となる (**完全性**).
2. もし $x \notin L$ ならば, 全ての $\pi \in \{0,1\}^*$ に対して, (V の乱択に関して) 確率 $1/3$ 以上で $V^\pi(x) = 0$ となる (**健全性**).
3. さらに, 入力長が $n = |x|$ のとき, $V^\pi(x)$ はオラクル π のうち高々 $q(n)$ 個の文字

を読み, そのランダムシード長は $r(n)$ で抑えられる.

このようなオラクル乱択アルゴリズム V を **PCP 検証者**といい, 証明 π を **PCP** という.

注釈 1.2.6 （PCP の長さ）

$\mathsf{PCP}(r,q)$ の証明 π の長さは $q(n)2^{r(n)}$ で抑えられる. 各ランダムシード $s \in \{0,1\}^{r(n)}$ に対して検証者は π のうち高々 $q(n)$ 個の文字を読むため, 全てのランダムシードを列挙すると, アクセスされる可能性のある π の文字数は高々 $q(n)2^{r(n)}$ で抑えられる.

一般にランダムシード長 $r(n)$ と読み込む文字数 $q(n)$ が小さいほど良い PCP 検証者であると考えられる. PCP 検証者の構成は非常に難しい. 例えば例 1.2.4 のグラフ彩色問題に対する次の検証者を考えてみよう: 入力としてグラフ $G=(V,E)$ と証明として関数 $\pi\colon V \to [k]$ を受け取り, この関数 π が実際に G の k-彩色であるかどうかを判定する. 検証者は G の辺 $\{u,v\} \in E$ をランダムに一つ選び, $\pi(u) \neq \pi(v)$ であるかどうかによって判定する. この検証者は π のうち高々 $q(n)=O(1)$ 個の文字を読み, そのランダムシード長は $r(n)=O(\log n)$ で抑えられる. しかし, この検証者は健全性の条件を満たさない. 実際, G が k-彩色可能でない場合, どのような関数 $\pi\colon V \to [k]$ を与えても, 少なくとも一つの辺 $\{u,v\} \in E$ が存在して $\pi(u)=\pi(v)$ となるが, 検証者がこのような辺を引き当てる確率は最悪の場合, $1/|E|$ となるからである.

PCP 定理とは, ある $r(n)=O(\log n)$, $q(n)=O(1)$ に対して $\mathsf{PCP}(r,q)=\mathsf{NP}$ が成り立つことを主張する定理である. 例えばグラフ彩色問題は NP に属するため, 実は全段落の $r(n), q(n)$ を達成する PCP 検証者が存在するのである!

定理 1.2.7 （PCP 定理）

ある $r(n)=O(\log n)$, $q(n)=O(1)$ に対して $\mathsf{PCP}(r,q)=\mathsf{NP}$ が成り立つ.

注釈 1.2.8 （片側の包含関係）

PCP 定理において, $\mathsf{PCP}(r,q) \subseteq \mathsf{NP}$ は容易に示すことができる. 実際, 注釈 1.2.6 により, 証明 π の長さは $q(n)2^{r(n)}=n^{O(1)}$ で抑えられる. また, $r(n)=O(\log n)$ より, 検証者は $2^{r(n)}=n^{O(1)}$ 個のランダムシードを列挙し, それら全てに対して PCP 検証者を適用し, その出力値の多数決をとることで, 入力 x に対して $x \in L$ かどうかを多項式時間で判定できる. PCP 定理の証明の本質的な難しさは逆側の包含関係 $\mathsf{NP} \subseteq \mathsf{PCP}(r,q)$ の証明にある.

1.2.3 NP 完全性と Cook-Levin の定理

クラス NP に属する全ての問題に対しそれ以上に難しいという性質を **NP 困難性**という. そして NP に属する問題が NP 困難であるとき, その問題は **NP 完全**であるという. ここでは判定問題[3]に対して NP 困難性と NP 完全性の定義を与える.

[3]文脈によっては最適化問題や数え上げ問題といった, 判定問題ではない問題に対しても NP 困難性の概念が自然に定義されることもあり, この講義でも PCP 定理の応用を紹介する際に「最適化問題 X は NP 困難で

定義 1.2.9 （NP 完全性）

判定問題 $L \in \mathsf{NP}$ は, 任意の $L' \in \mathsf{NP}$ に対して以下を満たす決定的多項式時間アルゴリズム A が存在するとき, **NP 完全**であるという (A は L' に依存してよい): 文字列 $x \in \{0,1\}^*$ を入力として受け取ったアルゴリズム A の出力を $A(x) \in \{0,1\}^*$ とする. このとき, 任意の $x \in \{0,1\}^*$ に対して, $x \in L'$ と $A(x) \in L$ は同値である. また, このようなアルゴリズム A を L' から L への**カープ帰着**という.

注釈 1.2.10 （「NP 以上に難しい」の意味）

NP 完全な判定問題 L を解く多項式時間アルゴリズム A_0 が存在するならば, 任意の $L' \in \mathsf{NP}$ に対し L' を解く多項式時間アルゴリズムが存在する. 実際, L' から L へのカープ帰着を A とすると, 与えられた $x \in \{0,1\}^*$ が $x \in L$ かどうかはまず, $y = A(x)$ を計算し, その後 $A_0(y)$ を計算することで判定できる. この意味で L は L' 以上に難しいといえる.
しかし, 二つの問題の困難性を比較する際は必ずしもカープ帰着に拘る必要はない. 実際, 上述のアルゴリズムは L を解くアルゴリズム A_0 を一度だけ用いてしかもその出力 $A_0(y) \in \{0,1\}$ がそのまま $x \in L'$ かどうかの答えと一致しているが,「L が解けたら L' も解ける」ということを示すのであれば複数の入力 $y_1, \ldots, y_m$ に対して $A_0(y_1), \ldots, A_0(y_m)$ を計算してもよいし, それらの出力を組み合わせて $x \in L'$ かどうかを判定してもよい. このような自由度の高い操作を許しつつ A_0 が多項式時間で動くならば全体も多項式時間で動くことが保証されるような帰着を**チューリング帰着**という.

具体的にカープ帰着含め NP 完全性の枠組みが確立されたのは Karp [Kar72] であるが, 実はそれ以前の Cook [Coo71] は, カープ帰着とは少し異なる帰着を用いて**充足可能性判定問題 (SAT)** と呼ばれる問題が NP 完全であることを示していた. SAT は論理回路に関する問題だが, カープ帰着の下でも [Coo71] の証明は成立することを利用して Karp [Kar72] は最大クリーク問題, ハミルトン閉路問題, 彩色数の計算, ナップザック問題といった様々な自然な組合せ最適化問題の NP 困難性を証明していった. Karp [Kar72] が NP 完全性を示したこれらの問題は現在では **Karp の 21 の NP 完全問題** (Karp's 21 NP-complete problems) と呼ばれており, この成果によって Cook は 1982 年と Karp は 1985 年にそれぞれチューリング章を受賞している. また, 実は Levin [Lev73] も同様の結果を独立に示していたことが判明し (論文の発行当時は冷戦下であったことが災いして違いの認識が遅れてしまった), Levin は 2012 年にクヌース章を受賞している. このことから以下に定める充足可能性判定問題の NP 完全性を示す定理を **Cook-Levin の定理**といい, この定理は全ての NP 完全性の理論の基礎となる計算量理論における最も重要な定理の一つである.

定理 1.2.11 （Cook-Levin の定理）

論理回路 $C\colon \{0,1\}^n \to \{0,1\}$ を入力として受け取り, $C(x) = 1$ を満たす $x \in \{0,1\}^n$ が存在するかどうかを判定する問題を**充足可能性判定問題** (SAT; satisfiability problem) という. 充足可能性判定問題 (SAT) は NP 完全である.

ある」という言い方を用いる箇所がある. その際は最適化問題 X を解く多項式時間アルゴリズムを用いると全ての NP に属する問題が多項式時間で解けるということを意味する.

Cook-Levin の定理の証明はクラス NP の検証者を非決定的チューリング機械で模倣し, その機械を回路として表現することによって与えられるが, それにはチューリング機械や論理回路の定義なども必要になってしまい本講義の本筋から離れていってしまうため割愛する. なお, 文脈によっては上記の定義における充足可能性判定問題を**回路充足可能性判定問題** (CSAT; Circuit SAT) と呼ぶこともある.

Cook-Levin の定理を用いると 3 彩色問題 3COL の NP 完全性を示すことができる. 具体的には 3-SAT と呼ばれる NP 完全な問題を 3COL に帰着させることで 3COL の NP 完全性を示せる (本講義では割愛).

定理 1.2.12 （3 彩色問題の NP 完全性）

3 彩色問題 3COL は NP 完全である. すなわち, 任意の $L \in \mathsf{NP}$ に対して, 多項式時間アルゴリズム f が存在して, 任意の $x \in \{0,1\}^*$ に対して $x \in L$ と $f(x) \in \mathrm{3COL}$ は同値である.

定理 1.2.12 により, PCP 定理を証明するには, 3COL に対する効率的な PCP 検証者を構成すればよいことがわかる.

定理 1.2.13 （3 彩色問題の PCP 検証者）

ある $r(n) = O(\log n), q(n) = O(1)$ に対して, 3 彩色問題 3COL は $\mathsf{PCP}(r, q)$ に属する.

演習問題 2

定理 1.2.12 と 1.2.13 を仮定して, PCP 定理 (定理 1.2.7) を証明せよ.

1.3 PCP 定理の応用

PCP 定理の応用として, 様々な組合せ最適化問題に対する近似の NP 困難性を導出することができる.

1.4 PCP 定理の背景

PCP 定理は組合せ最適化の分野への著しい応用を持つため, その視点から PCP 定理を紹介する文献ではなぜ PCP 定理が発見されたのかという動機が語られることが少ないように思われる. そこで, この節では PCP という計算量クラスを考えるに至った背景について軽く触れたい. PCP 定理は 1992 年に Arora と Safra によって証明された. その後, 2005 年に Dinur によって証明が簡略化され, 2010 年に Arora と Barak によって証明が再構成された.

Chapter 2

局所的な検証と弱いPCP定理の証明

この章はPCP検証者における「少ないクエリ回数」という要件がなぜ達成できるかについて説明することを目的とする. そのため, まずはPCP定理の要件であるランダムビットの短さはひとまず無視した以下の形の「弱いPCP定理」を証明する: 任意のNPに属する判定問題が$O(1)$クエリかつ$\mathsf{poly}(n)$ビットのランダムネスを用いて確率的に検証可能である.

2.1 局所的な検証

本講義では, 検証者が証明の文字列πのうち$O(1)$個の文字を読み込んで検証を行うことを**局所的な検証**と呼ぶことにする. 一般的な感覚として, ある主張が成り立つことを示す証明を検証する際, **冗長に表現で記述しない限り**その証明を全て読み込まなければ検証できないと思われる (全ての文字に本質的な意味があるならば全てを確認しなければならないはずであろう). では逆に, 証明を冗長に表現することで局所的な検証を可能にすることはできるだろうか?

2.1.1 線形方程式の局所的な検証 (1/2)

興味深いことに, 非常に冗長性が大きい記述で証明が与えられたならば局所検証が可能であることを, 線形方程式の検証を例に説明する.

定義 2.1.1 (線形方程式)

有限体$\mathbb{F}$上の行列$M \in \mathbb{F}^{m\times n}$とベクトル$z \in \mathbb{F}^m$に対して

$$My = z$$

を満たす$y \in \mathbb{F}^n$が存在するかどうかを判定する問題を LinEq とする. ここで$|\mathbb{F}| = q$はm, nとは独立な定数であるとする (従って有限体上の演算は$O(1)$時間で計算できると仮定する).

愚直な検証者として$y \in \mathbb{F}^n$を証拠として受け取り, $My = z$を確認することで検証を行うことを考える (実際の計算機では全てを二進文字列で表現するため, $\mathbb{F}$の元は$\lceil \log_2 |\mathbb{F}| \rceil = O(1)$ビットで表現されている). 明らかにこの検証者は$My$を計算するために$y$の全ての成分 (す

なわち $O(n)$ ビット) を読み込む必要があるため, 局所的な検証者とはならない. もちろん, この判定問題は標準的な線型方程式の解の存在性判定であるため, 掃き出し法を用いれば証拠へのオラクルアクセスなしでも多項式時間で解ける. しかしここではあえて, 局所的な検証の構成例を与えるという目的で取り扱う.

重要な性質として, 二つのベクトル $a, b \in \mathbb{F}^n$ に対する次の性質を考える (証明は演習問題 3):

$$\Pr_{r\sim\mathbb{F}^n}\left[r^\top a \neq r^\top b\right] = \begin{cases} 0 & \text{if } a = b \\ 1 - \frac{1}{q} & \text{if } a \neq b. \end{cases} \tag{2.1}$$

さて, 式 (2.1) の性質を用いると一様ランダムな $r \sim \mathbb{F}^n$ に対して

$$r^\top My = r^\top z \tag{2.2}$$

かどうかを確認することで, $My = z$ かどうかを確率的に検証できる. 実際, $My = z$ ならば確率 1 で検証者は受理し, $My \neq z$ ならば確率 $1 - 1/q$ で検証者は拒否する (何度も繰り返せばこの拒否率を任意に 1 に近い定数まで近づけることができる). ベクトル z は入力として与えられているため式 (2.2) の右辺は $O(n)$ 時間で計算できる. 一方で, ベクトル y は証拠として与えられるため, $r^\top My$ の計算は愚直に考えると $O(mn)$ 時間かかる上に y の全ての成分を読み込む必要がある. ところが, 全てのベクトル $w \in \mathbb{F}^n$ に対して $y^\top w \in \mathbb{F}$ を連結して得られる長さ q^n の文字列 $(y^\top w)_{w\in\mathbb{F}^n}$ を証拠 π として与えることにより, 自身で $r^\top M$ を計算した後に $r^\top My = y^\top(M^\top r)$ の値を**一文字の証拠の読み込み**で計算できるのである.

演習問題 3 （ランダムなベクトルとの内積）

式 (2.1) を証明せよ.

このように, ベクトル $y \in \mathbb{F}^n$ を, それを係数とする線形関数 $w \mapsto y^\top w$ の**真理値表 (全ての点での評価値を並べた文字列) として冗長に表現する**ことで, 証拠の読み込み回数を定数に抑えるというのが基本的なアイデアである. このように文字列を冗長に表現する手法を総称して**誤り訂正符号**という. 本講義では, **アダマール符号**と呼ばれる以下の誤り訂正符号を用いる:

定義 2.1.2 （アダマール符号）

有限体 $\mathbb{F}$ 上のベクトル $y \in \mathbb{F}^n$ に対して, 長さ q^n の文字列

$$\mathrm{Had}_n(y) = (y^\top w)_{w\in\mathbb{F}^n}$$

を考える. 集合 $\mathrm{Had}_n = \{\mathrm{Had}_n(y) \colon y \in \mathbb{F}^n\}$ を**アダマール符号**といい, その元を**アダマール符号の符号語**という. また, $\mathrm{Had}_n(y)$ を y **をアダマール符号で符号化した文字列**という.
なお, 次元 n が明らかなときは省略して $\mathrm{Had}(y)$ などと表す.

注釈 2.1.3

本講義では文字列, ベクトル, 関数をしばし同一視する. すなわち, ベクトル $x \in \mathbb{F}^n$ を単に文字列と呼ぶこともあり, または関数 $x\colon [n] \to \mathbb{F}$ とみなすこともある. 例えば $y \in \mathbb{F}^n$ をアダマール符号で符号化した文字列 $\mathrm{Had}(y)$ は, 長さ q^n の文字列とみなすこともあれば, 関数 $\mathrm{Had}(y)\colon \mathbb{F}^n \to \mathbb{F}$ とみなすこともある.
これは, アダマール符号の解析の過程では $\mathrm{Had}(y)$ を関数とみなして扱う方が都合が良いものの, PCP 定理の証明の過程では $\mathrm{Had}(y)$ を文字列として扱い, 証明 π として与えることになるからである.

素朴な証拠 y をアダマール符号で符号化したものを証明として扱うというアイデアを素朴に実装すると, LinEq に対する局所的な検証 (の候補) として以下の検証者を考えることができる:

アルゴリズム 2.1.4 （LinEq に対する局所的な検証?）

1. 入力として M, z を受け取り, 証拠として $\pi \in \mathbb{F}^{q^n}$ へのオラクルアクセスを受け取る.
2. 一様ランダムな $r \sim \mathbb{F}^m$ を選択し, $r^\top M$ と $r^\top z$ を計算する.
3. 証拠 $\pi\colon \mathbb{F}^{q^n}$ を関数 $\pi\colon \mathbb{F}^n \to \mathbb{F}$ として解釈し, オラクルアクセスを用いて $a = \pi(M^\top r)$ を求める.
4. $a = r^\top z$ ならば 1 を出力し, そうでなければ 0 を出力する.

上記の検証者は PCP 検証者の性質を持つだろうか? まず, (M, z) が Yes インスタンスであるとき, ある $y \in \mathbb{F}^n$ が存在して $My = z$ が成り立つ. このとき, $\pi = \mathrm{Had}(y)\colon \mathbb{F}^n \to \mathbb{F}$ とすると, $\pi(M^\top r) = y^\top (M^\top r) = r^\top (My) = r^\top z$ となるため, 検証者は確率 1 で受理する.

一方で (M, z) が No インスタンスであるときに検証者が拒否する確率はどうなるだろうか? PCP 検証者であることを示す (cf. 定義 1.2.5) には, 任意の $\pi \in \mathbb{F}^n \to \mathbb{F}$ に対して拒否確率は少なくとも $1/3$ 以上でなければならない. これは示せるのであろうか? 証明 π がアダマール符号の符号語である (すなわち, ある $y \in \mathbb{F}^n$ に対して $\pi = \mathrm{Had}(y)$ と表せる) 場合は, 式 (2.2) の性質および (M, z) が No インスタンスであることから $My \neq z$ より, 少なくとも確率 $1 - 1/q$ で検証者は拒否する. しかしながら, 一般の $\pi \in \mathbb{F}^n \to \mathbb{F}$ はこのような線形関数として表現できるとは限らず, そのような π に対しても拒否しなければならない.

2.1.2 線形性テスト

与えられた (M, z) が No インスタンスであるときに任意の $\pi \in \mathbb{F}^{q^n}$ を定数確率で拒否するために, アルゴリズム 2.1.4 を以下のように修正する: 証拠 π がアダマール符号の符号語ならば, アルゴリズム 2.1.4 のステップ 2-4 を実行し, そうでないならば 0 を出力する. ここで新たに一つの問題が生じる: どのようにして π の線形性を局所検証すればよいだろうか?

一般にオラクルアクセスで与えられた関数 $\pi\colon \mathbb{F}^n \to \mathbb{F}$ が線形関数かどうかを厳密に確認

するには,

$$\forall x, y \in \mathbb{F}^n, \quad \pi(x+y) = \pi(x) + \pi(y) \tag{2.3}$$

が成り立つかどうかを確認する必要があり, q^n 回のオラクルアクセスが必要となってしまう.

そこで,「線形関数であるかどうか」ではなく「線形関数に近いかどうか」を局所検証することを考える.

定義 2.1.5 （関数同士の近さ）

二つの関数 $f, g\colon A \to B$ に対して $\mathrm{dist}(f,g)$ を

$$\mathrm{dist}(f,g) = \Pr_{a\sim A}[f(a) \neq g(a)]$$

と定義し, $\mathrm{dist}(f,g) \leq \delta$ を満たすとき, f は g に **δ-近い** (δ-close) という.
また, 関数クラス $\mathcal{F} \subseteq \{g\colon A \to B\}$ および関数 $f\colon A \to B$ に対して

$$\mathrm{dist}(f,\mathcal{F}) = \min_{g\in\mathcal{F}} \mathrm{dist}(f,g)$$

と定義し, $\delta(f,\mathcal{F}) \leq \delta$ であるとき, f は $\mathcal{F}$ に δ-近いという.

関数 $f\colon A \to B$ をベクトル $f \in B^A$ と同一視すると, $\mathrm{dist}(f,g)$ はベクトル f, g 間の正規化されたハミング距離 (ℓ_0 ノルム) に一致する.

線形関数全体 $\mathrm{Had} \subseteq \{\mathbb{F}^n \to \mathbb{F}\}$ を考える (定義 2.1.2). オラクルアクセスとして与えられた関数 $\pi\colon \mathbb{F}^n \to \mathbb{F}$ が Had に近いかどうかを検証する局所検証者が構成できる Blum, Luby, and Rubinfeld [BLR93].

定理 2.1.6 （線形性テスト）

以下を満たす乱択多項式時間オラクルアルゴリズム $A^\pi(1^n)$ が存在する[a]: 関数 $\pi\colon \mathbb{F}^n \to \mathbb{F}$ がオラクルアクセスとして与えられたとき,

- $\pi \in \mathrm{Had}$ ならば $A^\pi(1^n)$ は確率 1 で受理する.
- $\mathrm{dist}(\pi, \mathrm{Had}) \geq 0.01$ ならば $A^\pi(1^n)$ は確率 0.99 で拒否する.

[a] アルゴリズム A は 1^n を入力として与えられているため, 多項式時間であることは計算量が $\mathrm{poly}(n)$ で抑えられることを意味する.

注釈 2.1.7 （符号の局所検査性）

定理 2.1.6 はアダマール符号 $\mathrm{Had} \subseteq \mathbb{F}^{q^n}$ は, オラクルアクセスとして与えられた文字列 $\pi \in \mathbb{F}^{q^n}$ がその符号語か, もしくは Had から遠いかどうかを, $O(1)$ 回のオラクルアクセスで確立的に判定できることを意味する. このような性質を持つ誤り訂正符号を**局所検査可能符号** (locally testable code) という.

証明. アルゴリズム A^π は非常に単純で, 線形関数であることの特徴づけ式 (2.3) をランダムな点で確認するというものである. 具体的には以下で与えられる:

アルゴリズム 2.1.8 （線形性テスト）

1. オラクルアクセスとして関数 $\pi\colon \mathbb{F}^n \to \mathbb{F}$ を受け取り, 入力として 1^n を受け取る.
2. 一様ランダムに $x, y \sim \mathbb{F}^n$ を選び, $\pi(x+y) \neq \pi(x) + \pi(y)$ が成り立つならば拒否する.
3. 十分大きな定数 $K \in \mathbb{N}$ に対し, ステップ 2 を K 回繰り返す. この繰り返しの中で一度も拒否しなければ, 受理する.

実際, $\pi \in \mathrm{Had}$ ならば式 (2.3) が成り立つため, $A^\pi(1^n)$ は確率 1 で受理する. 一方で, $\mathrm{dist}(\pi, \mathrm{Had}) \geq 0.01$ であるときに拒否確率を下から抑えたい. これは以下の主張から従う:

主張 2.1.9 （線形性テストの拒否確率）

任意の関数 $\pi\colon \mathbb{F}^n \to \mathbb{F}$ に対して

$$\Pr_{x,y\sim\mathbb{F}^n}[\pi(x+y) \neq \pi(x) + \pi(y)] \geq \frac{1}{6} \cdot \mathrm{dist}(\pi, \mathrm{Had}).$$

主張 2.1.9 は後で証明する. 仮定より $\mathrm{dist}(\pi, \mathrm{Had}) \geq \delta := 0.01$ なので, ステップ 3 の定数 K を十分大きくすることによって, アルゴリズム 2.1.8 の拒否確率を 0.99 より大きくすることができる. □

次に主張 2.1.9 を示す. 実際にはより一般的な次の補題を証明する.

補題 2.1.10 （準同型性テスト）

有限アーベル群 G, H に対し, 写像 $f\colon G \to H$ が準同型であるとは, 任意の $x, y \in G$ に対して $f(x+y) = f(x) + f(y)$ が成り立つことをいい, G から H への準同型な写像の全体を $\mathcal{H}$ と表す. このとき, 任意の関数 $f\colon G \to H$ に対して

$$\Pr_{x,y\sim G}[f(x+y) \neq f(x) + f(y)] \geq \frac{1}{12} \cdot \mathrm{dist}(f, \mathcal{H}).$$

証明. 記号の簡単のため, $\rho_f := \Pr_{x,y\sim G}[f(x+y) \neq f(x) + f(y)]$ と表す. また, 各 $y \in G$ に対し, 写像 $v_y\colon G \to H$ を $v_y(x) = f(x+y) - f(y)$ とし, $v\colon G \to H$ を

$$v(x) = \mathrm{argmax}_{a\in H}\left\{\Pr_{y\sim G}[v_y(x) = a]\right\}$$

で定める (タイが存在する場合は任意). すなわち, $v(x)$ は $(v_y(x))_{y\in G}$ の中での多数決, すなわち最も出現頻度の高い値として定める. 証明は以下の三つの主張を組み合わせることによって得られる.

主張 2.1.11

$\mathrm{dist}(f, v) \leq 2\rho_f$ が成り立つ.

証明. 定義より $\rho_f = \Pr_{x,y}[f(x)+f(y) \neq f(x+y)] = \Pr_{x,y}[f(x) \neq v_y(x)]$ である. また, $v(x) \neq h \Rightarrow \Pr_y[v_y(x) = h] \leq 1/2 \iff \Pr_y[v_y(x) \neq h] \geq 1/2$ が成り立つ. 実際, $(v_y(x))_{y \in G}$ の中で多数決をとったときに h が選ばれなかったということは, 過半数に至らなかったことを意味するからである. 特に, $h = f(x)$ とすると, $\mathbf{1}_{v(x) \neq f(x)} \leq 2 \Pr_y[v_y(x) \neq f(x)]$. 両辺の $x \sim G$ に関する期待値をとると

$$\operatorname{dist}(f, v) = \Pr_x[v(x) \neq f(x)] \leq 2 \Pr_{x,y}[v_y(x) \neq f(x)] = 2\rho_f.$$

となり主張を得る. □

主張 2.1.12

$\rho_f < 1/6$ ならば, 全ての $x \in G$ に対して $\Pr_y[v_y(x) = v(x)] > 2/3$.

証明. 任意に $x \in G$ を固定し, 確率変数 Φ を, 一様ランダムに $y \sim G$ を選び $\Phi = v_y(x)$ として定める. 我々の目標は, ある $a \in H$ に対して $\Pr[\Phi = a] > 2/3$ が成り立つことを示すことである. Φ_1, Φ_2 を Φ の独立なコピーとする (固定する x は同一). Φ がある値をとる傾向にあることを示すために, まずは $\Phi_1 = \Phi_2$ が成り立つ確率が大きいことを示す.

$$\begin{aligned}
\Pr[\Phi_1 = \Phi_2] &= \Pr_{y_1,y_2 \sim G}[v_{y_1}(x) = v_{y_2}(x)] \\
&= \Pr_{y_1,y_2}[f(x+y_1) - f(y_1) = f(x+y_2) - f(y_2)] \\
&= \Pr_{y_1,y_2}[f(x+y_1) - f(x+y_2) = f(y_1) - f(y_2)] \\
&\geq 1 - 2\rho_f \\
&> 2/3.
\end{aligned}$$

最初の不等式では, $\Pr_{y_1,y_2}[f(x+y_1) - f(x+y_2) \neq f(y_1 - y_2)] \leq \rho_f$ および $\Pr_{y_1,y_2}[f(y_1) - f(y_2) \neq f(y_1 - y_2)] \leq \rho_f$ を用いた. ここで, $2/3 < \Pr[\Phi_1 = \Phi_2] = \sum_{a \in H} \Pr[\Phi = a]^2 \leq \max_a \Pr[\Phi = a] \cdot \sum_{a \in H} \Pr[\Phi = a] = \max_a \Pr[\Phi = a]$ より主張を得る. □

主張 2.1.13

全ての $x \in G$ に対して $\Pr_y[v_y(x) = v(x)] > 2/3$ が成り立つならば, $v \in \mathcal{H}$, すなわち h は準同型である.

証明. 全ての $x, y \in G$ に対して $f(x) + f(y) = f(x+y)$ が成り立つことを示せばよい. 任意に $x, y \in G$ を固定する. 一様ランダムな $z \sim G$ を選ぶ. このとき, 仮定より

- $\Pr_z[v_z(x) \neq v(x)] < 1/3$,
- $\Pr_z[v_{z-y}(y) \neq v(y)] < 1/3$,
- $\Pr_z[v_{z-y}(x+y) \neq v(x+y)] < 1/3$

従って, ユニオンバウンドより, 任意の $x, y \in G$ に対してある $z \in G$ が存在して次の三つが同時に成り立つ:

- $v(x) = v_z(x) = f(x+z) - f(z)$,
- $v(y) = v_{z-y}(y) = f(z) - f(z-y)$,
- $v(x+y) = v_{z-y}(x+y) = f(x+z) - f(z-y)$

これらの等式から

$$v(x) + v(y) - v(x+y) = f(x+z) - f(z-y) - (f(x+z) - f(z-y)) = 0.$$

より主張を得る. □

最後に補題 2.1.10 の証明を完成させる. $\rho_f < 1/6$ ならば, 主張 2.1.12 と 2.1.13 より, $v \in \mathcal{H}$ である. さらに主張 2.1.11 より,

$$\rho_f \geq \frac{\mathrm{dist}(f, v)}{2} \geq \frac{\mathrm{dist}(f, \mathcal{H})}{2}$$

である. 一方, そうでなければ $\rho_f \geq 1/6 \geq \mathrm{dist}(f, \mathcal{H})/6$ である. いずれにせよ, $\rho_f \geq \mathrm{dist}(f, \mathcal{H})/6$ が成り立つ. □

以上より, 与えられた π が Had に近いかどうかを局所検証できることを示した. では, 仮に π が Had に近いとし, 最も近い線形関数を $f \in \mathrm{Had}$ としよう. このような $f \in \mathrm{Had}$ は一意である (演習問題 4). このとき, 99%の $x \sim \mathbb{F}^n$ に対して $\pi(x) = f(x)$ が成り立つ. 一方でアルゴリズム 2.1.4 では $f(M^\top r)$ を計算する必要があり, 一般に $r \sim \mathbb{F}^m$ に対して $M^\top r$ の分布は一様ランダムとは限らない. そこで, 任意の点 $x \in \mathbb{F}^n$ に対して $f(x)$ を計算することが必要であり, 次の補題はこれが可能であることを示している.

補題 2.1.14 (最も近い線形関数の任意点での評価)

関数 $\pi\colon \mathbb{F}^n \to \mathbb{F}$ が $\mathrm{dist}(\pi, \mathrm{Had}) \leq \varepsilon$ を満たすとし, π に最も近い線形関数を $f \in \mathrm{Had}$ とする. このとき, 任意の $x \in \mathbb{F}^n$ を入力として受け取り, π への 2 回のオラクルアクセスを用いて $O(n)$ 時間で $f(x)$ を確率 $1 - 2\varepsilon$ で出力する乱択オラクルアルゴリズム $A^\pi(x)$ が存在する.

証明. 入力として与えられた $x \in \mathbb{F}^n$ に対して, $A^\pi(x)$ はランダムな $r \sim \mathbb{F}^n$ を選び, $\pi(x+r) - \pi(r)$ を出力する. 明らかに時間計算量は $O(n)$ であり, $\Pr_r[\pi(x+r) \neq f(x+r)], \Pr_r[\pi(r) \neq f(r)]$ はそれぞれ ε 以下である. 従って, 確率 $1-2\varepsilon$ で出力値は $\pi(x+r)-\pi(r) = f(x+r)-f(r) = f(x)$ となる. □

演習問題 4 (最も近い線形関数の唯一性)

関数 $\pi\colon \mathbb{F}^n \to \mathbb{F}$ が $\mathrm{dist}(\pi, \mathrm{Had}) \leq 0.01$ を満たすとき, $\mathrm{dist}(\pi, f) \leq 0.01$ を満たす $f \in \mathrm{Had}$ は唯一存在することを示せ.

2.1.3 線形方程式の局所的な検証 (2/2)

ここまでの議論を用いて LinEq に対する局所的な検証者を構成する.

アルゴリズム 2.1.15 （LinEq に対する局所的な検証）

1. 入力として M, z を受け取り, 証拠として $\pi \in \mathbb{F}^{q^n}$ へのオラクルアクセスを受け取る.
2. 定理 2.1.6 のアルゴリズムを π をオラクルとして実行し, $\mathrm{dist}(\pi, \mathrm{Had}) \geq 0.01$ ならば拒否して終了する.
3. 一様ランダムな $r \sim \mathbb{F}^m$ を選択し, $r^\top M$ と $r^\top z$ を計算する.
4. 補題 2.1.14 のアルゴリズムを $M^\top r$ を入力, π をオラクルとして実行し, その出力を a とする.
5. $a \neq r^\top z$ ならば拒否する.
6. ステップ 3-5 を 3 回繰り返し, 一度も拒否しなければ受理して終了する.

アルゴリズム 2.1.4 の検証者と比較すると, まず, 証拠として与えられた関数 π が線形関数であることを検証するためのステップが追加されている. また, ステップ 4 では $\pi(M^\top r)$ の代わりに補題 2.1.14 のアルゴリズムが用いられている. これまでの議論から, アルゴリズム 2.1.15 は LinEq に対する局所的な検証者であることがわかる. ステップ 6 は単に成功確率を増幅させるための繰り返しである.

定理 2.1.16 （線形方程式の局所的な検証）

アルゴリズム 2.1.15 の検証者を $V^\pi(M, z)$ とする. このとき, 任意の $M \in \mathbb{F}^{m \times n}, z \in \mathbb{F}^m$ に対して以下が成り立つ:

- $V^\pi(M, z)$ の π へのオラクルアクセスの回数は $O(1)$ である.
- $My = z$ を満たす $y \in \mathbb{F}^n$ が存在するならば, $\pi = \mathrm{Had}(y)$ としたとき, $V^\pi(M, z)$ は確率 1 で受理する.
- $My = z$ を満たす $y \in \mathbb{F}^n$ が存在しないならば, 任意の $\pi \colon \mathbb{F}^n \to \mathbb{F}$ に対して $V^\pi(M, z)$ は確率 $2/3$ で拒否する.

証明. オラクルアクセスの回数は明らかに $O(1)$ である. また, $My = z$ を満たす $y \in \mathbb{F}^n$ が存在するならば, $\pi = \mathrm{Had}(y)$ としたとき, 確率 1 で受理する. 一方, $My = z$ を満たす $y \in \mathbb{F}^n$ が存在しないときに検証者が拒否する確率を考える. オラクル π に関して以下の二つのケースを考える:

ケース 1: π が線形関数に 0.01-近いとき. このとき, $V^\pi(M, z)$ がステップ 5 で拒否する確率を評価する. オラクル π に最も近い線形関数を $f \colon \mathbb{F}^n \to \mathbb{F}$ とする. このとき, 補題 2.1.14 (入力を $M^\top r$, $\varepsilon = 0.01$ とする) より, ステップ 4 で計算される a は, 確率 $1 - 2\varepsilon = 0.98$ で

$f(M^\top r)$ と等しい. ここで, 線形関数 f が $f(x) = x^\top y$ と表せるとすると, 拒否確率は

$$\begin{aligned}
\Pr\left[a \neq r^\top z\right] &\geq \Pr_r\left[a \neq r^\top z \mid a = f(M^\top r)\right] \cdot \Pr_r\left[a = f(M^\top r)\right] \\
&\geq \Pr_r\left[r^\top My \neq r^\top z\right] \cdot 0.98 \\
&\geq \left(1 - \frac{1}{q}\right) \cdot 0.98 && \because \text{式 (2.2) および } My \neq z \\
&\geq 0.4 && \because q \geq 2
\end{aligned}$$

を満たす. これを 4 回繰り返すため, 拒否確率は少なくとも $1 - (1 - 0.4)^3 \geq \frac{2}{3}$ を満たす.

ケース 2: π が線形関数に 0.01-遠いとき. このとき, 定理 2.1.6 より, $V^\pi(M, z)$ はステップ 2 において少なくとも確率 0.99 で拒否する.

従って, どちらのケースでも, $V^\pi(M, z)$ は少なくとも確率 2/3 で拒否する.

□

2.2 弱い PCP 定理とその証明

定理 2.1.16 のアイデアを拡張して PCP 定理を証明するには何が必要だろうか?

- まず, LinEq は実際には多項式時間で解ける判定問題であるが, 実際には何かしらの NP 完全な判定問題に対して定理 2.1.16 のような検証者を構成しなければならない.
- 次に, 定理 2.1.16 の検証者 $V^\pi(M, z)$ は, ランダムに $r, r' \sim \mathbb{F}^n$ を選ぶ時点で $\Omega(n)$ ビットのランダムネスを必要とする. 実際, 証拠として与えられた関数 π を文字列として表現するとその長さは q^n に比例するため, その文字列のランダムなインデックスを指定しようとすると必然的に $\Omega(n)$ ビットのランダムネスが必要である. そこで証明 π の長さを指数的な長さ $2^{\Theta(n)}$ から多項式 $n^{O(1)}$ に抑える必要がある (このとき, インデックスの指定に必要なビット数は $O(\log n)$ で抑えられる).

本節では前者の問題を解決し, 以下の弱い PCP 定理を証明する.

定理 2.2.1 (弱い PCP 定理)

ある $r(n) = \mathsf{poly}(n), q(n) = O(1)$ に対して, $\mathsf{NP} \subseteq \mathsf{PCP}(r, q)$ が成り立つ.

我々の最終的な目標である PCP 定理 (定理 1.2.7) では, PCP 検証者のランダムビットの長さが $r(n) = O(\log n)$ であることを要求しているため, $r(n)$ に関しては定理 2.2.1 は指数的に大きい値になってしまっているが, 読み込む証明の文字数は n に依存しない定数で抑えられるという点では同一である. なお, 定理 2.2.1 は後に定理 1.2.7 の証明でも用いられる.

2.2.1 方針

NP 完全な判定問題として 3 彩色問題 3COL を考え, ある $r = O(1), q = \mathsf{poly}(n)$ に対して $\mathrm{3COL} \in \mathsf{PCP}(r, q)$ を示す. すると, 定理 2.2.1 の証明は演習問題 2 と同じ議論から得られる.

3 彩色問題 3COL のインスタンス $G = (V, E)$ を考える. 以降, $\mathbb{F}_3$ を $\mathbb{F}$ と略記する. これが Yes インスタンスであることと, ある $x = (x(u))_{u \in V} \in \mathbb{F}^V$ が存在して

$$\forall e = \{u, v\} \in E, \quad (x(u) - x(v))^2 = 1 \tag{2.4}$$

を満たすことは同値である. 従って, 二次方程式系式 (2.4) に対する所望の PCP 検証者を構成すればよい. まずはこの二次方程式系を線型方程式系として表現するために, 以下の行列 $M \in \mathbb{F}^{E \times V^2}$ を定義する:

$$M(e, (u, v)) = \begin{cases} 1 & \text{if } u = v \in e, \\ -2 & \text{if } e = \{u, v\}, \\ 0 & \text{otherwise.} \end{cases} \tag{2.5}$$

ベクトル $x \in \mathbb{F}^V$ に対して $y \in \mathbb{F}^{V^2}$ を $y(u, v) = x(u) \cdot x(v)$ と定義する (特に $u = v$ ならば $y(u, u) = x(u)^2$ である). このとき, 式 (2.4) は $My = \mathbf{1}$ と表現できる. ここまでの議論から, $G = (V, E)$ が Yes インスタンスであることと, ある $x \in \mathbb{F}^V, y \in \mathbb{F}^{V^2}$ が存在して

条件 1 全ての $u, v \in V$ に対し $y(u, v) = x(u) \cdot x(v)$.

条件 2 $My = \mathbf{1}$.

の両方を満たすことは同値である. 方針としては, この二つの条件を検証する PCP 検証者を構成して組み合わせることによって, 上記の問題に対する検証者を構成する. 条件 2 に関しては, 定理 2.1.16 の検証者を用いて $V^{\pi}(M, \mathbf{1})$ を実行すればよい. この検証者はオラクルアクセスの回数が $O(1)$ でランダムビット長が $O(|V^2|) = O(n^2)$ である.

2.2.2 証明の構成

PCP 検証者が受け取る証明 π は, 二つの関数 $\pi_y \colon \mathbb{F}^{V^2} \to \mathbb{F}$ と $\pi_x \colon \mathbb{F}^V \to \mathbb{F}$ からなる. 具体的には, π_y と π_x を表すそれぞれ長さ q^{V^2} と q^V の文字列を結合して得られる文字列を π と定める (従って π の文字列としての長さは $q^{V^2} + q^V$ である). 検証者は, **条件 1** と**条件 2** を満たす $x \in \mathbb{F}^V, y \in \mathbb{F}^{V^2}$ に対し, これらをアダマール符号で符号化したものを $\pi_y \colon \mathbb{F}^{V^2} \to \mathbb{F}$ と $\pi_x \colon \mathbb{F}^V \to \mathbb{F}$ として受け取ることを想定している. 従って, これらが実際に x, y を符号化したものであることと, 確かに (x, y) が**条件 1** と**条件 2** を満たすことを検証することになる.

定理 2.1.16 の検証者 V^{π} を用いる ($z = \mathbf{1}$ とする) と**条件 2** を定数回のオラクルアクセスで確率的に検証できる. また, π_y と π_x がそれぞれ線形関数に近いかどうかは定理 2.1.6 より, 定数回のオラクルアクセスで確率的に検証できる. 従って, π_y, π_x がそれぞれ線形関数に非常に近いことを仮定した上で**条件 1** を検証する PCP 検証者を構成すればよい.

2.2.3 テンソル積の検証

我々の目標は, ある二つのベクトル $y \in \mathbb{F}^{V^2}$ と $x \in \mathbb{F}^V$ に対して, $\mathrm{dist}(\pi_y, \mathrm{Had}(y)) \leq 0.01$ と $\mathrm{dist}(\pi_x, \mathrm{Had}(x)) \leq 0.01$ となることが保証されている二つのオラクル π_y と π_x への定数回のオラクルアクセスを用いて, その x, y が**条件 1** を満たすことを検証することである. なお, そのような x, y が存在するならばそれらは一意であることが演習問題 4 から保証されている. ここで, $y \in \mathbb{F}^{V^2}$ を $V \times V$-行列として表現したものを $Y \in \mathbb{F}^{V \times V}$, $X = xx^{\top}$ とする. **条件 1** は $Y = X$ を満たすことと同値である.

補題 2.2.2 （テンソル積の検証）

ある $x \in \mathbb{F}^V, y \in \mathbb{F}^{V^2}$ に対し, $\mathrm{dist}(\pi_y, \mathrm{Had}_{n^2}) \leq 0.01$ と $\mathrm{dist}(\pi_x, \mathrm{Had}_n) \leq 0.01$ となる二つのオラクル π_y と π_x への定数回のオラクルアクセスを用いて, その x, y が全ての $u, v \in V$ に対し $y(u,v) = x(u) \cdot x(v)$ を満たすことを検証する PCP 検証者 $V^{\pi_x,\pi_y}(1^n)$ が存在する.
すなわち, $Y = (y(u,v))_{u,v \in V} \in \mathbb{F}^{V \times V}$ と $X = xx^\top$ に対し,

- $Y = X$ ならば確率 1 で $V^{\pi_x,\pi_y}(1^n)$ は受理する.
- $Y \neq X$ ならば確率 0.99 で $V^{\pi_x,\pi_y}(1^n)$ は拒絶する.

方針としては, 式 (2.1) の考え方を行列に拡張することである.

証明. 任意の行列 $A, B \in \mathbb{F}^{n \times n}$ に対し

$$\Pr_{r_1,r_2 \sim \mathbb{F}^n}\left[r_1^\top X r_2 \neq r_1^\top Y r_2\right] = \begin{cases} 0 & \text{if } X = Y, \\ 1 - \frac{1}{q} & \text{if } X \neq Y. \end{cases} \tag{2.6}$$

が成り立つ. π_x, π_y へのオラクルアクセスを用いて式 (2.6) の両辺を計算し, その結果を比較することで $Y = X$ かどうかを確認できる.

まず, $X = xx^\top$ であるため, 任意の $r_1, r_2 \in \mathbb{F}^V$ に対し

$$\begin{aligned} r_1^\top X r_2 &= \sum_{u,v \in V} r_1(u) r_2(v) x(u) x(v) \\ &= \sum_{u \in V} r_1(u) x(u) \sum_{v \in V} r_2(v) x(v) \\ &= \mathrm{Had}_n(x)(r_1) \cdot \mathrm{Had}_n(x)(r_2) \end{aligned}$$

である. 特に, $\mathrm{dist}(\pi_x, \mathrm{Had}_n(x)) \leq 0.01$ より, 一様ランダムに $r_1, r_2 \sim \mathbb{F}^n$ を選んだとき, 確率 0.98 で $\mathrm{Had}_n(x)(r_1) = \pi_x(r_1)$, $\mathrm{Had}_n(x)(r_2) = \pi_x(r_2)$ となる. 従って式 (2.6) の左辺は π_x への 2 回のオラクルアクセスを用いて確率 0.98 で計算できる.

次に $r_1^\top Y r_2$ を計算する. 任意の $r_1, r_2 \in \mathbb{F}^V$ に対し, $\bar{r} \in \mathbb{F}^{V^2}$ を, (u,v) 番目の文字が $r_1(u) r_2(v)$ であるような文字列とする. このとき

$$\begin{aligned} r_1^\top Y r_2 &= \sum_{u,v \in V} r_1(u) r_2(v) y(u,v) \\ &= \mathrm{Had}_{n^2}(y)(\bar{r}) \end{aligned}$$

である. また, $\mathrm{dist}(\pi_y, \mathrm{Had}_{n^2}) \leq 0.01$ より, 補題 2.1.14 を $\varepsilon = 0.01$, $f = \mathrm{Had}_{n^2}(y)$, $\pi = \pi_y$ として適用すると, 確率 0.98 で $\mathrm{Had}_{n^2}(y)(\bar{r})$ を計算できる. すなわち, 式 (2.6) の右辺は π_y への 2 回のオラクルアクセスを用いて確率 0.98 で計算できる.

まとめると, PCP 検証者 $V^{\pi_x,\pi_y}(1^n)$ は, 以下のアルゴリズムとして記述される:

アルゴリズム 2.2.3 （テンソル積の検証）

1. 入力として 1^n を受け取り, π_x, π_y へのオラクルアクセスを受け取る.
2. 一様ランダムな $r_1, r_2 \sim \mathbb{F}^V$ を選択する.
3. オラクルアクセスを用いて $\pi_x(r_1), \pi_x(r_2)$ を計算する.
4. 補題 2.1.14 を用いて, $\mathrm{Had}_{n^2}(y)(\bar{r})$ を計算する. ここで $\bar{r} \in \mathbb{F}^{V^2}$ は, (u, v) 番目の文字が $r_1(u)r_2(v)$ であるような文字列である.
5. $\pi_x(r_1) \cdot \pi_x(r_2) \neq \mathrm{Had}_{n^2}(y)(\bar{r})$ ならば拒否する.
6. ステップ 2-5 を 10 回繰り返し, 一度も拒絶しなければ受理して終了する.

ステップ 3,4 では確率 0.96 で式 (2.6) の両辺を計算でき, さらに $X \neq Y$ ならば ($r_1, r_2 \sim \mathbb{F}^n$ を選ぶランダムネスに関して) 確率 $1 - 1/q$ で拒絶できる. 従って, 一回の試行で少なくとも確率 $0.96 - \frac{1}{q} \geq 0.4$ ($\because q \geq 2$) で拒絶できる. この試行を十分大きい定数 (10 回で十分) だけ繰り返すことによって, $X \neq Y$ である時の拒絶確率は任意に増幅できる. □

2.2.4 弱い PCP 定理の証明

ここまでの議論を用いて弱い PCP 定理を証明する. 以下の補題を必要とする.

補題 2.2.4 （線型方程式の解の検証）

あるベクトル $y \in \mathbb{F}^n$ に対し, $\mathrm{dist}(\pi, \mathrm{Had}(y)) \leq 0.01$ を満たすことが保証されているオラクル π への定数回のオラクルアクセスを用いて, 入力として与えられた $M \in \mathbb{F}^{m \times n}, z \in \mathbb{F}^m$ に対し, 以下を満たす乱択 $O(mn)$ 時間オラクルアルゴリズム $V^\pi(M, z)$ が存在する:

- $My = z$ ならば $V^\pi(M, z)$ は確率 1 で受理する.
- $My \neq z$ ならば $V^\pi(M, z)$ は確率 0.99 で拒絶する.

演習問題 5 （線型方程式の解の検証）

補題 2.2.4 を証明せよ (ヒント: 式 (2.2) と補題 2.1.14 を用いる).

定理 *2.2.1* の証明. NP 完全問題である 3 彩色問題 3COL に対し, 所望の PCP 検証者を構成する. 体 $\mathbb{F} = \mathbb{F}_3$ を考える. インスタンス $G = (V, E)$ が Yes インスタンスであることと, ある $x, y \in \mathbb{F}^V$ が存在して, 式 (2.5) で定義される行列 $M \in \mathbb{F}^{E \times V^2}$ に対し, **条件 1** と**条件 2** を満たすことは同値である. 以下では, この条件を検証する PCP 検証者を記述する.

アルゴリズム 2.2.5 （3 彩色問題の PCP 検証者 $V^{\pi_x,\pi_y}(G)$）

1. 入力として $G=(V,E)$ および証明として $\pi_x\colon \mathbb{F}^V \to \mathbb{F}$ と $\pi_y\colon \mathbb{F}^{V^2} \to \mathbb{F}$ へのオラクルアクセスを受け取る.
2. 入力に対し式 (2.5) で定義される行列 $M \in \mathbb{F}^{E\times V^2}$ を計算する.
3. 定理 2.1.6 のアルゴリズムを π_x と π_y それぞれに対して (オラクルとして用いて) 実行し, それぞれが線形関数に 0.01-近いかどうかを検証する. もしいずれかの実行においてこのアルゴリズムが拒否したら拒否する.
4. 補題 2.2.2 の検証者を, オラクルとして π_x と π_y を用いて実行する. この検証者が拒否したら拒否する.
5. 補題 2.2.4 の検証者を, 入力として $(M,\mathbf{1})$, オラクルとして π_y を用いて実行する. この検証者が拒否したら拒否する.
6. ここまでのステップで拒否していなければ受理して終了する.

この検証者 $V^{\pi_x,\pi_y}(G)$ が所望の PCP 検証者であることを確認する. 定理 2.1.6 と 2.1.16 と補題 2.2.2 と 2.2.4 ではそれぞれオラクルアクセスは $O(1)$ 回であり, 内部で用いるランダムビットは $O(n^2)$ である.

入力 $G=(V,E)$ が 3COL の Yes インスタンスであるとする. このとき, ある $x \in \mathbb{F}^V$ が存在して, $y(u,v)=x(u)\cdot x(v)$ で定まる $y \in \mathbb{F}^{V^2}$ に対して $My=\mathbf{1}$ を満たす. この x,y に対して, それぞれ $\pi_x=\mathrm{Had}_n(x)$ と $\pi_y=\mathrm{Had}_{n^2}(y)$ とする. このとき

- ステップ 3 では, π_x,π_y はどちらも線形関数であるため, 拒否する確率は 0 である.
- ステップ 4 では, 実際に全ての $u,v \in V$ に対して $y(u,v)=x(u)x(v)$ が成り立つため, 拒否する確率は 0 である.
- ステップ 5 では, 実際に y は $My=\mathbf{1}$ の解であるため, 拒否する確率は 0 である.

以上より, $V^{\pi_x,\pi_y}(G)$ は確率 1 で受理する.

次に $G=(V,E)$ が 3COL の No インスタンスであるとする. 任意の $\pi_x\colon \mathbb{F}^V \to \mathbb{F}$ と $\pi_y\colon \mathbb{F}^{V^2} \to \mathbb{F}$ を考える.

- $\mathrm{dist}(\pi_x,\mathrm{Had}_n) \geq 0.01$ または $\mathrm{dist}(\pi_y,\mathrm{Had}_{n^2}) \geq 0.01$ ならば, $V^{\pi_x,\pi_y}(G)$ はステップ 3 において確率 0.99 で拒絶する. 以降では $\mathrm{dist}(\pi_x,\mathrm{Had}_n) < 0.01$ かつ $\mathrm{dist}(\pi_y,\mathrm{Had}_{n^2}) < 0.01$ とし, π_x,π_y に最も近い線形関数をそれぞれ $\mathrm{Had}_n(x),\mathrm{Had}_{n^2}(y)$ とする.
- ステップ 4 は, 補題 2.2.2 の仮定が満たされているため, 確率 0.99 で拒絶する. 以降ではさらに $x \in \mathbb{F}^V, y \in \mathbb{F}^{V^2}$ が, 全ての $u,v \in V$ に対し $y(u,v)=x(u)x(v)$ を満たすことが保証されているとする.
- ステップ 5 は, G が No インスタンスであることから, $My \neq \mathbf{1}$ であるため, 補題 2.2.4 の検証者が確率 0.99 で拒絶する.

以上より, $V^{\pi_x,\pi_y}(G)$ が一度のステップ 3-5 のループで拒絶する確率は少なくとも 0.97 である. □

Chapter 3

制約充足問題

制約充足問題 (Constraint Satisfaction Problem, CSP) は, 計算量理論において重要な問題の一つであり, PCP 定理の証明においても中心的な役割を果たす.

3.1 制約充足問題の定義

制約充足問題とは端的に言えば連立方程式の解の存在性判定を問う判定問題である.

定義 3.1.1 （制約充足問題）

制約充足問題 (CSP) とは次の要素からなる組 $\varphi = (X, \Sigma, \mathcal{I}, \mathcal{C})$ を入力とする判定問題である:

- **アルファベット**: 有限集合 Σ.
- **変数列**: $X = (\mathrm{x}_1, \ldots, \mathrm{x}_n)$.
- 制約の引数の列: $\mathcal{I} = (I_1, \ldots, I_m)$. ここで, 各 I_i は $I_i \neq \emptyset$ かつ $I_i \subseteq [n]$ を満たす.
- **制約の列**: $\mathcal{C} = (c_I)_{I \in \mathcal{I}}$. ここで, 各 c_I は $c_I \subseteq \Sigma^{|I|}$ を満たす.

各 $\mathrm{x} \in X$ を**変数**, 各 $c_I \in \mathcal{C}$ を**制約**という.
入力 $(X, \Sigma, \mathcal{I}, \mathcal{C})$ は, ある変数への割り当て $a\colon X \to \Sigma$ が存在して, 任意の $I = \{i_1, \ldots, i_k\} \in \mathcal{I}$ (ただし $i_1 < \cdots < i_k$) について,

$$(a(\mathrm{x}_{i_1}), \ldots, a(\mathrm{x}_{i_k})) \in c_I$$

であるとき, かつその時に限り Yes インスタンスである.
全ての $i \in [m]$ について $|I_i| \leq q$ であるとき, この CSP は **q-CSP** という.
また, 固定した割り当て $a\colon X \to \Sigma$ に対する φ の**不満足値**を

$$\mathrm{UNSAT}(a; \varphi) = \Pr_{\{i_1, \ldots, i_k\} \sim \mathcal{I}} [(a(\mathrm{x}_{i_1}), \ldots, a(\mathrm{x}_{i_k})) \notin c_I]$$

とする. ここで $I \sim \mathcal{I}$ は I が $\mathcal{I}$ から一様ランダムに選ばれたことを意味する. さらに, 全ての割り当てに関して不満足値の最小値

$$\mathrm{UNSAT}(\varphi) = \min_{a\colon X \to \Sigma} \mathrm{UNSAT}(a; \varphi)$$

を φ の**不満足値**という.

各制約 $c_I \in \mathcal{C}$ は, その制約が充足される割り当ての集合を表す. 例えば $c_I = \Sigma^{|I|}$ であるならば, 任意の割り当てに対して c_I は充足されるし, $c_I = \emptyset$ であるならば, どの割り当ても c_I を充足しない.

例 3.1.2 (グラフ彩色問題)

グラフ彩色問題 (例 1.2.4) は 2-CSP である. 実際, グラフ $G = (V, E)$ に対して

- 変数列を $X = V$ とする.
- アルファベットを $\Sigma = [k]$ とする.
- 各制約 c_e は各辺 $e = \{u, v\} \in E$ に付随していて,

$$(a_u, a_v) \in c_e \iff a_u \neq a_v$$

と定義する.

このとき, グラフ G が k-彩色可能であることと, この CSP が Yes インスタンスであることは同値である.

3.1.1 PCP との関係

ランダムシード長 $r = r(n)$ かつクエリ数 $q = O(1)$ の PCP 検証者は $\Sigma = \{0, 1\}$ の場合の q-CSP によって表現でき, 逆に q-CSP は PCP 検証者として表現できる. 実際, q クエリの PCP 検証者 $V^\pi(x)$ を考えよう. 証明の長さを $|\pi| = \ell$ とする. 入力 x とランダムシード s を固定したときの $V^\pi(x; s)$ が証明中の読み込む文字のインデックスの集合を $I_s \subseteq [\ell]$ とする (ここで $|I_s| \leq q$). このとき, $V^\pi(x; s)$ は $\{0, 1\}^{I_s}$ を $\{0, 1\}$ に写す関数を定める. この関数を制約 c_s とみなすことで, 検証者 $V^\pi(x)$ は q-CSP のインスタンスとして表現できる. このとき, q-CSP のインスタンスの変数集合は証明 π に対応する. もし $x \in L$ であるならば, 確率 1 で $V^\pi(x) = 1$ となるような π が存在する. つまり, 全てのランダムシード s に対して $V^\pi(x; s) = 1$ となるような π が存在するため, 先ほど構成した q-CSP のインスタンスは Yes インスタンスである. 逆に $x \notin L$ であるならば, 全ての π に対して確率 $1/3$ 以上で $V^\pi(x) = 0$ となる. これは, q-CSP インスタンスに対して, 全ての割り当てを考えても, 全体の制約のうち少なくとも $1/3$ の割合は充足されない (すなわち, UNSAT の値が $1/3$ 以上となる) ことを意味する.

この対応関係に基づいて, PCP 定理を CSP を用いた言葉で表すことができる.

PCP 検証者	q-CSP
PCP π	割り当て
ランダムネスを固定した時の判定	CSP の制約
PCPπ を拒否する確率	割り当ての不満足値 UNSAT(π)

Table 3.1: PCP と CSP の対応関係

定理 3.1.3 （PCP 定理の CSP 版）

ある関数 $m = n^{O(1)}$, $q = O(1)$, $\ell = n^{O(1)}$, 定数 $c \in \mathbb{N}$, $\varepsilon > 0$, および次の性質を満たす多項式時間決定的アルゴリズム A が存在する: 3 彩色問題のインスタンス $G = (V, E)$ を入力として受け取り, G の頂点数を n としたとき, A は高々 $\ell(n)$ 個の変数と $m(n)$ 個の制約および要素数 c のアルファベットからなる q-CSP のインスタンス φ を出力する. さらにこのインスタンス φ は

- 入力 G が 3COL の Yes インスタンスであるとき, UNSAT(φ) $= 0$ となる (すなわち φ は Yes インスタンス).
- 入力 G が 3COL の No インスタンスであるとき, UNSAT(φ) $\geq \varepsilon$ となる.

補題 3.1.4

定理 3.1.3 と定理 1.2.13 は同値である.

証明. それぞれの方向を別々に証明する.

定理 3.1.3⇒ 定理 1.2.13 の証明. ある $r = O(\log n)$, $q' = O(1)$ に対して, 3COL に対するシード長 r, クエリ回数 q' の PCP 検証者 V^π を構成する. 入力としてグラフ $G = (V, E)$ を受け取り, 定理 3.1.3 のアルゴリズム A を用いて q-CSP のインスタンス φ を出力する. また, PCP π はこの q-CSP のインスタンス φ の割り当てとして解釈し, PCP 検証者 V^π は以下の操作を十分大きな定数回繰り返す: 一様ランダムに制約 c_i を選択し, その制約に含まれる変数に対する割り当て π の値を読み込み, この制約が充足**されない**ならば 0 を出力し終了する. 何度も繰り返した末に終了しなかったのであれば, 1 を出力して終了する. この検証者は繰り返しの回数が $O(1)$ であり, それぞれの繰り返しにおいては高々 q 個の変数の値を読み込むため, クエリ回数は $q' = O(q) = O(1)$ となる. なお, ここでアルファベットサイズ c が定数であることに留意する (実際には PCP π は二進文字列なので, 変数割り当てを読み込む際には $\log_2 c$ 文字を読み込んでいる). また, ランダムシードはランダムな制約を選ぶために使われてるため, その長さは $O(\log m) = O(\log n)$ となる.

もしグラフ G が Yes インスタンスならば, φ も Yes インスタンスであるため, π をその充足割り当てとすれば, 全ての制約 c_i が充足されるため, (制約の選び方のランダムネスに関して) 確率 1 で $V^\pi(G) = 1$ となる. もしグラフ G が No インスタンスならば, UNSAT(φ) $\geq \varepsilon$ である. 従って, 任意の割り当て π に対して, 一様ランダムな制約 c_i が充足される確率は高々 $1 - \varepsilon$ である. よって, この操作を $\lceil 10/\varepsilon \rceil = O(1)$ 回繰り返すと, 少なくとも確率 1/3 で充足されない制約が一度以上選ばれ, 検証者は 0 を出力する.

定理 1.2.13⇒ 定理 3.1.3 の証明. 仮定より, 3COL に対する, ランダムシード長 $r = O(\log n)$, クエリ回数 $q = O(1)$ の PCP 検証者 V^π が存在する. アルゴリズム A は, 入力 G に

対して, 全てのランダムシード $s \in \{0,1\}^r$ を列挙して, それぞれの $V^\pi(G;s)$ を関数 c_s とみなして, これらを制約とする CSP を出力する. 各 $V^\pi(G;s)$ は, π を変数とみなしたとき, 高々 q 個の変数の値を読み込むため, $(c_s)_{s\in\{0,1\}^r}$ は $2^r = n^{O(1)}$ 個の制約からなる q-CSP となる. なお, V^π は多項式時間アルゴリズムなので, A も多項式時間アルゴリズムである. □

従って, 以降は定理 3.1.3 の証明に注力する.

3.1.2 制約グラフ

PCP 定理の証明は, NP-完全な 2-CSP である 3 彩色問題のインスタンスからスタートし, このインスタンスを適当な q-CSP にうまく変換することによって与えられる. この変換の記述を容易にするために, 2-CSP のインスタンスをグラフとして表現する方法として, **制約グラフ**の概念を導入する.

定義 3.1.5 （制約グラフ）

2-CSP のインスタンス $\varphi = (X, \Sigma, \mathcal{I}, \mathcal{C})$ に対し, 以下で定まる組 $G = \langle (V, E), \Sigma, \mathcal{C}' \rangle$ を**制約グラフ**という[a]:

- (V, E) はグラフである. ただし頂点集合は $V = X$ であり, 辺集合は $E = \mathcal{I}$ である. なお, ここで考えるグラフは多重辺や自己ループを持ちうるもの[b]とし, 特に $|I_i| = 1$ の場合は対応する辺は自己ループとする.
- アルファベット Σ.
- 制約の列 $\mathcal{C}' = (c'_e)_{e\in E}$ は, $|e| = 2$ ならば $c'_e = c_e$ とし, $|e| = 1$ ならば $c'_e = \{(u,u) \in \Sigma^2 : u \in e\}$ とする. これにより, 全ての c'_e は Σ^2 の部分集合となる.

制約グラフ $G = \langle (V, E), \Sigma, \mathcal{C} \rangle$ および割り当て $a : V \to \Sigma$ に対して, その**不満足値**を

$$\mathrm{UNSAT}(a; G) = \Pr_{e\sim E}[a(e) \notin c_e]$$

と定義し (ここで $e = \{u, v\}$ $(u < v)$ に対して $a(e) = (a(u), a(v)) \in \Sigma^2$ とする), G の不満足値を

$$\mathrm{UNSAT}(G) = \min_{a\in\Sigma^V} \mathrm{UNSAT}(a; G)$$

と定義する.
制約グラフ $G = \langle (V, E), \Sigma, \mathcal{C} \rangle$ に対して, その**サイズ**を

$$\mathrm{size}(G) = |V| + |E|$$

と定義する.

[a] 制約グラフの表記に用いる括弧は形式的には本来 $((V, E), \Sigma, \mathcal{C}')$ のようにすべきだが, 可読性のためあえて外側の括弧を $\langle\cdot\rangle$ としている.
[b] 具体的には E は多重集合であり, 自己ループに対応する辺は $\{u\}$ と表す. 詳細は定義 3.2.1 を参照.

注釈 3.1.6 （入力長とサイズの関係）

本講義で考える制約グラフのアルファベットサイズ $|\Sigma|$ は常に定数, すなわち $|V|$ や $|E|$ に依存しない値であるとする. この仮定の下, 制約グラフ $G = \langle (V, E), \Sigma, \mathcal{C} \rangle$ を指定するために必要なビット数を考える. グラフ (V, E) は隣接行列で表現すると $|V|^2$ ビットで表現できる. 各辺 $e \in E$ に付随する制約 $c'_e \subseteq \Sigma^2$ は, 各 $(a, b) \in \Sigma^2$ について $(a, b) \in c'_e$ かどうかを表すビットを (a, b) について並べれば良いため, 全部で $|E| \cdot |\Sigma|^2$ ビットで表現できる. よって, 制約グラフ G は $|V|^2 + |E| \cdot |\Sigma|^2 = O(\mathrm{size}(G)^2)$ ビットで表現できる. このことから, 制約グラフを入力として受け取るアルゴリズムが多項式時間かどうかを議論する際は, その時間計算量が $\mathrm{size}(G)$ に関して多項式かどうかを議論すれば良いことになる.

3.2 多重グラフの導入とエクスパンダーグラフ

以後, 制約グラフに関する様々な操作をしていく上で多重グラフのフォーマルな定義を与えておく.

定義 3.2.1 （多重グラフ）

有限集合 V と多重集合 E の組 (V, E) を**多重グラフ**という. ここで E は $V \cup \binom{V}{2}$ の元から構成される多重集合であり, $e \in E$ は $|e| = 1$ ならば**自己ループ**であるという.
二頂点 $u, v \in V$ の間の**重み**を

$$w(u, v) = |\{e \in E \colon e = \{u, v\}\}|$$

と定義し, 頂点 u の**次数**を

$$\deg(u) = \sum_{v \in V} w(u, v)$$

と定義する.[a] また, $W = (w(u, v))_{u,v \in V}$ を**重み行列**と呼び, $P(u, v) := \frac{w(u,v)}{\deg(u)}$ で定まる行列 $P \in [0, 1]^{V \times V}$ を**遷移確率行列**という.

[a] 自己ループの次数への寄与は 1 であることに留意されたい (文脈によってはこの寄与が 1 である場合もある).

グラフの次数を並べたベクトル $d = (\deg(v))_{v \in V} \in \mathbb{R}^V$ を**次数ベクトル**という. 次数ベクトルは

$$d = W\mathbf{1}$$

で与えられる. グラフ G が自己ループを含まない場合は握手補題が成り立つため $\sum_{u \in V} \deg(u) = 2|E|$ が成り立つが, そうでない場合は次数の総和は自己ループを一度ずつしかカウントしないため, 自己ループの個数を ℓ とすると

$$\sum_{u \in V} \deg(u) = 2|E| - \ell \leq 2|E| \tag{3.1}$$

が成り立つ.

以下に正則グラフとエクスパンダーグラフの定義を述べる.

定義 3.2.2 （正則性とエクスパンダー性）

n 頂点の多重グラフ $G = (V, E)$ の全ての頂点の次数が d に等しいとき, G は **d-正則**であるという.
また, 遷移確率行列 P の固有値 $1 = \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \geq -1$ が

$$\lambda(P) := \max\{|\lambda_2|, |\lambda_n|\} \leq \lambda$$

を満たすとき, G は **λ-エクスパンダー**であるという.

多重グラフ G が正則ならば P は対称行列となるため実固有値を持つことが直ちに従うが, 一般の場合でも実固有値を持つことが示せる. さらに, Gershgorin の定理からそれらの固有値の絶対値は 1 以下であることが従う. また, 全成分 1 のベクトル $\mathbf{1}$ は P の固有値 1 に対する固有ベクトルとなるため, $\lambda_1 = 1$ である.

演習問題 6

任意の多重グラフ G の遷移確率行列 P は実固有値を持つことを示せ.

第一固有値 1 の多重度はグラフの連結成分の個数に等しいことが知られている. ここではその特殊ケースである以下の事実を用いる.

命題 3.2.3

多重グラフ G が連結であるならば, 遷移確率行列 P の固有値 1 の多重度は 1, すなわち $\lambda_2 < 1$ である.

3.2.1 正則エクスパンダーの性質

この節では正則かつエクスパンダー性を持つ単純グラフの性質は同様に多重グラフに対しても成り立つことを確認する. 特に, 正則性より遷移確率行列 P は対称となることに留意されたい.

補題 3.2.4 （エクスパンダー混交補題）

連結な頂点数 n の多重グラフ G が d-正則かつ λ-エクスパンダーであるとする. 二つの頂点部分集合 $S, T \subseteq V$ に対して

$$W(S, T) = \sum_{u \in S, v \in T} w(u, v)$$

とすると, 任意の $S, T \subseteq V$ に対して

$$\left| W(S,T) - \frac{d}{n}|S||T| \right| \leq \frac{\lambda d}{n}\sqrt{|S||T||V \setminus S||V \setminus T|}$$

が成り立つ. 特に, $|S| \leq n/2$ を満たす任意の $S \subseteq V$ に対して

$$W(S, V \setminus S) \geq (1-\lambda)\frac{d|S|}{2}$$

が成り立つ.

注釈 3.2.5 （直感的な意味）

簡単のため自己ループを持たないグラフを考える. このグラフが n 頂点 d-正則ならば全部で $nd/2$ 本の辺を持つ. 全部で $\binom{n}{2} \approx n^2/2$ 個の頂点対があるため, 辺密度は d/n である. さて, グラフの辺が $\binom{V}{2}$ に「均一に」散らばっていると仮定すると, 任意に固定した頂点部分集合 $S, T \subseteq V$ に対してその間をまたがる辺の本数 $W(S,T)$ はおよそ $(d/n) \cdot |S||T|$ であることが期待される. グラフ G がエクスパンダー性を持つ場合, この期待値からのずれの上からの評価を与えるのがエクスパンダー混交補題である.

証明. 「特に, ...」の部分は前半の主張に $T = V \setminus S$ を代入して $|V \setminus S| \geq n/2$ を用いれば示せるので, 前半の主張を証明する.

全成分 1 の行列 $J \in \mathbb{R}^{V \times V}$ に対し, $M := P - \frac{1}{n}J$ とおく. また, 頂点部分集合 $S \subseteq V$ に対し, $\mathbf{1}_S \in \mathbb{R}^V$ を

$$\mathbf{1}_S(u) = \begin{cases} 1 & \text{if } u \in S \\ 0 & \text{otherwise} \end{cases}$$

で定める. 二つのベクトル $x, y \in \mathbb{R}^V$ を,

$$x = \mathbf{1}_S - \frac{|S|}{n}\mathbf{1},$$
$$y = \mathbf{1}_T - \frac{|T|}{n}\mathbf{1}$$

とする. ベクトル x, y は $\mathbf{1}$ に直交するので $\mathbf{1}_S = x + \frac{|S|}{n}\mathbf{1}$ は $\mathbf{1}_S$ の直交分解となっていること, 及び $W\mathbf{1} = d\mathbf{1}$ に着目すると,

$$\begin{aligned} W(S,T) &= \mathbf{1}_S^T W \mathbf{1}_T \\ &= x^T W y + \frac{|S||T|}{n^2}\mathbf{1}^T W \mathbf{1} \\ &= x^\top W y + \frac{d}{n}|S||T| \end{aligned}$$

が成り立つ.

従って

$$\begin{aligned}\left|W(S,T)-\frac{d}{n}|S||T|\right| &= \left|x^\top W y\right| \le \|x\|_2\|Wy\|_2 && \because \text{Cauchy-Schwarz の不等式}\\ &\le d\lambda\|x\|_2\|y\|_2 && \because \text{レイリー商と固有値の関係}\\ &= \frac{\lambda d}{n}\sqrt{|S||T||V\setminus S||V\setminus T|} && \because \|x\|_2^2=\frac{|S|(n-|S|)}{n}\end{aligned}$$

を得る. □

次にグラフの**べき乗**の操作を定義する.

定義 3.2.6 （グラフのべき乗）

重み行列 W を持つ多重グラフ $G=(V,E)$ および $k\ge 0$ に対し, 多重グラフ $G^k=(V,E^k)$ を

$$W'=W^k$$

を重み行列とする多重グラフと定める.

元のグラフ G の各二頂点 $u,v\in V$ に対し, uv 間の長さ k の路の個数だけ uv 間の辺を追加することで G^k を得られる.

補題 3.2.7 （正則エクスパンダー性のべき乗）

頂点数 n の d-正則かつ λ-エクスパンダーである多重グラフ G が与えられたとする. このとき, G^k は d^k-正則かつ λ^k-エクスパンダーである.

証明. べき乗で得られるグラフ G^k の重み行列は $W'=W^k$ であるため, その次数ベクトルは

$$W'\mathbf{1}=W^k\mathbf{1}=d^k\mathbf{1}$$

となるため, G^k は d^k-正則である.

さらに, G^k の遷移確率行列は P^k で与えられるため, $1=\lambda_1\ge\cdots\ge\lambda_n\ge -1$ に対して P^k の固有値は $1=\lambda_1^k\ge\cdots\ge\lambda_n^k\ge -1$ となる. 今, $\max\{|\lambda_2|,|\lambda_n|\}\le\lambda$ であるため, $\max\{|\lambda_2^k|,|\lambda_n^k|\}\le\lambda^k$ である. よって, G^k は λ^k-エクスパンダーである. □

3.2.2 エクスパンダーグラフの構成

エクスパンダーグラフは, その構造から様々な応用が知られている. 特に, 各 $n\ge 2$ に対して頂点数 n のエクスパンダーグラフを $n^{O(1)}$ 時間で構成できることが知られている.

定理 3.2.8 （エクスパンダーグラフの構成）

ある定数 $d_0 \geq 3$, $\lambda_0 < 1$ および以下を満たす多項式時間アルゴリズム A が存在する: アルゴリズム A は入力として $1^n = (\underbrace{1, \ldots, 1}_{n})$ を受け取り, 頂点数 n の d_0-正則かつ λ_0-エクスパンダーグラフの隣接行列 W を出力する.

定理 3.2.8 の証明はエクスパンダーグラフの構成に関するブレイクスルーの結果 [RVW02] の結果に軽微な修正を施すことによって得られる.

定理 3.2.9 （エクスパンダーグラフ族の構成）

ある定数 $d_0' \in \mathbb{N}$, $\lambda_0' < 1$ および以下を満たす多項式時間アルゴリズム A' が存在する: 各 $k \in \mathbb{N}$ に対して 1^{2^k} を入力として受け取り, 頂点数 2^k の d_0'-正則かつ λ_0'-エクスパンダーグラフの隣接行列 W' を出力する.

定理 3.2.9 の証明はジグザク積と呼ばれるグラフの積に関する手法を用いることで得られるが, 本講義のスコープからは逸脱するので割愛する. 定理 3.2.8 の証明, すなわち一般の頂点数のエクスパンダーグラフを得るには, まず $2^{k-1} < n \leq 2^k$ を満たす k に対して定理 3.2.9 を適用し, その後にそのグラフが n 頂点になるようにうまく二つの頂点を縮約し, 必要に応じて頂点に自己ループまたは多重辺を追加することによって正則にすることによって得られる.

なお, 補題 3.2.7 より, 次数を大きくすることによって固有値 λ_0 をいくらでも 0 に近づけることが可能である.

Chapter 4

ギャップ増幅補題

Dinur による PCP 定理の証明は, 与えられた制約グラフの不満足値を段階的に増幅していくアプローチに基づく. その中核を担うのがこのギャップ増幅補題であり, 制約グラフの不満足値を増幅できることを保証する補題である. 本チャプターはこの補題の証明を与える.

4.1 主張

ギャップ増幅補題は以下のように述べられる.

補題 4.1.1 （ギャップ増幅補題）

二つの定数 $c > 0, \alpha \in (0,1)$, アルファベット Σ, および以下の性質を満たす決定的多項式時間アルゴリズム A が存在する: アルゴリズム A は制約グラフ $G = \langle (V,E), \Sigma, \mathcal{C} \rangle$ を入力として受け取り, 次の性質を満たす別の制約グラフ $G' = \langle (V',E'), \Sigma, \mathcal{C}' \rangle$ を出力する:

- $\mathrm{size}(G') \leq c \cdot \mathrm{size}(G)$.
- $\mathrm{UNSAT}(G) = 0$ ならば $\mathrm{UNSAT}(G') = 0$.
- $\mathrm{UNSAT}(G) > 0$ ならば $\mathrm{UNSAT}(G') \geq \min\{\alpha, 2 \cdot \mathrm{UNSAT}(G)\}$.

PCP 定理 (定理 3.1.3) の証明は, この補題を繰り返し適用することによって得られる.

補題 *4.1.1* の下での定理 *3.1.3* の証明. 3 彩色問題のインスタンスを入力として受け取り, その制約グラフを G_0 とし, 頂点数を n とする. この制約グラフは単純グラフであるため, $\mathrm{UNSAT}(G_0) = 0$ もしくは $\mathrm{UNSAT}(G_0) \geq \frac{1}{n^2}$ である. 補題 4.1.1 のアルゴリズムを A とし, 各 $i = 1, \ldots, \lceil 2\log_2 n \rceil$ について, 制約グラフ G_i を $G_i = A(G_{i-1})$ として定義し, 最終的に得られる制約グラフを $G' = G_{\lceil 2\log_2 n \rceil}$ とする. 各 G_i のサイズは $\mathrm{size}(G_i) \leq c \cdot \mathrm{size}(G_{i-1})$ であるため, $\mathrm{size}(G') \leq c^{\lceil 2\log_2 n \rceil} \cdot \mathrm{size}(G_0) = \mathrm{size}(G_0)^{O(1)}$ である. 従って G' は多項式時間で構成できる.

また, 不満足値については, 以下のようになる:

- もしも G_0 が Yes インスタンスであるならば, 全ての i に対して G_i も Yes インスタンスであり, 特に $\mathrm{UNSAT}(G') = 0$ である.

- もしも G_0 が No インスタンスであるならば, $\mathrm{UNSAT}(G_0) \geq \frac{1}{n^2}$ かつ $\mathrm{UNSAT}(G_i) \geq \min\{\alpha, 2 \cdot \mathrm{UNSAT}(G_0)\}$ であるため, $\mathrm{UNSAT}(G') \geq \min\left\{\alpha, 2^{2\log_2 n} \cdot \frac{1}{n^2}\right\} = \alpha$ である.

□

ギャップ増幅補題では与えられた制約グラフを変換していく. 表記の簡略化のため, グラフに対する性質を表す用語を制約グラフにもそのまま適用することとする. 例えば (V, E) が連結であるときに $G = \langle (V, E), \Sigma, \mathcal{C} \rangle$ は連結であるという. また, (V, E) が正則であるときに $G = \langle (V, E), \Sigma, \mathcal{C} \rangle$ は正則であるという.

4.2 制約グラフの定数次数エクスパンダー化

まず, 与えられた制約グラフを, 不満足値をそれほど減らさずに定数次数の正則性かつエクスパンダー性を持つように変形する.

補題 4.2.1 （定数次数エクスパンダー化）

ある定数 $\lambda < 1$, $d \in \mathbb{N}$, $c > 0$, $\beta > 0$ が存在して, 任意の制約グラフ $G = \langle (V, E), \Sigma, \mathcal{C} \rangle$ を入力として受け取り, 以下の性質を満たす別の制約グラフ $G' = \langle (V', E'), \Sigma, \mathcal{C}' \rangle$ を出力する決定的多項式時間アルゴリズム A が存在する:

- G' は自己ループを持つ d-正則 λ-エクスパンダーである.
- $\mathrm{size}(G') \leq c \cdot \mathrm{size}(G)$.
- $\mathrm{UNSAT}(G) = 0$ ならば $\mathrm{UNSAT}(G') = 0$.
- $\mathrm{UNSAT}(G') \geq \beta \cdot \mathrm{UNSAT}(G)$.

この補題の証明は次数の削減とエクスパンダー化の二つのステップからなる.

4.2.1 次数の削減

まず, 与えられた制約グラフを定数次数の正則グラフに変換する補題を示す. この変換によって UNSAT の値は定数倍しか変化しない.

補題 4.2.2 （次数削減補題）

ある定数 $d \in \mathbb{N}$, $c' > 0$ が存在して, 任意の制約グラフ $G = \langle (V, E), \Sigma, \mathcal{C} \rangle$ を入力として受け取り, 以下の性質を満たす別の制約グラフ $G' = \langle (V', E'), \Sigma, \mathcal{C}' \rangle$ を出力する決定的多項式時間アルゴリズムが存在する:

- G' は d-正則である.
- $|V'| \leq 2|E|$.
- $\mathrm{UNSAT}(G) = 0$ ならば $\mathrm{UNSAT}(G') = 0$.
- $\mathrm{UNSAT}(G') \geq c' \cdot \mathrm{UNSAT}(G)$.

証明. 与えられた制約グラフを $G = \langle (V, E), \Sigma, \mathcal{C} \rangle$ とし, 変換によって得られる制約グラフを $G' = \langle (V', E'), \Sigma, \mathcal{C}' \rangle$ とする. フォーマルな構成を与える前に, まず図例を先に示す.

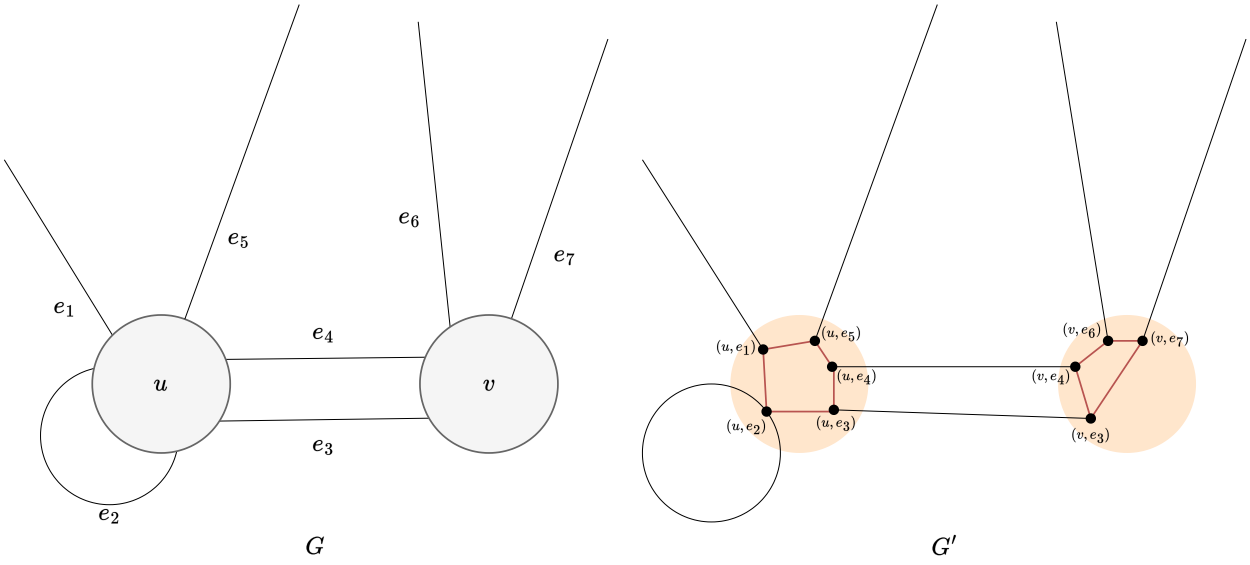

図 4.1: 次数削減変換の例. 橙色の内部の頂点集合がクラウドであり, クラウド内の辺は定理 3.2.8 によって構成されるが, ここでは図の簡単のため X_d を長さ d の閉路としている. G' における黒い辺の制約は対応する元のグラフの辺と同一の制約とし, クラウド内の赤辺の制約は等式制約とする.

元のグラフの各頂点 $u \in V$ に対し,

$$[u] = \{(u, e) \in \{u\} \times E \colon e = \{u, v\} \text{ for some } v \in V\}$$

を (この証明のローカルな用語として)u-**クラウド**と呼ぶことにする. 新しい頂点集合 V' は $V' = \bigcup_{u \in V} [u]$ である. すなわち, u をそれに接続する辺の本数 (自己ループは 1 個分としてカウント) だけコピーして得られる集合が $[u]$ である.

次に辺集合 E' を以下のように構成する. まず, $(X_n)_{n \in \mathbb{N}}$ を定理 3.2.8 によって構成される n 頂点 d_0-正則 λ-エクスパンダーの族とする. 元のグラフの各頂点 $u \in V$ に対し, $d = |[u]|$ としたとき, 頂点集合 $[u]$ 上で X_d と同型なグラフを構成し, それを $([u], E_u)$ とする. このとき, 各 u-クラウド内部の辺集合 E_{inner} は

$$E_{\text{inner}} = \bigcup_{u \in V} E_u$$

とする. 次に異なるクラウド間を繋ぐ辺集合 E_{outer} を

$$E_{\text{outer}} = \{\{(u, e), (v, e)\} \colon e \notin [u] \cap [v]\}$$

とする. このとき, グラフ G' の辺集合 E' は

$$E' = E_{\text{inner}} \cup E_{\text{outer}}$$

となる.

次に各辺 $e' \in E'$ の制約 $c'_{e'}$ を以下のように定める:

- 辺 $e' \in E_{\text{outer}}$ がクラウド間をつなぐ辺ならば, その制約は対応する元の辺 e の制約と同じとする. すなわち, $e' = \{(u,e),(v,e)\}$ ならば, $c'_{e'} = c_e$ である.
- 辺 $e' \in E_{\text{inner}}$ がクラウド内部の辺ならば, その制約を $c'_{e'} = \{(\sigma,\sigma)\colon \sigma \in \Sigma\}$, すなわち等式制約とする.

このようにして得られる制約グラフ $G' = \langle (V',E'),\Sigma,\mathcal{C}'\rangle$ が補題の主張を全て満たすことを確認する. まず, G' は (d_0+1)-正則である. 実際, G' の各頂点 $(u,e) \in V'$ に接続する辺は, クラウド内の辺が d_0 本, クラウド間の辺が 1 本である. また, G から G' を構成する際に新たに追加する辺は E_{inner} のみであり, これは高々 $\sum_{u\in V} \frac{d_0 \deg_G(u)}{2} \le d_0|E|$ 本である (ここで式 (3.1) を用いた). 従って

$$|E'| = |E_{\text{inner}}| + |E_{\text{outer}}| \le (d_0+1)|E| \tag{4.1}$$

を満たす. また, V' の要素数は次数の総和に等しいため, $|V'| \le 2|E|$ を満たす. 次に, $\mathrm{UNSAT}(G) = 0$ ならば $\mathrm{UNSAT}(G') = 0$ である. 実際, 元の制約グラフの全ての制約を満たす割り当て $a\colon V \to \Sigma$ に対し, G' の割り当て $a'\colon V' \to \Sigma$ を

$$a'(u,e) = a(u)$$

と定めると, a' は G' の制約を満たす (同一クラウド内の頂点には全て同じ値が割り当てられるため, クラウド内の辺の制約は全て満たされ, クラウド間の辺に対応する制約は a の取り方により全て満たされることがわかる).

最後の主張, すなわち $\mathrm{UNSAT}(G') \ge c' \cdot \mathrm{UNSAT}(G)$ となる定数 $c' > 0$ が存在すること示す. 定数 $c' > 0$ を

$$c' = \min\left\{\frac{(1-\lambda_0)d_0}{8(d_0+1)}, \frac{1}{2(d_0+1)}\right\} \tag{4.2}$$

と定める. $\mathrm{UNSAT}(a';G') = \mathrm{UNSAT}(G')$ を満たす割り当て $a'\colon V' \to \Sigma$ を任意に取る. この割り当て a' に対し, 元のグラフ G の割り当て $a\colon V \to \Sigma$ を

$$a(u) = \mathrm{Maj}((a'(u,e))_{(u,e)\in[u]})$$

と定める. ここで $\mathrm{Maj}(\cdot)$ は多数決関数であり, タイは任意に選ぶとする. 例えば $\mathrm{Maj}(1,2,2) = 2$, $\mathrm{Maj}(1,2,3,3) = 3$, $\mathrm{Maj}(1,2,2,3,3) = 2$ である ($\mathrm{Maj}(1,2,2,3,3) = 3$ としても良いが, ここでは便宜上小さい方の数字を採用している).

以降, 割り当て $a'\colon V' \to \Sigma$ に対し, 頂点 $(u,e) \in V'$ への割り当て $a'(u,e)$ を頂点 (u,e) の**意見**と呼ぶこととする. 同様に, 割り当て $a\colon V \to \Sigma$ に対し, 頂点 $u \in V$ への割り当て $a(u)$ を頂点 u の意見と呼ぶこととする. 直感的には頂点 u の意見は対応する u-クラウド内の頂点の意見の多数決である.

辺集合 $F \subseteq E$ を割り当て a によって充足されない辺の集合, すなわち

$$F = \{e = \{u,v\} \in E\colon (a(u),a(v)) \notin c_e\}$$

と定義する. ここで $\mathcal{C} = (c_e)_{e\in E}$ は元の制約グラフ G の制約である. 同様に, $F' \subseteq E'$ を割り当て a' によって充足されない辺の集合とする. このとき $\mathrm{UNSAT}(a;G) = \frac{|F|}{|E|}$ および

$\mathrm{UNSAT}((a';G') = \frac{|F'|}{|E'|}$ である. 頂点部分集合 $S \subseteq V'$ を a の多数決で選ばれなかった意見を持つ頂点の集合, すなわち

$$S = \{(u,e) \in V' \colon a(u) \neq a'(u,e)\}$$

と定義し, 各 $v \in V$ に対し $S^v = S \cap [v]$ とし, 各 $\sigma \in \Sigma$ に対し $S^v_\sigma = \{(u,e) \in S^v \colon a'(u,e) = \sigma\}$ とする (図 4.2). なお, $\sigma \in \Sigma$ が多数決の意見 (すなわち $\sigma = a(v)$) のとき, $S^v_\sigma = \emptyset$ とする.

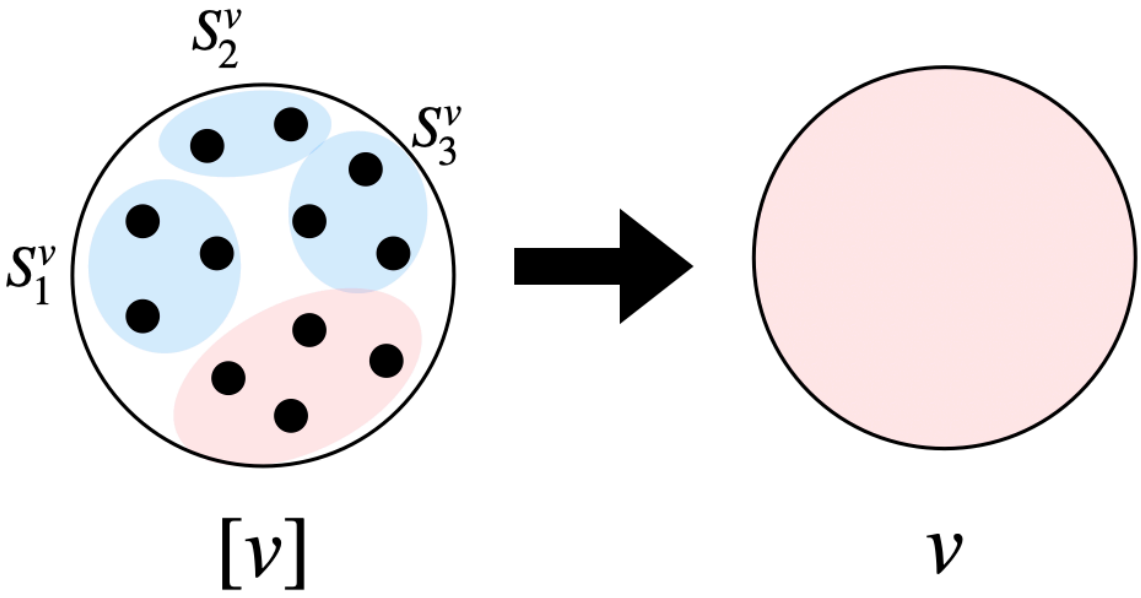

図 4.2: 多数決によって選ばれなかった v-クラウド内の頂点の集合を $S_v \subseteq [v]$ とする.

以下の二つのケースを考える:

ケース 1. $|S| \geq \frac{\mathrm{UNSAT}(a;G)}{2}|E|$ **の場合.** 各頂点 $v \in V$ と多数決によって選ばれなかった意見 $\sigma \in \Sigma \setminus \{a(v)\}$ に対し $|S^v_\sigma| \leq \frac{|[v]|}{2}$ である (そうでなければ意見 σ が多数決によって選ばれなかったことに矛盾する). ここで v-クラウド内部のグラフは d_0-正則 λ_0-エクスパンダーであるため, その頂点部分集合 $S^v_\sigma \subseteq [v]$ に対するエクスパンダー混交補題 (補題 3.2.4) より,

$$W(S^v_\sigma, [v] \setminus S^v_\sigma) \geq (1-\lambda_0) \cdot \frac{d_0|S^v_\sigma|}{2}$$

となる. ここで $W(S,T)$ は v-クラウド内部のグラフ $([v], E_v)$ の辺であって S と T の間をまたがる辺の本数である. ここで S^v_σ と $[v] \setminus S^v_\sigma$ をまたがる全ての辺の両端点の意見は異なるた

め, a' はこれらの辺を充足しない. よってこれらの辺は全て F' に含まれる. 従って

$$\begin{aligned}
|F'| &\geq \frac{1}{2}\sum_{v,\sigma} W(S^v_\sigma, [v] \setminus S^v_\sigma) \\
&\geq \frac{(1-\lambda_0)d_0}{4}\sum_{v,\sigma}|S^v_\sigma| \\
&= \frac{(1-\lambda_0)d_0}{4}|S| \\
&\geq \frac{(1-\lambda_0)d_0}{8}\mathrm{UNSAT}(a;G)\cdot|E| && \because \text{ケース 1 の仮定} \\
&\geq \frac{(1-\lambda_0)d_0}{8(d_0+1)}\mathrm{UNSAT}(a;G)\cdot|E'|. && \because \text{式 (4.1)}
\end{aligned}$$

特にこれから $\mathrm{UNSAT}(a';G') = \frac{|F'|}{|E'|} \geq \frac{(1-\lambda_0)d_0}{8(d_0+1)} \cdot \mathrm{UNSAT}(a;G)$ が成り立つ.

ケース 2. $|S| < \frac{\mathrm{UNSAT}(a;G)}{2}|E|$ の場合. 任意の辺 $e = \{u, v\} \in F$ に対して, それに対応する G' の辺 $e' = \{(u, e), (v, e)\}$ を考える.

- もしも $a(u) = a'(u, e)$ かつ $a(v) = a'(v, e)$ が成り立つならば, e' と e は同じ制約を持ち, $e \in F$ であることから $e' \in F'$ である.
- そうでない, すなわち $a(u) \neq a'(u, e)$ または $a(v) \neq a'(v, e)$ が成り立つならば, (u, e) と (v, e) のうち少なくとも一方はその意見が多数決として選ばれない, すなわち $(u, e) \in S$ または $(v, e) \in S$ が成り立つ.

以上より $|F| \leq |F'| + |S|$ が成り立つので

$$\begin{aligned}
|F'| &\geq |F| - |S| \\
&\geq |E|\mathrm{UNSAT}(a;G) - \frac{\mathrm{UNSAT}(a;G)}{2}|E| && \because \text{ケース 2 の仮定} \\
&= \frac{\mathrm{UNSAT}(a;G)}{2}|E| \\
&\geq \frac{\mathrm{UNSAT}(a;G)}{2(d_0+1)}|E'| && \because \text{式 (4.1)}
\end{aligned}$$

となり, これから

$$\mathrm{UNSAT}(a';G') = \frac{|F'|}{|E'|} \geq \frac{1}{2(d_0+1)} \cdot \mathrm{UNSAT}(a;G)$$

が成り立つ.

ケース 1,2 より, 式 (4.2) で定まる定数 $c' > 0$ に対して $\mathrm{UNSAT}(a';G') \geq c' \cdot \mathrm{UNSAT}(a;G) \geq c' \cdot \mathrm{UNSAT}(G)$ が成り立つ. □

4.2.2 エクスパンダー化

次に, 与えられた正則な制約グラフをエクスパンダーグラフに変換する補題を示す.

補題 4.2.3 （エクスパンダー化補題）

ある定数 $\lambda < 1, d, d_0 \in \mathbb{N}$ および次を満たす多項式時間アルゴリズム A が存在する: アルゴリズム A は入力として d-正則な制約グラフ $G = \langle (V, E), \Sigma, \mathcal{C} \rangle$ を受け取り, $(d + d_0 + 1)$-正則で全頂点が自己ループを持ち, さらに λ-エクスパンダーである制約グラフ $G' = \langle (V, E'), \Sigma, \mathcal{C}' \rangle$ を出力する. さらに, この制約グラフ G' は $\mathrm{UNSAT}(G') = \frac{d}{d+d_0+1}\mathrm{UNSAT}(G)$ を満たす.

証明. 入力として与えられた制約グラフを $G = \langle (V, E), \Sigma, \mathcal{C} \rangle$ とする. 変換は以下のようにして行われる:

1. まず, 定理 3.2.8 を用いて頂点集合 V 上 d_0-正則 λ_0-エクスパンダーグラフを構成し, その辺集合を E に追加する (この際多重辺も許す).
2. 次に, 各頂点に自己ループを付与する.
3. ステップ 1,2 で追加した辺 e に対応する制約 c'_e を自明な制約 $c'_e = \Sigma^2$ とする.

このようにして得られる制約グラフを $G' = \langle (V', E'), \Sigma, \mathcal{C}' \rangle$ とする. 元の制約グラフ G が d-正則であることから, G' は $(d + d_0 + 1)$-正則である (自己ループの次数への寄与は 1 であることに注意). また, ステップ 2 より G' は全頂点が自己ループを持つ.

次に $\mathrm{UNSAT}(G') = \frac{d}{d+d_0+1}\mathrm{UNSAT}(G)$ を示す. 任意の割り当て $a\colon V \to \Sigma$ に対し, ステップ 1,2 で追加した辺に対応する制約は常に充足されるため, 両者が充足**しない**制約の個数は一致する. すなわち $|E|\mathrm{UNSAT}(a; G) = |E'|\mathrm{UNSAT}(a; G')$ が成り立つので

$$\mathrm{UNSAT}(G') = \min_a \mathrm{UNSAT}(a; G') = \frac{|E|}{|E'|} \min_a \mathrm{UNSAT}(a; G) = \frac{d}{d + d_0 + 1}\mathrm{UNSAT}(G)$$

が成り立つ.

最後に G' が $\lambda := \frac{d}{d+d_0+1} + \frac{\lambda_0(d_0+1)}{d+d_0+1}$ に対し λ-エクスパンダーであることを示す. 元のグラフ G の遷移確率行列を $P \in [0,1]^{V \times V}$ とし, ステップ 1,2 で追加した辺からなるグラフを G_0 とし, その遷移確率行列を $P_0 \in [0,1]^{V \times V}$ とする. また, 最終的に得られるグラフ G' の遷移確率行列を $P' \in [0,1]^{V \times V}$ とする. 元のグラフ G は d-正則, 追加したグラフ G_0 は $(d_0 + 1)$-正則であるため

$$P' = \frac{d}{d + d_0 + 1} P + \frac{d_0 + 1}{d + d_0 + 1} P_0$$

となる. さらに P_0 は λ_0-エクスパンダーであるため, 全成分 1 のベクトル $\mathbf{1}$ と直交する任意のベクトル $x \in \mathbb{R}^V$ に対し

$$\begin{aligned} x^\top P' x &= \frac{d}{d + d_0 + 1} x^\top P x + \frac{d_0 + 1}{d + d_0 + 1} x^\top P_0 x \\ &\geq \frac{d}{d + d_0 + 1} \|x\|^2 + \frac{d_0 + 1}{d + d_0 + 1} \lambda_0 \|x\|^2 \qquad \because P_0 \text{ のエクスパンダー性} \\ &\leq \left(\frac{d}{d + d_0 + 1} + \frac{\lambda_0(d_0 + 1)}{d + d_0 + 1} \right) \|x\|^2 \end{aligned}$$

より, $\lambda(P') \leq \frac{d}{d+d_0+1} + \frac{\lambda_0(d_0+1)}{d+d_0+1}$ が成り立つ. □

これにより補題 4.2.1 を示す準備が整った.

補題 *4.2.1* の証明. 与えられた制約グラフ G に対し, まず G を入力として補題 4.2.2 のアルゴリズムを実行し, その出力を G_1 とする. この制約グラフ G_1 は補題 4.2.2 の主張より, d-正則である. 次に, G_1 を入力として補題 4.2.3 のアルゴリズムを実行し, その出力を最終的な出力 G とする. この制約グラフ G' は補題 4.2.3 の主張より, 全頂点が自己ループを持ち, $(d+d_0+1)$-正則かつ λ-エクスパンダーである. また, この一連の変換によって $\mathrm{size}(G')$ は $\mathrm{size}(G)$ の定数倍にしかならない.

最後に UNSAT の値を考える. 元の制約グラフ G が $\mathrm{UNSAT}(G)=0$ ならば, 補題 4.2.2 と 4.2.3 より $\mathrm{UNSAT}(G')=\frac{d}{d+d_0+1}\mathrm{UNSAT}(G_1)=0$ を得る. 一方で

$$\mathrm{UNSAT}(G')=\frac{d}{d+d_0+1}\mathrm{UNSAT}(G_1)\geq\frac{d}{d+d_0+1}\cdot c'\cdot\mathrm{UNSAT}(G).$$

すなわち $\beta=\frac{d}{d+d_0+1}\cdot c'$ に対して主張が成り立つことが示された. □

4.3 制約グラフのべき乗

補題 4.2.1 により, 任意の制約グラフを定数次数エクスパンダーに変換することができる. この節では, そのような制約グラフに対し, 以下で定義する**べき乗**という操作を考えることで, その UNSAT の値を増幅させることができることを示す.

定義 4.3.1 (制約グラフの冪乗)

パラメータ $t\in\mathbb{N}$ および全頂点が自己ループを持つ d-正則な制約グラフ $G=\langle(V,E),\Sigma,\mathcal{C}\rangle$ に対し, べき乗 $G^t=\langle(V,\boldsymbol{E}),\Sigma^{\lceil t/2\rceil},\mathcal{C}^t\rangle$ を, 以下のようにして定義する:

- 頂点集合は同じ V とする.
- 二頂点 $u,v\in V$ に対し, 長さ t の uv-路の個数と同じだけ uv 間に多重辺を用意する. このようにして得られる辺の多重集合を $\boldsymbol{E}$ とし, その元を $\boldsymbol{e}=\{u,v\}\in\boldsymbol{E}$ とする. すなわち, $(V,\boldsymbol{E})$ は (V,E) のべき乗 (定義 3.2.6) に一致する.
- アルファベット集合は $\Sigma^{d^{\lceil t/2\rceil}}$ とする. ここで, 頂点 u に $\boldsymbol{\sigma}=(\sigma_1,\dots,\sigma_{d^{\lceil t/2\rceil}})\in\Sigma^{d^{\lceil t/2\rceil}}$ が割り当てられたとき, 次の意味を持つ: 元の d-正則グラフ G 上で頂点 u から距離 $\lceil t/2\rceil$ 以内の頂点の集合を $\Gamma(u)$ とし, その元を頂点番号の大小順で並べて $\Gamma(u)=\{v_1,\dots,v_\ell\}$ とする (ここで G の d-正則性より $\ell\leq d^{\lceil t/2\rceil}$ である). このとき, $\boldsymbol{\sigma}$ は $v_1,\dots,v_\ell$ の各頂点に Σ の元を割り当てる関数 $\boldsymbol{\sigma}\colon\Gamma(u)\to\Sigma$ とみなし, $w\in\Gamma(u)$ に対し $\sigma=\boldsymbol{\sigma}(w)$ を, u **の** w **に対する意見**と呼ぶ. なお, $\ell<d^{\lceil t/2\rceil}$ の場合は相異なる二つの $\boldsymbol{\sigma},\boldsymbol{\sigma}'\in\Sigma^{d^{\lceil t/2\rceil}}$ が同一の関数としてみなされることもある.
- 辺 $\boldsymbol{e}=\{u,v\}\in\boldsymbol{E}$ に対応する制約 $\boldsymbol{c_e}\in\mathcal{C}^t$ は次の二つを満たす割り当て $(\boldsymbol{\sigma},\boldsymbol{\sigma}')\in\Sigma^{d^{\lceil t/2\rceil}}\times\Sigma^{d^{\lceil t/2\rceil}}$ によって満たされる:
  - 全ての頂点 $a\in\Gamma(u)\cap\Gamma(v)$ に対し, $\boldsymbol{\sigma}(a)=\boldsymbol{\sigma}'(a)$ が成り立つ.
  - 一方の端点が $\Gamma(u)$, もう一方の端点が $\Gamma(v)$ に属する元のグラフ G の全ての辺 $e=\{s,t\}\in E$ に対し, $(\boldsymbol{\sigma}(s),\boldsymbol{\sigma}'(t))\in c_e$ が成り立つ.

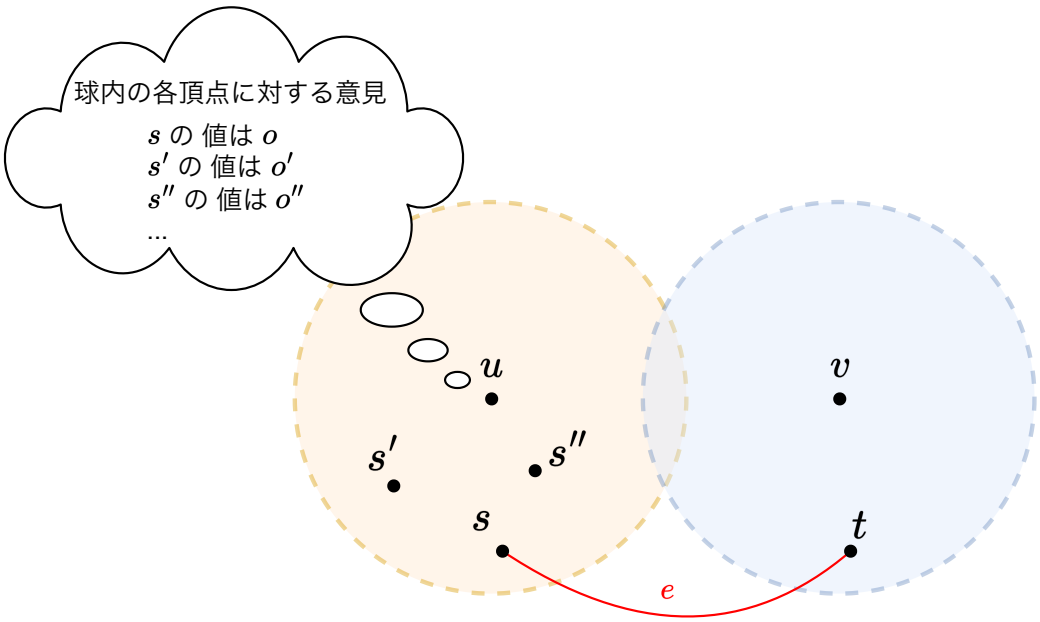

図 4.3: 制約グラフ G のべき乗 G^t の図例. 頂点 u, v にそれぞれ $\boldsymbol{\sigma}, \boldsymbol{\sigma}' \in \Sigma^{\lceil t/2 \rceil}$ が割り当てられているとき, それぞれ関数 $\boldsymbol{\sigma}\colon \Gamma(u) \to \Sigma$ と $\boldsymbol{\sigma}'\colon \Gamma(v) \to \Sigma$ とみなす. なお, 元のグラフ G が自己ループを持つことから, 距離 $\lceil t/2 \rceil$ 以内の全ての頂点に対して意見が定義される.

パラメータ t は制約グラフのサイズ $\mathrm{size}(G)$ には依存しない定数であることを常に想定する. べき乗をで得られた制約グラフは元の制約グラフに対して UNSAT の値が増幅している.

補題 4.3.2 （べき乗の性質）

ある定数 $\beta = \beta(\lambda, d, |\Sigma|)$ が存在して以下が成り立つ: パラメータ $t \in \mathbb{N}$ および全頂点が自己ループを持つ d-正則な制約グラフ $G = \langle (V, E), \Sigma, \mathcal{C} \rangle$ に対し, べき乗 $G^t = \langle (V, \boldsymbol{E}), \Sigma^{\lceil t/2 \rceil}, \mathcal{C}^t \rangle$ を, 定義 4.3.1 で定義する. このとき

- 元の制約グラフ G が λ-エクスパンダーであるとき, G^t は d^t-正則 λ^t-エクスパンダーである.
- $\mathrm{size}(G^t) \leq \mathrm{size}(G) \cdot d^t$ である.
- $\mathrm{UNSAT}(G) = 0$ ならば $\mathrm{UNSAT}(G^t) = 0$ である.
- $\mathrm{UNSAT}(G^t) \geq \beta\sqrt{t} \cdot \min\left\{\mathrm{UNSAT}(G), \frac{1}{t}\right\}$ である.

証明. 補題 3.2.7 により, G が d-正則かつ λ-エクスパンダーならば G^t は d^t-正則かつ λ^t-エクスパンダーとなる. 特に, $\mathrm{size}(G^t) \leq \mathrm{size}(G) \cdot d^t$ である.

また, $\mathrm{UNSAT}(G) = 0$ ならば, 全制約を満たす割り当て $a\colon V \to \Sigma$ を用いて G^t の全ての

制約を満たす割り当て $a' \colon V \to \Sigma^{\lceil t/2 \rceil}$ を, 各頂点 $u \in V$ に対し

$$\boldsymbol{\sigma}(u) \colon s \mapsto a(s)$$

となるような $\boldsymbol{\sigma} \in \Sigma^{\lceil t/2 \rceil}$ を $a'(u)$ と定めることによって得られる. すなわち $\mathrm{UNSAT}(G^t) = 0$ である.

最後に $\mathrm{UNSAT}(G^t)$ の下界を証明する. べき乗の制約グラフ G^t における割り当て $a' \colon V \to \Sigma^{\lceil t/2 \rceil}$ に対し, 元の制約グラフに対する割り当て $a \colon V \to \Sigma$ を, 「重み付き多数決」によって定める. 頂点 $u \in V$ に対し,

□

4.4 アルファベット削減

Bibliography

[AKS04] M. Agrawal, N. Kayal, and N. Saxena. "PRIMES is in P". en. In: **Annals of mathematics** 160 (2 Sept. 1, 2004), pp. 781–793. DOI: 10.4007/annals.2004.160.781 (cit. on p. 11).

[ALMSS98] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. "Proof verification and the hardness of approximation problems". In: **Journal of the ACM** 45 (3 May 1, 1998), pp. 501–555. DOI: 10.1145/278298.278306 (cit. on p. 7).

[AS98] S. Arora and S. Safra. "Probabilistic checking of proofs: a new characterization of NP". In: **Journal of the ACM** 45 (1 Jan. 1, 1998), pp. 70–122. DOI: 10.1145/273865.273901 (cit. on p. 7).

[BLR93] M. Blum, M. Luby, and R. Rubinfeld. "Self-testing/correcting with applications to numerical problems". In: **Journal of Computer and System Sciences** 47 (3 1993), pp. 549–595. DOI: 10.1016/0022-0000(93)90044-W (cit. on p. 20).

[Coo71] S. A. Cook. "The complexity of theorem-proving procedures". en. In: **Proceedings of the third annual ACM symposium on Theory of computing - STOC '71**. the third annual ACM symposium (Shaker Heights, Ohio, United States). New York, New York, USA: ACM Press, 1971. DOI: 10.1145/800157.805047 (cit. on p. 14).

[Din07] I. Dinur. "The PCP theorem by gap amplification". In: **Journal of the ACM** 54 (3 June 1, 2007), 12–es. DOI: 10.1145/1236457.1236459 (cit. on p. 7).

[Kar72] R. M. Karp. "Reducibility among Combinatorial Problems". en. In: **Complexity of Computer Computations**. Boston, MA: Springer US, 1972, pp. 85–103. DOI: 10.1007/978-1-4684-2001-2_9 (cit. on p. 14).

[Lev73] L. A. Levin. "Universal Sequential Search Problems". In: **Problems of Information Transmission (translated from Problemy Peredachi Informatsii (Russian))** 9 (3 1973), pp. 115–116 (cit. on p. 14).

[RVW02] O. Reingold, S. Vadhan, and A. Wigderson. "Entropy Waves, the Zig-Zag Graph Product, and New Constant-Degree Expanders". In: **Annals of mathematics** 155 (1 Jan. 2002), p. 157. DOI: 10.2307/3062153 (cit. on p. 39).