PCP 定理の証明

清水 伸高 (東京科学大学)

目次

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で開催される集中講義「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.

第 1 章

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]が, 主張 1.2.12 において簡単に触れる.

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.

[*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.

- 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.
- 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.
- 集合 S 上のベクトル $x \in S^n$ および $I = \{i_1, \ldots, i_\ell\} \subseteq [n]$ に対し, $x_I = (x_{i_1}, \ldots, x_{i_\ell}) \in S^\ell$ を, x の I への制限とする.
- $\{0,1\}^* = \bigcup_{n \in \mathbb{N}} \{0,1\}^n$ 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.

[*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.

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入力を問題 L の**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, $i = \lceil k/|V| \rceil + 1, j = k \mod |V| + 1$ 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. このときのイン

デックス i を**クエリ**と呼ぶ[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.
オラクルアルゴリズム $A^{\pi}(x)$ の各オラクルアクセスのクエリ i が, それ以前のオラクルアクセスの内容に依存するとき, そのオラクルアルゴリズムを**適応的**といい, そうでない場合は**非適応的**という. 本講義では特に常に非適応的なオラクルアルゴリズムのみを扱う.

[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, 一回の質問を行うたびに最悪の場合 $O(\log |\pi|)$ 時間かかる.

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. より詳細な背景はセクション 1.3 を参照されたい.

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. このとき V は入力 (x,π) を**受理する**という.
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. このとき V は入力 (x,π) を**拒否する**という.

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.

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.

例 1.2.3 では, 与えられた数字が合成数であることの証拠として非自明な約数を与えることで, 検証者が多項式時間で検証できることを確認した. では, 与えられた数が素数であることの証拠は提示できるだろうか? 興味深いことに, 実はこれは可能である.

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 $\mathrm{Primes} = \{(a)_2 \in \mathbb{N} : a \text{ は素数 }\}$ 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 $\mathrm{Primes} \in \mathsf{NP}$ 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 $\gamma \in \{1, \ldots, a-1\}$ に対し，$\gamma^0, \gamma^1, \ldots, \gamma^{a-2} \pmod a$ 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 $\gamma \in \{1, \ldots, a-1\}$ 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．

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.

学術雑誌の査読を例に考えてみよう. 入力 x は査読している論文の主張であり, その証明を表す文字列が π で与えられている. PCP 検証者とは, 主張が正しいかどうかを証明の中の q 個の文字を読むだけで確率的に判定する. 一般的な感覚で考えると $q \ll |\pi|$ ならば, そのようなことは不可能であるように思える. ところが, 例えば手料理における味見では, 調理中の料理の味を, そのランダムな部分を少し食べることで判定するという行為であり, これはある意味で PCP 検証者のようなものである.

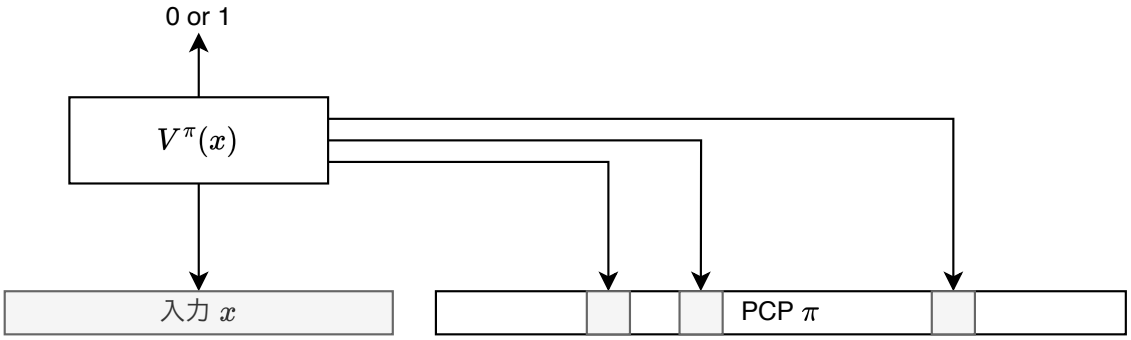


図 1.1 PCP 検証者のイメージ図

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．

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**回路充足可能性判定問題**と呼ばれる問題が NP 完全であることを示していた．回路充足可能性判定問題は論理回路に関する問題であり一見するとグラフなどとは関係なさそうに見えるが, Karp [Kar72] は, 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

[*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．

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回路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**回路充足可能性判定問題** (Circuit SAT) という. 回路充足可能性判定問題は NP 完全である.

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. すなわち, 回路による計算はチューリング機械を模倣できるという性質が完全性において肝要な性質であり, 以下のように述べられる:

主張 1.2.12 （チューリング機械を模倣する回路）

任意の多項式時間チューリング機械 M に対して, ある回路族 $(C_n)_{n\in\mathbb{N}}$ が存在して以下が成り立つ:

- 全ての $n \in \mathbb{N}$ に対して, $C_n\colon \{0,1\}^n \to \{0,1\}$ は $n^{O(1)}$ 個のゲート (AND,OR,NOT) を含む. なお, ここでは全ての素子は二つの入力を受け取り一つの出力を持つ.
- 任意の $n \in \mathbb{N}$ と全ての $x \in \{0,1\}^n$ に対し, M が x を受理することと $C_n(x) = 1$ は同値.
- 各 C_n は $n^{O(1)}$ 時間で構成できる.

主張 1.2.12 を証明するにはチューリング機械の厳密な定義なども必要であり講義では詳しくは扱わないが, その概要を以下に説明する.

証明の概要. チューリング機械とは, 無限の長さの**記憶テープ**を備えた有限状態機械である. **記憶テープ**とは無限の長さの記号列であり, 入力 $x \in \{0,1\}^*$ に対してまずは

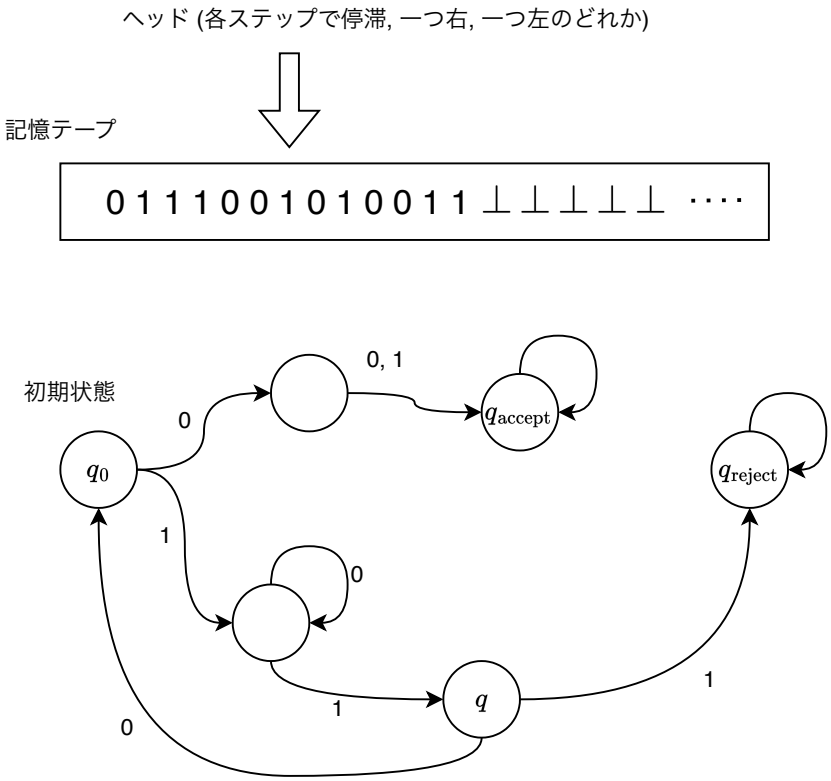


図 1.2 チューリング機械の一例. 各状態において, ヘッドの指し示す記憶テープの値のラベルがついた有効辺に沿って遷移し, そのヘッドの位置にある記憶テープの値を必要があれば書き換え, 最後にヘッドの位置を一つ右, 一つ左のいずれかに移動するかそのままにする.

最初の $|x|$ 文字は x が記載されており, それ以外の文字は全て $\perp$(空白であることを表す記号) である. また, **ヘッド**と呼ばれるポインタがあり, これは現在は記憶テープのどの位置の文字を読み書きするかを表す. 各ステップにおいて, 現在の状態とヘッドが指し示す文字を参照し, 次の状態, 現在のヘッドに書き込む内容, ヘッドの移動方向 (左右もしくはそのまま) を決める. さらに機械には二つの相異なる二つの特別な状態 q_{accept} と q_{reject} があり, これらに遷移すると計算が終了する. 終了時の状態に応じてチューリング機械の出力 (accept または reject) が決まる (図 1.2 を参照).

計算が終了するまでに要したステップ数をこのチューリング機械の **(時間) 計算量**という. チューリング機械 M が多項式時間であったとする. このとき, 任意の十分大きな $n \in \mathbb{N}$ と任意の $x \in \{0,1\}^n$ に対して, M の計算の任意の時点での記憶テープの内容 ($\perp$ 以外の文字の個数) は高々 $n^{O(1)}$ 文字で表現できる. また, ヘッドの指し示す位置は $O(\log n)$ ビットで表現でき, 状態の個数は高々 $O(1)$ である. 従って, 任意の時点でのチューリング機械の, 記憶テープの内容も含めた状態 (便宜上これを計算状態と呼ぶことにする) は高々 $n^{O(1)}$ ビットで表現できる. また, 記憶テープの書き換え

は高々 1 ビットの書き換えであり, ヘッド位置の変更も加算と減算で計算できるため, チューリング機械の計算の各ステップもまた小さいサイズの回路によって模倣できる. すなわち, 現在の計算状態を入力として受け取り, 計算状態の第 i 番目のビットを出力する回路 C_i を用意してこれを全ての $i \in [n^{O(1)}]$ に対して並べることによって, 回路として表現できる. これを全てのステップについて並べれば, チューリング機械 M を模倣する多項式サイズの回路 C (すなわち, $M(x)$ の最終状態が q_{accept} または q_{reject} であるかどうかを判定する回路) が得られる. □

これを用いて Cook-Levin の定理の証明を与える.

定理 *1.2.11* の証明. NP の定義より, 任意の判定問題 $L \in \mathsf{NP}$ に対して, 多項式時間の NP 検証者 $V(x;\pi)$ が存在する. 主張 1.2.12 より, この NP 検証者を模倣する回路族 $(C_n)_{n\in\mathbb{N}}$ が存在する. 特に, 入力 $x \in \{0,1\}^*$ が与えられたとき, $V(x;\cdot)$ というチューリング機械を模倣する回路 $C_x\colon \{0,1\}^{|\pi|} \to \{0,1\}$ が多項式時間で構成できる. さらにこれが L から回路充足可能性判定問題へのカープ帰着になっていることを確認すれば証明は完了する. 実際, x が Yes インスタンスであるならば $V(x;\pi) = 1$ を満たす π が存在することから, C_x は充足可能である. 一方で x が No インスタンスであるならば C_x は充足可能でない. 従って $x \mapsto C_x$ は確かに L から回路充足可能性判定問題へのカープ帰着になっている. □

Cook-Levin の定理を用いると, 以下に示す二次方程式系の充足可能性判定問題が NP 完全であることが示せる:

定理 1.2.13 (二次方程式系の充足可能性判定問題)

判定問題 QuadEq を以下のように定義する: m 個の n 変数二次関数 $f_1,\dots,f_m\colon \mathbb{F}^n \to \mathbb{F}_2$ が与えられる. このとき, ある $x \in \mathbb{F}^n$ が存在して $f_1(x) = \cdots = f_m(x) = 0$ を満たすかどうかを判定せよ. 特に, 全ての $i \in [m]$ に対し, $f_i(x_1,\dots,x_n)$ の値が高々 k 個の変数 $x_{i_1},\dots,x_{i_k}$ の値のみに依存するようなとき, k-QuadEq と呼ぶ.
3-QuadEq は NP 完全である.

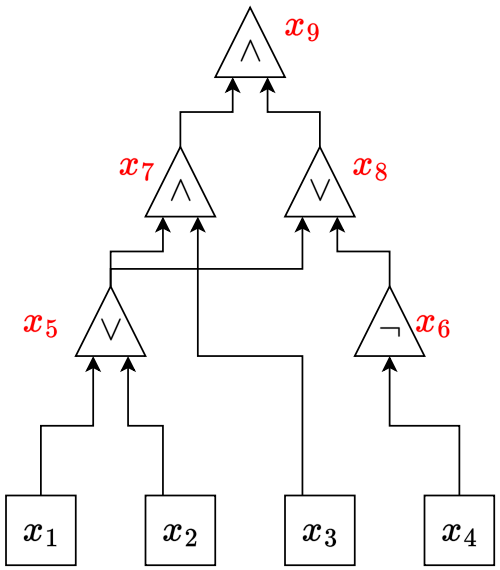
証明. 3-QuadEq は NP に属する. 実際, NP 証拠として連立方程式の解 $x \in \mathbb{F}^n$ を与え, それが実際に $f_1(x) = \cdots = f_m(x) = 0$ を満たすかどうかは $n^{O(1)}$ 時間で判定できる (入力として $f_1,\dots,f_m$ が与えているため, これを二進文字列として表現したときの長さは n 以上である).

次に, 回路充足可能性判定問題から 3-QuadEq へのカープ帰着を構成する. 回路充足可能性判定問題の入力 $C\colon \{0,1\}^n \to \{0,1\}$ が与えられたとき, これらの変数を

$x_1, \ldots, x_n$ とする．各ゲートに対し，そのゲートに対応する変数を新たに用意し，これらを $x_{n+1}, \ldots, x_{n+s}$ とする (s は回路 C に含まれる入力以外のゲートの個数)．各 $k \in [s]$ に対し，変数 x_{n+k} に対応するゲート c の入力に対応する変数を x_i, x_j とし (もしもゲートが not ゲートであったならば x_i のみを考える)，さらに $i, j < n + k$ を満たすとする (こうなるように，入力 x に近い順にゲートの番号づけを行う)．関数 $f_k \colon \mathbb{F}^n \to \mathbb{F}_2$ を以下のように定義する: $k \in [s]$ に対し

$$f_k(x) = \begin{cases} x_k - x_i x_j & \text{if ゲート } c \text{ が AND ゲート}, \\ x_k - (x_i + x_j - x_i x_j) & \text{if ゲート } c \text{ が OR ゲート}, \\ x_k - (1 - x_i) & \text{if ゲート } c \text{ が NOT ゲート}. \end{cases}$$

$k = s + 1$ に対して $f_{s+1}(x) = 1 - x_{n+s}$ と定義する (図 1.3 を参照)．このとき，各 f_k は高々三つの変数にのみ依存するため，$(f_k)_{k \in [s+1]}$ は 3-QuadEq のインスタンスとなり，回路 C が入力として与えられたとき多項式時間で構成できる．$f_k(x) = 0$ であるという制約は，$k \leq s$ においては k 番目のゲートの出力が x_k であることを強制する．従って全ての $k \in [s]$ に対して $f_k(x) = 0$ であるならば x_{n+s} は入力 $x_1, \ldots, x_n$ に対する回路 C の出力に一致する．最後に $f_{s+1}(x) = 0 \iff x_{n+s} = 1$ であるとき，その入力が $C(x_1, \ldots, x_n) = 1$ であることを意味する．



$$x_5 = x_1 + x_2 - x_1 x_2$$
$$x_6 = 1 - x_4$$
$$x_7 = x_5 x_3$$
$$x_8 = x_5 + x_6 - x_5 x_6$$
$$x_9 = x_7 x_8$$

図 1.3 回路充足可能性判定問題から 3-QuadEq へのカープ帰着の例．(左辺) – (右辺) が各 f_k となる．

最後に C が回路充足可能性判定問題の Yes インスタンスであることと $(f_k)_{k \in [s+1]}$ が 3-QuadEq の Yes インスタンスであることが同値であることを示す．識別のため，C の入力は $x \in \{0,1\}^n$ とし，f_k の入力は $\overline{x} \in \{0,1\}^{n+s}$ とする．まず，C が Yes イン

スタンスであるとき, ある $x \in \{0,1\}^n$ が存在して $C(x) = 1$ を満たす. 出力値 $C(x)$ を計算するときの途中の各ゲートの出力値をそのまま対応する変数に割り当てることによって $\overline{x}$ を構成すると, 確かに全ての k に対して $f_k(\overline{x}) = 0$ を満たす. 一方, (f_k) が Yes インスタンスであるとき, 最初の $x = (\overline{x}_1, \ldots, \overline{x}_n)$ とすれば, $C(x) = 1$ となる. 従って, $C \mapsto (f_k)_{k\in[s+1]}$ は確かに回路充足可能性判定問題から 3-QuadEq へのカープ帰着になっている. □

Cook-Levin の定理を用いると [Kar72] のように, 一見すると回路と全く関係がない様々な問題に対して NP 完全性を示すことができる. 例えば, 3 彩色問題 3COL は NP 完全である.

PCP 定理を証明するには, 任意の $L \in \mathsf{NP}$ に対して所望の PCP 検証者を構成すれば良いが, 実際には一つの NP 完全問題 (例えば QuadEq や 3COL) を固定してそれに対する PCP 検証者を構成すれば良いことがわかる.

定理 1.2.14 （二次方程式系に対する PCP 検証者）

ある $r(n) = O(\log n), q(n) = O(1)$ に対して, 二次方程式系の判定問題 3-QuadEq は $\mathsf{PCP}(r, q)$ に属する.

演習問題 2

定理 1.2.13 と 1.2.14 を仮定して, PCP 定理 (定理 1.2.7) を証明せよ.

1.3 PCP 定理の背景: ランダムネスを用いる証明の検証

PCP 定理は近似アルゴリズム設計の困難性という, 組合せ最適化の分野への著しい応用を持つ. しかしながら元々は計算量理論における「証明の検証」の概念の捉え方の変容が背景にある. この節の内容のほとんどは [ODo] によるものであるため, 詳細はそちらを参照されたい.

定義 1.2.1 は, よく知られるように正しいとされる主張に対してはそれを確定するに足る証拠が存在し, さらにその検証もまた効率的 (多項式時間) に行えるという性質を捉えたものである. 例えば「グラフ G は 3 彩色可能である」という主張は実際に 3 彩色を証明として提示できる. 一方で「グラフ G は 3 彩色可能でない」という主張はそのような証明を提示することができるだろうか? 計算量理論の重要予想 $\mathsf{NP} \neq \mathsf{coNP}$ によれば, このような証明を提示することができないとされているがその証明は与えられていない.

そのような中, **対話型証明**, **ゼロ知識証明**, **Arthur-Merlin 証明**といった新しい証明の概念が独立同時期に複数の研究グループによって提案された Goldwasser, Micali, and Rackoff [GMR85] and Babai [Bab85]. 対話型証明は証明者が証明を提示して検証者がそれを読み取るという 1 ラウンドのプロセスを多項式個のラウンドに拡張し, さらに乱択を用いて確率的な証明の検証を行うというものである. 直感的には学術雑誌の査読と同様の形態で, 論文の著者 (証明者) と査読者 (検証者) の間で質問とその返答を繰り返して最終的に論文の正しさを確認するというものであるが, 検証者はコイントスを行うことが許されており, 実際に証明が正しい場合は高確率で受理され, 間違っている場合は高確率で拒否される. 多項式時間アルゴリズムの検証者が無限の計算能力を持つ証明者の多項式ラウンドの会話によって確率的に検証できる判定問題のクラスを IP という. ここで証明者とは関数 $P\colon \{0,1\}^* \to \{0,1\}^*$ であり, 検証者は質問 $q \in \{0,1\}^{\mathsf{poly}(n)}$ を証明者に送り, 証明者は $P(q)$ を返す. ただし関数 $q \mapsto P(q)$ の計算時間は多項式時間でなくてもよい.

なお, コイントスを行わない決定的な検証を考える場合は NP 検証と同等であることが知られているため, 対話型証明と NP 証明のギャップは本質的にはランダムネスに依拠することになる. 従って対話型証明は当初, 証明能力をそれほど大きくしないと考えられており, 前述の論文は [ODo] によれば, 理論計算機科学ではなかなか受け入れられなかったようである.

この状況の変化の兆しが見えたのは Goldreich, Micali, and Wigderson [GMW86] による**グラフ非同型性判定問題 (GNI)** に対するゼロ知識証明の構築である. この問題は, 二つのグラフの組 (G_1, G_2) が**非同型**ならば Yes インスタンスであり, 同型ならば No インスタンスであるとする問題である[*4]. 「二つのグラフは同型である」という主張は実際に同型写像を持ってくれば検証できるが, 「二つのグラフは非同型である」という不可能性の証明はどのようにすれば良いのだろうか? 以下に証明の直感的な説明を与える: 盲目の人に色の概念の証明をすることを考える. 具体的には二つのくつ下 1, くつ下 2 があり, これらの色が異なることを V さんという盲目の人に納得してもらうためにはどのようにすれば良いか? まず, 両方のくつ下の色を V さんに渡す. V さんは証明者に見えないところでコイントスを行い, どちらか一方のくつ下をランダムに選び, そのくつ下を証明者に見せてそのくつ下の番号を質問する. 証明者はそのくつ下の番号を答え, V さんは事前に聞いた説明と合致するかどうかを検証する.

- もしも両方のくつ下が同じ色だとすれば, 証明者は区別できないので当てずっぽうをするしかない. この場合の正解確率は $1/2$ である.

[*4] 二つのグラフ $G_1 = (V_1, E_1), G_2 = (V_2, E_2)$ が同型であるとは, 全単射 $\phi\colon V_1 \to V_2$ が存在して, 任意の $u, v \in V_1$ に対し, $u, v \in E_1$ と $u, v \in E_2$ が同値であることをいう.

- もしも両方のくつ下が異なる色だとすれば, 証明者は色によってくつ下の番号を当てることができる. この場合の正解確率は 1 である.

この検証を複数回繰り返せば, 同じ色であった場合の正解確率を限りなく 0 まで近づけることができる.[*5]

このことは対話によって証明の検証能力が非自明に向上することを示唆するが, 実はさらなる向上が可能であり, 実は PSPACE と呼ばれる問題クラス (多項式ビットの記憶領域を用いて解ける判定問題のクラス) は全て対話証明可能であることが Shamir [Sha02] and Lund, Fortnow, Karloff, and Nisan [LFKN02] によって示され, 対話型証明が可能な判定問題の特徴づけが与えられた (理論計算機科学では通常, 論文の著者はアルファベット順に名前を並べるが, Lund らの論文に関してはその慣例を破っている点が特徴的である).

定理 1.3.1

IP = PSPACE.

上記の定理は対話型証明の特徴づけを与えており, 特に検証にランダムネスを付与するとその検証能力が飛躍的に高まることを示唆している. そこで, この定理を「スケールダウン」することによって, クラス NP を特徴づけられないだろうか? 具体的には, ランダムネスを持つものの能力が制限された検証者によって NP を特徴づけることは可能であろうか? (能力が制限されていない時はクラス IP と一致する). クラス PCP はこのような背景の下で考えられており, 具体的には検証者がアクセスできる証明の文字数を制限することによって NP が特徴づけられる, という結果が PCP 定理である.

[*5] このアイデアはくつ下が識別できるという主張のみを証明しており, それぞれのくつ下の色の情報を証明者に伝える必要がない. このような証明を**ゼロ知識証明**という.

第 2 章

弱い PCP 定理

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 局所的な検証はなぜ可能なのか?

PCP 検証者は, オラクルアクセスとして与えられた証明のうち, 定数文字のみを見てその証明がちゃんと証明たりうるかを判定しなければならない. ここでは, 定数個の文字のみを見るという性質を検証者の局所性と呼ぶことにする.

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 線形方程式に対する局所的な検証 (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, オラクルアクセスの回数が大きくなり局所性を持たない．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

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 $\mathsf{poly}(n)$ 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

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

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

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, $2/3 < \Pr[\Phi_1 = \Phi_2] =$

$\sum_{a\in H}\Pr[\Phi=a]^2 \le \max_a \Pr[\Phi=a]\cdot\sum_{a\in H}\Pr[\Phi=a] = \max_a \Pr[\Phi=a]$ 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,

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

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

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. π をオラクルとして定理 2.1.6 のアルゴリズムを実行し, $\mathrm{dist}(\pi, \mathrm{Had}) \geq 0.01$ 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

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

を満たす．三つ目の不等号では式 (2.2) および $My \neq z$ を用いている．四つ目の不等式では $q \geq 2$ を用いている．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.

□

ここまでの議論を用いると以下のように, オラクルを用いて線形方程式の解を効率的に検証する検証者が構成できる.

補題 2.1.17 （線型方程式の解の検証）

以下を満たす定数 $q = O(1)$ と乱択 $O(mn)$ 時間オラクルアルゴリズム $V^{\pi}(M,z)$ が存在する: 入力として $M \in \mathbb{F}^{m\times n}, z \in \mathbb{F}^m$ を受け取り, オラクルとして $\pi\colon \mathbb{F}^n \to \mathbb{F}$ へのオラクルアクセスを受け取り, これらのオラクルに対して合計で高々 q 回のオラクルアクセスを行う. 任意のオラクル π に対し以下が成り立つ:

- あるベクトル $y \in \mathbb{F}^n$ に対して $\pi = \mathrm{Had}_n(y)$ であるならば, $V^{\pi}(M,z)$ は確率 1 で受理する.
- $My = z$ を満たす全てのベクトル $y \in \mathbb{F}^n$ に対して $\mathrm{dist}(\pi, \mathrm{Had}_n(y)) \geq 0.01$ ならば, $V^{\pi}(M,z)$ は確率 0.99 で拒否する.

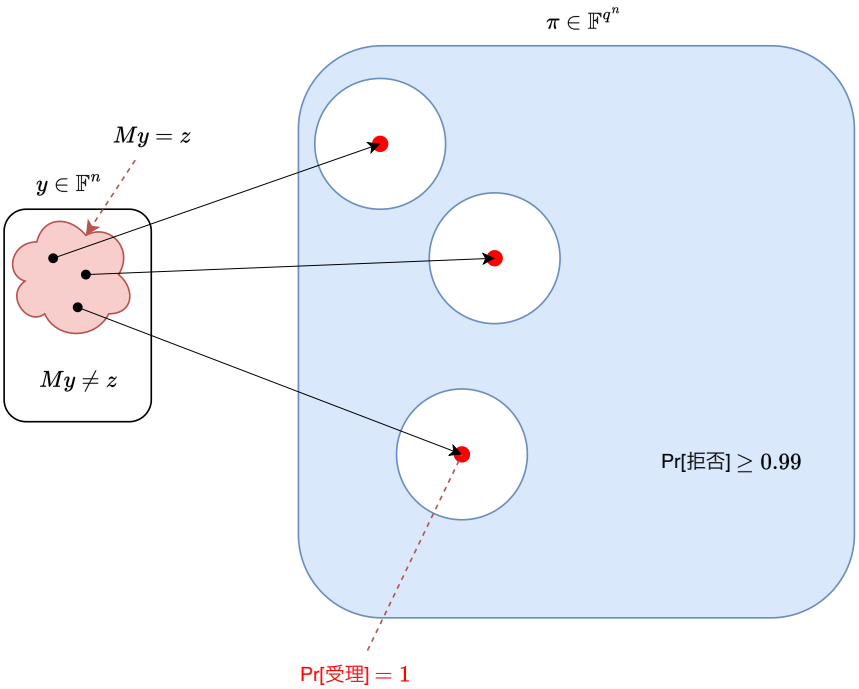


図 2.1　補題 2.1.17 の検証者のイメージ. $My = z$ を満たす y に対し $\mathrm{Had}_n(y)$ がオラクルとして与えられたときは常に受理する. 一方, そのような全ての $\mathrm{Had}_n(y)$ から離れた π は高確率で拒否する.

演習問題 5 （線型方程式の解の検証）

補題 2.1.17 を証明せよ (ヒント: 式 (2.2) と補題 2.1.14 を用いる).

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 の証明のアイデアは後に定理 1.2.7 の証明でも用いられる.

2.2.1　方針

定理 1.2.13 で定義した問題 QuadEq に対して PCP 検証者を構成すればよい. ここではより一般的に, ある $r = O(1), q = \mathsf{poly}(n)$ に対して $\mathrm{QuadEq} \in \mathsf{PCP}(r, q)$ を

示す. QuadEq のインスタンスは

$$\begin{aligned} f_1(x_1,\dots,x_n) &= 0, \\ &\vdots \\ f_m(x_1,\dots,x_n) &= 0 \end{aligned}$$

という m 個の $\mathbb{F}_2$ 上の二次連立方程式からなる.

ここで $y_{i,j} = x_i x_j$ としてこれらを並べたベクトル $y \in \mathbb{F}_2^{n^2}$ を考える (ここでは (i,j) を $[n^2]$ の元として扱う). 各 $k \in [m]$ と $i,j \in [n]$ に対し,

$$M_{k,(i,j)} = \begin{cases} 1 & \text{if } f_k(x_1,\dots,x_n) \text{ が項 } x_i x_j \text{ を含む}, \\ 0 & \text{otherwise}, \end{cases} \tag{2.4}$$

$$b_k = f_k(0) \tag{2.5}$$

によって行列 $M \in \mathbb{F}_2^{m \times n^2}$ とベクトル $b \in \mathbb{F}_2^m$ を定義すると, 連立方程式は $My = b$ と表現できる. 従って, QuadEq のインスタンス $(f_i)_{i\in[m]}$ が Yes インスタンスであることと, ある $x \in \mathbb{F}^n, y \in \mathbb{F}^{n^2}$ が存在して

条件 1 全ての $i,j \in [n]$ に対し $y_{i,j} = x_i \cdot x_j$.
条件 2 $My = b$.

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 $V^{\pi}(M,b)$ 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 $O(n^2)$ である.

2.2.2 証明の構成

有限体 $\mathbb{F}$ の要素数を q とする. PCP 検証者が受け取る証明は, 二つの関数 $\pi\colon \mathbb{F}^{n^2} \to \mathbb{F}$ と $\pi_x\colon \mathbb{F}^n \to \mathbb{F}$ からなる. 具体的には, π と π_x を表すそれぞれ長さ q^{n^2} と q^n の文字列を結合して得られる文字列が証明となる (従って全体の長さは $q^{n^2} + q^n$ である). 検証者は, **条件 1** と**条件 2** を満たす $x \in \mathbb{F}^n, y \in \mathbb{F}^{n^2}$ に対し, これらをアダマール符号で符号化したもの $\pi_x = \mathrm{Had}(x)$, $\pi = \mathrm{Had}(y)$ 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, そして確かに (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 ($z = b$ 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 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, 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 $y \in \mathbb{F}^{n^2}$ と $x \in \mathbb{F}^n$ に対して, $\mathrm{dist}(\pi, \mathrm{Had}(y)) \leq 0.01$ 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 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, $y \in \mathbb{F}^{n^2}$ を $n \times n$-行列として表現したものを $Y \in \mathbb{F}^{n \times n}$, $X = xx^\top$ 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 $x \in \mathbb{F}^n, y \in \mathbb{F}^{n^2}$ 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 $i, j \in [n]$ に対し $y_{i,j} = x_i \cdot x_j$ 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, $Y = (y_{i,j})_{i,j \in [n]} \in \mathbb{F}^{n \times n}$ と $X = xx^\top \in \mathbb{F}^{n \times n}$ 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 $V^{\pi_x, \pi_y}(1^n)$ は拒否する.

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

$$\begin{aligned} r_1^\top X r_2 &= \sum_{i,j \in [n]} r_1(i) r_2(j) x_i x_j \\ &= \sum_{i \in [n]} r_1(i) x_i \sum_{j \in [n]} r_2(j) x_j \\ &= \mathrm{Had}_n(x)(r_1) \cdot \mathrm{Had}_n(x)(r_2) \end{aligned}$$

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 $r_1, r_2 \in \mathbb{F}^n$ に対し，$\overline{r} \in \mathbb{F}^{n^2}$ を，(i,j) 番目の文字が $(r_1)_i \cdot (r_2)_j$ であるような文字列とする．このとき

$$\begin{aligned} r_1^\top Y r_2 &= \sum_{i,j \in [n]} (r_1)_i (r_2)_j y_{i,j} \\ &= \mathrm{Had}_{n^2}(y)(\overline{r}) \end{aligned}$$

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 $\mathrm{Had}_{n^2}(y)(\overline{r})$ 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．

アルゴリズム 2.2.3

PCP 検証者 $V^{\pi_x, \pi_y}(1^n)$ を以下のように記述する:

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 $r_1, r_2 \sim \mathbb{F}^n$ 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，$\mathrm{Had}_{n^2}(y)(\overline{r})$ を計算する．ここで $\overline{r} \in \mathbb{F}^{n^2}$ は，(i,j) 番目の文字が $(r_1)_i \cdot (r_2)_j$ であるような文字列である．
5. $\pi_x(r_1) \cdot \pi_x(r_2) \neq \mathrm{Had}_{n^2}(y)(\overline{r})$ ならば拒否する．
6. ステップ 2-5 を 10 回繰り返し，一度も拒否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 $1 - 1/q$ で拒否できる．従って，一回の試行で少なくとも確率 $0.96 - \frac{1}{q} \geq 0.4$ ($\because q \geq 2$) で拒否できる．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，$X \neq Y$ である時の拒否確率は任意に増幅できる． □

2.2.4 弱い PCP 定理の証明

まず，補題 2.1.17 を，線型方程式ではなく回路に対する検証に拡張する．

定理 2.2.4 （回路の検証）

以下を満たす定数 $q = O(1)$ と多項式時間乱択オラクルアルゴリズム $V^{\pi_x,\pi}(C)$ が存在する:

- 入力として回路 $C\colon \{0,1\}^n \to \{0,1\}$ を受け取り, 二つのオラクル π_x, π へのオラクルアクセスを受け取り, 全てのオラクルに対し合計で高々 q 回のオラクルアクセスを行う.
- $C(x) = 1$ を満たすベクトル $x \in \mathbb{F}_2^n$ に対して $\pi_x = \mathrm{Had}_n(x)$ であるならば, ある $\pi \in \mathbb{F}_2^{2^{n^2}}$ に対し, $V^{\pi_x,\pi}(C)$ は確率 1 で受理する.
- $C(x) = 1$ を満たす任意の $x \in \mathbb{F}_2^n$ に対して $\mathrm{dist}(\pi_x, \mathrm{Had}_n(x)) \geq 0.01$ を満たす π_x と任意の $\pi \in \mathbb{F}_2^{2^{n^2}}$ に対して $V^{\pi_x,\pi}(C)$ は確率 0.99 で拒否する.

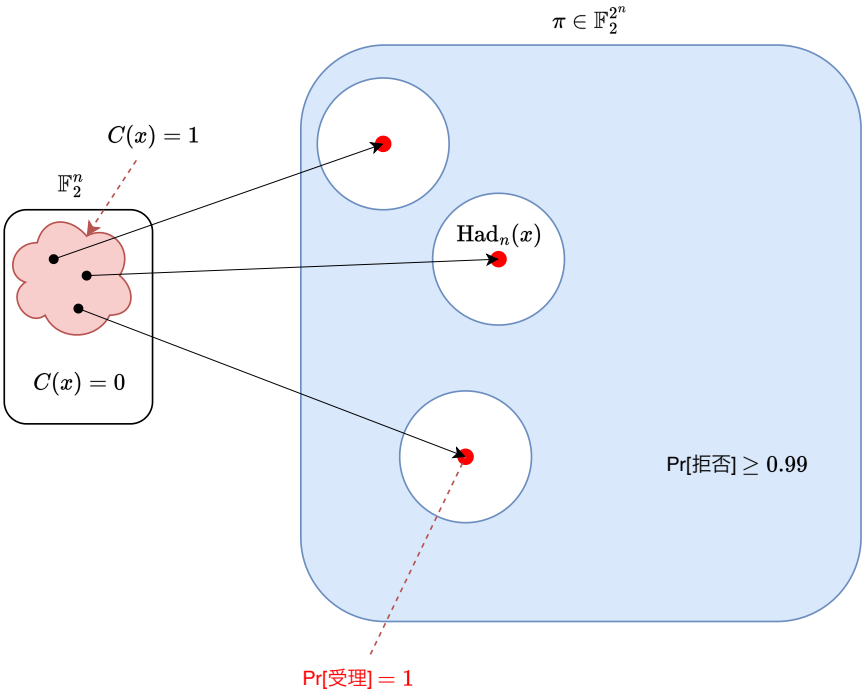


図 2.2　定理 2.2.4 の検証者のイメージ. $C(x) = 1$ を満たす $x \in \mathbb{F}_2^n$ に対し, $\pi_x = \mathrm{Had}_n(x)$ ならば, ある $\pi \in \mathbb{F}_2^{2^{n^2}}$ に対し, $V^{\pi_x,\pi}(C)$ は確率 1 で受理する. 一方, $C(x) = 1$ を満たす任意の $x \in \mathbb{F}_2^n$ に対して $\mathrm{dist}(\pi, \mathrm{Had}_n(x)) \geq 0.01$ を満たす任意の $\pi \in \mathbb{F}_2^{2^{n^2}}$ に対して, $V^{\pi_x,\pi}(C)$ は確率 0.99 で拒否する.

証明. 入力として与えられた回路 C に対し, 定理 1.2.13 で与えたカープ帰着を用いて QuadEq のインスタンス $(f_k)_{k\in[m]}$ を構成する (ここで, C のサイズ s に対し

て $m = n + s + 1$ である). さらにこの QuadEq のインスタンス $(f_k)_{k\in[m]}$ に対し, 式 (2.4) と (2.5) で定義される行列 $M \in \mathbb{F}^{m\times n^2}$ とベクトル $b \in \mathbb{F}^m$ を構成する. このとき, 任意の $x \in \mathbb{F}_2^n$ に対し, 以下は同値である:

- $C(x) = 1$ (ここでは x を $\{0,1\}^n$ の元として扱う)
- 全ての $k \in [m]$ に対し, $f_k(x) = 0$
- $y_{i,j} := x_i \cdot x_j$ で定まるベクトル $y \in \mathbb{F}_2^{n^2}$ は $My = b$ を満たす.

以下では, 三つ目の条件を検証する PCP 検証者を与える.

アルゴリズム 2.2.5

PCP 検証者 $V^{\pi_x,\pi}(M,b)$ を以下で記述する:

1. 入力として $M \in \mathbb{F}_2^{m\times n^2}$ と $b \in \mathbb{F}_2^m$ および証明として $\pi_x\colon \mathbb{F}_2^n \to \mathbb{F}_2$ と $\pi\colon \mathbb{F}_2^{n^2} \to \mathbb{F}_2$ 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_x, π のそれぞれに対して実行し, π_x, π 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.
3. 補題 2.2.2 の検証者を, オラクルとして π_x と π を用いて実行する. この検証者が拒否したら拒否する.
4. 補題 2.1.17 の検証者を, 入力として (M,b), オラクルとして π を用いて実行する. この検証者が拒否したら拒否する.
5. 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 $V^{\pi_x,\pi}$ 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 2.1.17 と 2.2.2 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.

$C(x) = 1$ となるような $x \in \mathbb{F}_2^n$ に対して $\pi_x = \mathrm{Had}_n(x)$ であるとする. このとき入力 (M,b) は, $y_{i,j} = x_i \cdot x_j$ で定まるベクトル $y \in \mathbb{F}_2^{n^2}$ に対して $My = b$ を満たす. この y に対して, $\pi = \mathrm{Had}_{n^2}(y)$ とする (補題の仮定より, $\pi_x = \mathrm{Had}_n(x)$ である). このとき,

- ステップ 2 では, π_x, π 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 3 では, 実際に全ての $i,j \in [n]$ に対して $y_{i,j} = x_i \cdot x_j$ が成り立つため, 拒否する確率は 0 である.
- ステップ 4 では, 実際に y は $My = b$ の解であるため, 拒否する確率は 0 で

ある.

以上より, $V^{\pi_x,\pi}$ は確率 1 で受理する.

次にオラクル π_x が, $C(x)=1$ を満たす任意の $x\in\mathbb{F}_2^n$ に対し

$$\mathrm{dist}(\pi_x,\mathrm{Had}_n(x))\geq 0.01$$

を満たすとする. オラクル $\pi\colon \mathbb{F}_2^{n^2}\to\mathbb{F}_2$ を任意に固定する.

- $\mathrm{dist}(\pi,\mathrm{Had}_{n^2})\geq 0.01$ または $\mathrm{dist}(\pi_x,\mathrm{Had}_n)\geq 0.01$ ならば, 検証者はステップ 2 において確率 0.99 で拒否する. 以降では π,π_x はどちらも線形関数に十分近いとし, π に最も近い線形関数を $\mathrm{Had}_{n^2}(y)$ とし, π_x に最も近い線形関数を $\mathrm{Had}_n(x)$ とする. このとき, π_x の仮定から $C(x)=0$ である.
- ステップ 3 は, 補題 2.2.2 の仮定が満たされているため, ある $i,j\in[n]$ に対して $y_{i,j}\neq x_i\cdot x_j$ となるならば, 確率 0.99 で拒否する. 従って以降ではさらに $x\in\mathbb{F}^n, y\in\mathbb{F}^{n^2}$ が, 全ての $i,j\in[n]$ に対し $y_{i,j}=x_i\cdot x_j$ を満たすとする.
- ステップ 4 は, $C(x)=0 \iff My\neq b$ であるため, 補題 2.1.17 の検証者が確率 0.99 で拒否する.

以上より, $V^{\pi_x,\pi}$ がステップ 2-4 のループで拒否する確率は少なくとも 0.97 である. □

実は定理 2.2.4 から即座に弱い PCP 定理が示せる.

定理 *2.2.1* の証明. 定理 1.2.11 より, 回路充足可能性判定問題は NP 完全である. 従って, この問題に対する PCP 検証者を与えればよい. 実は, 定理 2.2.4 の検証者がこの条件を満たす.

実際, 入力として与えられた回路 $C\colon\{0,1\}^n\to\{0,1\}$ が Yes インスタンスならば, ある $x\in\{0,1\}^n$ が存在して $C(x)=1$ となる. この x を $\mathbb{F}_2^n$ の元とみなして, $\pi_x=\mathrm{Had}_n(x)$ とすれば, ある π が存在して $V^{\pi_x,\pi}(C)$ は確率 1 で受理する.

一方, C が No インスタンスならば, $C(x)=1$ となるような $x\in\{0,1\}^n$ は存在しないため, どのような π_x,π に対しても, $V^{\pi_x,\pi}(C)$ は確率 0.99 で拒否する. □

注釈 2.2.6 （近接 PCP）

定理 2.2.4 のように, 入力として回路 C が与えられ, さらに C が受け取る入力 x を符号化したもの π_x および「追加の情報」をもつ π へのオラクルアクセスが与えられたとき, x が C の充足割り当てを符号化したものから遠い場合に高確率で拒否するような検証者は一般に充足可能性の検証よりも強いこ

とを主張しており, これを達成するような証明のことを**近接 PCP** (PCP of Proximity) と呼び, 検証者のことを PCPP 検証者と呼ぶ.

2.3　二入力回路の割り当てに対する PCP 検証者

定理 2.2.4 はある意味で, $x \in \{0,1\}^n$ を全て見ることなく, 与えられた回路 C に対し $C(x) = 1$ となるかどうかを検証する PCP 検証者を構成した (実際には $\pi_x = \mathrm{Had}_n(x)$ および追加の情報 π_y へのオラクルアクセスも仮定している).

ここでは回路 C が二つの入力 $x, y \in \{0,1\}^n$ を受け取る場合でも同様の PCPP 検証者が構成できることを示す. この設定は人工的な問題設定に見えるが, PCP 定理の証明において重要な役割を果たす.

定理 2.3.1 （二入力回路の割り当てに対する PCP 検証者）

以下を満たす定数 $q = O(1)$ と多項式時間乱択オラクルアルゴリズム $V^{\pi_x,\pi_y,\pi}(C)$ が存在する:

- 入力として $C\colon \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ および三つのオラクル π_x, π_y, π へのオラクルアクセスを受け取り, これらのオラクルに対して合計で高々 q 回のオラクルアクセスを行う.
- $C(x,y) = 1$ を満たす $x, y \in \mathbb{F}_2^n$ に対して $\pi_x = \mathrm{Had}_n(x)$ および $\pi_y = \mathrm{Had}_n(y)$ であるならば, ある $\pi \in \mathbb{F}_2^{2^{2n}+2^{4n^2}}$ が存在して, $V^{\pi_x,\pi_y,\pi}(C)$ は確率 1 で受理する.
- $C(x,y) = 1$ を満たす任意の $x, y \in \mathbb{F}_2^n$ に対して $\mathrm{dist}(\pi_x, \mathrm{Had}_n(x)) \geq 0.01$ または $\mathrm{dist}(\pi_y, \mathrm{Had}_n(y)) \geq 0.01$ を満たす任意の π_x, π_y および任意の π に対して, $V^{\pi_x,\pi_y,\pi}(C)$ は確率 0.99 で拒否する.

(x,y) を一つの文字列 $z \in \mathbb{F}_2^{2n}$ とみなして定理 2.2.4 を適用したとき, 検証者は $\pi_z = \mathrm{Had}_{2n}(z)$ と追加の情報 π へのオラクルアクセスを持つことになるが, 定理 2.3.1 の検証者は, x, y それぞれの符号化 π_x, π_y へのオラクルを持つという点が異なっている.

定理 2.3.1 の証明においては, ある文字列 $x \in \{0,1\}^n$ が別の文字 $\overline{x} \in \{0,1\}^{n+m}$ の接頭辞かどうかを (オラクルアクセスを用いて) 局所的に検証することを意味する以下の補題を用いる.

補題 2.3.2 （接頭辞の検証）

以下を満たす定数 $q = O(1)$ と多項式時間乱択オラクルアルゴリズム $V_0^{\pi_x,\pi_{\overline{x}}}(1^n, 1^m)$ が存在する:

- オラクル $\pi_x, \pi_{\overline{x}}$ は次が保証されている: ある $x \in \mathbb{F}_2^n$ と $\overline{x} \in \mathbb{F}_2^{n+m}$ に対して, $\mathrm{dist}(\pi_x, \mathrm{Had}_n(x)) < 0.01$ および $\mathrm{dist}(\pi_{\overline{x}}, \mathrm{Had}_{n+m}(\overline{x})) < 0.01$.
- 検証者 $V_0^{\pi_x,\pi_{\overline{x}}}$ は高々 q 回のオラクルアクセスを行う.
- $\pi_x = \mathrm{Had}_n(x)$, $\pi_{\overline{x}} = \mathrm{Had}_{n+m}(\overline{x})$ であって, さらに $\overline{x}$ の先頭 n 文字が x と一致するならば, $V_0^{\pi_x,\pi_{\overline{x}}}$ は確率 1 で受理する.
- 一方, $\overline{x}$ の先頭 n 文字が x に一致しない場合, 確率 0.99 で拒否する.

証明. アルゴリズム $V_0^{\pi_x,\pi_{\overline{x}}}$ は単純で, 以下のように記述できる:

1. 一様ランダムなベクトル $r \sim \mathbb{F}_2^n$ を選び, $\overline{r} = (r, 0^m) \in \mathbb{F}_2^{n+m}$ とする.
2. $\pi_x(r) \neq \pi_{\overline{x}}(\overline{r})$ ならば拒否する. ここで, $\pi_{\overline{x}}(\overline{r})$ は補題 2.1.14 を用いて計算する.
3. ステップ 1-2 を 10 回繰り返し, 一度も拒否しなければ受理して終了する.

$\overline{x}$ の最初の n 文字を $y \in \{0,1\}^n$ とする. もしも $\pi_x = \mathrm{Had}_n(x)$, $\pi_{\overline{x}} = \mathrm{Had}_{n+m}(\overline{x})$ であって, $y = x$ ならば, ステップ 2 においてどのような $r \sim \mathbb{F}_2^n$ を選んでも $\mathrm{Had}_n(x)(r) = \mathrm{Had}_n(y)(r) = \mathrm{Had}_{n+m}(\overline{x})(r, 0^m)$ となり, 確率 1 でアルゴリズムは受理する. 一方で $x \neq y$ の場合を考える. オラクル $\pi_{\overline{x}}$ は $\mathrm{Had}(\overline{x})$ に 0.01-近いため, 補題 2.1.14 より, ステップ 2 において, 確率 0.98 で $\pi_{\overline{x}}(\overline{x}) = \mathrm{Had}_n(y)(r)$ を計算できる. また, r が一様ランダムなので, 確率 0.99 で $\pi_x(r) = \mathrm{Had}_n(x)(r)$ である. 従って一度の試行で少なくとも確率 0.97 で検証者は拒否するため, ステップ 1-2 を 10 回だけ繰り返せば少なくとも確率 0.99 で検証者は拒否する. □

注釈 2.3.3 （接尾辞の検証）

補題 2.3.2 のアルゴリズムは x が $\overline{x}$ の接頭辞かどうかの検証を行うが, 同様のアルゴリズム (0^m と r の順番を反転させるだけ) を考えれば接尾辞の検証にも利用できる.

定理 *2.3.1* の証明. 三つ目のオラクル $\pi \in \{0,1\}^{2^{2n}+2^{4n^2}}$ を, 以下の二つの関数の組として解釈する:

- $\pi_1 \colon \mathbb{F}_2^{2n} \to \mathbb{F}_2$. この関数は, ベクトル $\overline{x} := (x, x') \in \mathbb{F}_2^{2n}$ に対して $\pi_1 =$

$\mathrm{Had}_{2n}(\overline{x})$ であることを想定する.

- $\pi_2 \colon \mathbb{F}_2^{4n^2} \to \mathbb{F}_2$. この関数は, 上記のベクトル $\overline{x}$ に対し $\overline{y}_{i,j} = \overline{x}_i \cdot \overline{x}_j$ で定まるベクトル $\overline{y} \in \mathbb{F}_2^{4n^2}$ に対し, $\pi_2 = \mathrm{Had}_{4n^2}(\overline{y})$ であることを想定する.

なお, 実際には π_1, π_2 は関数としては解釈できるが, その関数が線形関数であるとは限らない. アルゴリズム 2.2.5 を少し修正した以下の検証者 $V^{\pi_x,\pi_y,\pi}(C)$ を考える:

アルゴリズム 2.3.4

PCP 検証者 $V^{\pi_x,\pi_y,\pi}(C)$ を以下で記述する:

1. オラクル π を二つの関数 $\pi_1 \colon \mathbb{F}_2^{2n} \to \mathbb{F}_2, \pi_2 \colon \mathbb{F}_2^{4n^2} \to \mathbb{F}_2$ の組として解釈する.
2. 入力で与えられた回路を $C \colon \{0,1\}^{2n} \to \{0,1\}$ とみなし, 定理 2.2.4 の証明で考えた行列 $M \in \mathbb{F}_2^{m \times (2n)^2}$ とベクトル $b \in \mathbb{F}_2^m$ を計算する.
3. 定理 2.1.6 のアルゴリズムを $\pi_x, \pi_y, \pi_1, \pi_2$ に対して実行し, π_1, π_2 がそれぞれ線形関数に 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. $\mathrm{Had}_n(x), \mathrm{Had}_n(y), \mathrm{Had}_{2n}(z)$ をそれぞれ π_x, π_y, π_1 に最も近い線形関数とする.
5. $z \in \mathbb{F}_2^{2n}$ の前半の n 文字が x, 後半の n 文字が y に一致するかどうかをそれぞれ補題 2.3.2 のアルゴリズムを用いて検証する. もしいずれかの検証においてこのアルゴリズムが拒否したら拒否する.
6. 補題 2.2.2 の検証者を, オラクルとして π_1 と π_2 を用いて実行する. この検証者が拒否したら拒否する.
7. 補題 2.1.17 の検証者を, 入力として (M, b), オラクルとして π_2 を用いて実行する. この検証者が拒否したら拒否する.
8. 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.

$C(x,y) = 1$ を満たす $x, y \in \mathbb{F}_2^n$ に対し $\pi_x = \mathrm{Had}_n(x)$, $\pi_y = \mathrm{Had}_n(y)$ とする. このとき, 文字列 x, y を結合して得られる文字列を $z = (x,y) \in \mathbb{F}_2^{2n}$, また文字列 $w \in \mathbb{F}_2^{4n^2}$ を, $w_{i,j} = z_i \cdot z_j$ と定め, 最後に $\pi = (\mathrm{Had}_{2n}(z), \mathrm{Had}_{4n^2}(w))$ とすると, $V^{\mathrm{Had}_n(x),\mathrm{Had}_n(y),\pi}(C)$ は確率 1 で受理する.

一方, オラクル π_x, π_y が, $C(x,y) = 1$ を満たす任意の $x, y \in \mathbb{F}_2^n$ に対して, $\mathrm{dist}(\pi_x, \mathrm{Had}_n(x)) \geq 0.01$ または $\mathrm{dist}(\pi_y, \mathrm{Had}_n(y)) \geq 0.01$ の少なくとも一方が成り立つとする. 任意にオラクル π_1, π_2 を固定する.

- ステップ 3 では全てのオラクルが線形関数に近いかどうかを確認しており, もしもどれかが線形関数から 0.01-遠いならば, 定理 2.1.6 より確率 0.99 で検証者は拒否する. それぞれに最も近い文字列を $x \in \mathbb{F}_2^n, y \in \mathbb{F}_2^n, z \in \mathbb{F}_2^{2n}$ とする.
- ステップ 4 では $z = (x, y)$ かどうかが確認される. もしも $z \neq (x, y)$ ならば確率 0.99 で拒否される.
- ステップ 5 以降は定理 2.2.4 と同一であり, 特に π_1 が C の任意の充足割り当てから 0.01-遠い場合は確率 0.99 で検証者は拒否する. さらにこのとき, 任意の充足割り当て $(x^*, y^*) \in C^{-1}(1)$ に対して, $\mathrm{dist}(x, x^*) \geq 0.01$ もしくは $\mathrm{dist}(y, y^*) \geq 0.01$ が成り立つ (演習問題 6).

□

演習問題 6 （二つに分割したベクトルの距離）

ベクトル $z \in \mathbb{F}_2^{2n}$ に対し, その前半 n 個の成分からなる部分ベクトルを $\mathrm{head}(z) \in \mathbb{F}_2^n$, 後半 n 個の成分からなる部分ベクトルを $\mathrm{tail}(z) \in \mathbb{F}_2^n$ とする. 集合 $S \subseteq \mathbb{F}_2^{2n}$ を任意に固定する. もしもベクトル z が $\mathrm{dist}(z, S) \geq \varepsilon$, すなわち, 任意の $s \in S$ に対して $\mathrm{dist}(z, s) \geq \varepsilon$ を満たすならば, 任意の $s \in S$ に対して

- $\mathrm{dist}(\mathrm{head}(z), \mathrm{head}(s)) \geq \varepsilon$
- $\mathrm{dist}(\mathrm{tail}(z), \mathrm{tail}(s)) \geq \varepsilon$

の少なくとも一方が成り立つことを示せ.

第 3 章

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 $\varphi = (X, \Sigma, \mathcal{I}, \mathcal{C})$ をインスタンス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. また, 変数の個数と制約の個数の和 $n + m$ を φ の**サイズ**といい, $\mathrm{size}(\varphi)$ と表す.
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

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

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.
- 辺の順序が固定された辺集合 $E = \{e_1, \dots, e_m\}$ に対し, $\mathcal{I} = (e_1, \dots, e_m)$ 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.

例 3.1.3 （二次方程式系の充足可能性判定問題）

定理 1.2.13 で定義した判定問題 3-QuadEq は 3-CSP である. 3-QuadEq のインスタンス $f_1, \dots, f_m\colon \mathbb{F}_2^n \to \mathbb{F}_2$ に対し, 変数を $x_1, \dots, x_n$ とし, $f_i(x) = 0$ という制約を考えると, 各 f_i の出力は高々三つの変数の値にのみ依存するため, これは 3-CSP である.

以後は, 例 3.1.3 のように, k-CSP の定義を与える際は, $calI$ や $calC$ を明示せずに全ての変数と制約を記述することによって与えることもある.

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.

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)

表 3.1　PCP と CSP の対応関係

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.

定理 3.1.4 （PCP 定理の CSP 版）

ある関数 $K(s) = s^{O(1)}$, 定数 $c \in \mathbb{N}$, $q \in \mathbb{N}$, $\alpha > 0$, 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が存在する: 3-QuadEq のインスタンス $\varphi_0 = (f_k)$ を入力として受け取り, q-CSP のインスタンス $\varphi = (X, \Sigma, \mathcal{I}, \mathcal{C})$ を出力する. このインスタンス φ は, 与えられた φ_0 のサイズが $s = \mathrm{size}(\varphi_0)$ であるとき, $\mathrm{size}(\varphi) = K(s)$, $|\Sigma| = c$ となる. さらにこのインスタンス φ は

- 入力 φ_0 が $\mathrm{UNSAT}(\varphi_0) = 0$ である (すなわち 3-QuadEq の Yes インスタンスである) とき, $\mathrm{UNSAT}(\varphi) = 0$ となる.
- 入力 φ_0 が $\mathrm{UNSAT}(\varphi_0) > 0$ である (すなわち 3-QuadEq の No インスタンスである) とき, $\mathrm{UNSAT}(\varphi) \geq \alpha$ となる.

注釈 3.1.5

一般に m 個の制約からなる CSP のインスタンス φ_0 が No インスタンスであることは $\mathrm{UNSAT}(\varphi_0) \geq \frac{1}{m}$ と同値である. ところが定理 3.1.4 では, 変換によって得られる CSP のインスタンス φ のサイズが元の CSP のインスタンス φ_0 のサイズ s に**よらない定数になる**点が重要である.

PCP 検証者の構成と CSP の変換が同値であることを証明する.

補題 3.1.6 （PCP 検証者と CSP の変換の同値性）

定理 3.1.4 と定理 1.2.14 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 3.1.4⇒ 定理 1.2.14 の証明. ある $r = O(\log n)$, $q' = O(1)$ に対して, 3-QuadEq 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. 検証者はまず入力として 3-QuadEq のインスタンス $\varphi_0 = (f_k)$ を受け取り, 定理 3.1.4 のアルゴリズムを用いて q-CSP のインスタンス φ を計算する. また, 証拠 π をこの 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 $V^\pi(\varphi_0)$ 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 φ の制約 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 c_i が読み込んだ割り当てによって充足**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 が定数であるため, 全体で読み込む証明のビット数は $q' \cdot \lceil \log_2 c \rceil = O(1)$ ことに留意する. 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 $O(\log K(\mathrm{size}(\varphi_0)) = O(\log \mathrm{size}(\varphi_0))$ となる.

もし φ_0 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 $V^\pi(\varphi_0) = 1$ となる. もし φ_0 が No インスタンスならば,

$\mathrm{UNSAT}(\varphi) \geq \alpha$ 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 $1-\alpha$ である．よって，この操作を $\lceil 10/\alpha \rceil = O(1)$ 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 1.2.14⇒ 定理 3.1.4 の証明. 仮定より，3-QuadEq 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 $\varphi_0 = (f_k)$ に対して，全てのランダムシード $s \in \{0,1\}^r$ を列挙して，それぞれに対して「$V^\pi(\varphi_0; s) = 1$」という制約を加えて得られる CSP を出力する．各 $V^\pi(\varphi_0; s)$ 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 3.1.4 の証明に注力する．

3.1.2 制約グラフ

PCP 定理の証明，すなわち定理 3.1.4 の証明は，まず 3-CSP の特殊ケースである 3-QuadEq を 2-CSP に変換し，この 2-CSP に対して様々な変換を繰り返し施すことによって得られる．2-CSP は各制約が高々一つの変数への割り当てのみに依存するものであるため，その制約をグラフとして記述することができる．このような 2-CSP のグラフ表現を**制約グラフ**という．

定義 3.1.7 （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 3.1.8 （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.

制約グラフは 2-CSP を表現するためのものであるが, 一般の k-CSP もまた制約グラフとして表現することができる.

補題 3.1.9 （k-CSP の制約グラフ表現）

任意の $k \in \mathbb{N}$ に対し, ある定数 $c > 0$ および多項式時間アルゴリズム A が存在して次が成り立つ: k-CSP のインスタンス $\varphi = (X, \Sigma, \mathcal{I}, \mathcal{C})$ を入力として受け取り, 以下を満たす制約グラフ $G = \langle (V, E), \Sigma^k, \mathcal{C}' \rangle$ を出力する.

- $\mathrm{size}(G) \leq ck \cdot (|X| + |\mathcal{I}|)$
- $\mathrm{UNSAT}(\varphi) = 0$ ならば $\mathrm{UNSAT}(G) = 0$.
- $\mathrm{UNSAT}(\varphi) > 0$ ならば $\mathrm{UNSAT}(G) \geq \frac{1}{k} \cdot \mathrm{UNSAT}(\varphi)$.

証明. 入力として与えられた k-CSP の変数集合 $X = \{\mathrm{x}_1, \ldots, \mathrm{x}_n\}$ および制約集合 $\mathcal{C} = (c_I)_{I \in \mathcal{I}}$ に対し, 新たに構成する制約グラフは次のような構造を持つ: φ の各変数 x_i に対し頂点 x_i を用意し, 各制約 c_I $(I \in \mathcal{I})$ に対し頂点 y_I を用意する. 変数 x_i が制約 c_I に含まれる, すなわち $\mathrm{x}_i \in I$ ならば, 二頂点 x_i と y_I の間に辺を引いて得られる二部グラフを (V, E) とする. 簡単のため, 以下では各 $I \in \mathcal{I}$ に対して $|I| = k$ とする (そうでない制約が存在するならば, そのような各制約は同じ変数を複数回読み込むことを許し, 対応する辺は多重辺とすることで, 読み込み回数をちょうど k にする). この二部グラフは, 各 y_i の次数はちょうど k であり, サイズは (k のみに依存する) 定数倍で抑えられる.

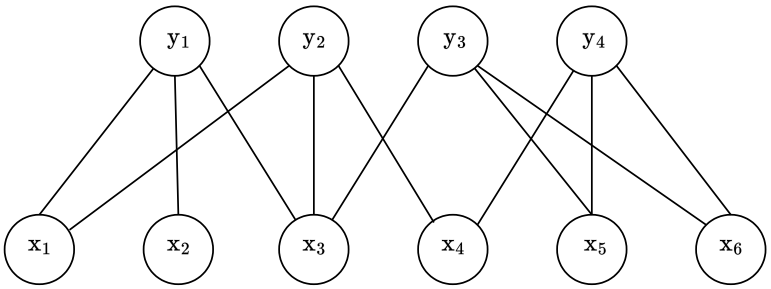


図 3.1　制約グラフの構成例. 例えば, 変数 y_1 に対応する制約は変数 $\mathrm{x}_1, \mathrm{x}_2, \mathrm{x}_3$ に依存している.

新たな制約グラフのアルファベットは Σ^k とする. 各辺 $e = \{x_i, y_I\}$ に対応する制約 c'_e は, 割り当て $x_i \leftarrow \vec{\sigma} = (\sigma_1, \ldots, \sigma_k) \in \Sigma^k$, $y_I \leftarrow \vec{\tau} = (\tau_1, \ldots, \tau_k) \in \Sigma^k$ が次の条件を満たすとき, かつその時に限り充足される: y_I の隣接頂点を $x_{i_1}, \ldots, x_{i_k}$ $(i_1 < \cdots < i_k)$ としたとき,

- 部分割り当て $(\mathrm{x}_{i_1} \leftarrow \tau_1, \ldots, \mathrm{x}_{i_k} \leftarrow \tau_k)$ が c_I を充足する.
- $\tau_i = \sigma_1$

ここでは, x_i への割り当てのうち, $\sigma_2, \ldots, \sigma_k$ は無視されていることに留意されたい.

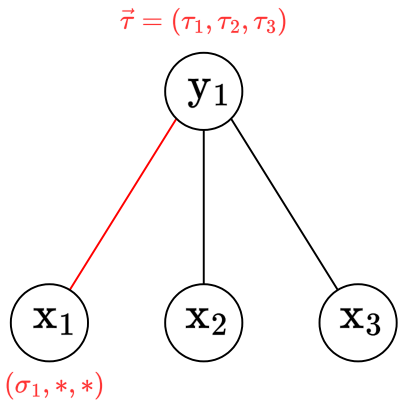


図 3.2　割り当ての例．ここでは, (τ_1, τ_2, τ_3) が y_1 に対する制約を充足し, さらに $\sigma_j = \tau_j$ $(j = 1, 2, 3)$ であれば, 辺 $\{y_1, x_j\}$ に対応する制約グラフの制約が満たされる．

元の k-CSP のインスタンス φ が充足可能, すなわち $\mathrm{UNSAT}(\varphi) = 0$ ならば, その充足割り当て $a\colon X \to \Sigma$ に対し, 制約グラフ G の割り当て $a'\colon V \to \Sigma^k$ を

$$a'(x_i) = (a(x_i), \underbrace{*, \ldots, *}_{k-1}) \quad (i \in [n]),$$
$$a'(y_I) = (a(x_{i_1}), \ldots, a(x_{i_k}), \sigma_1) \quad (I = \{i_1, \ldots, i_k\} \in \mathcal{I}).$$

と定める ($*$ は Σ の任意の元でよい)．この割り当て a' は G の全ての制約を満たすため, $\mathrm{UNSAT}(G) = 0$ である．

逆に, G の任意の割り当て $a'\colon V \to \Sigma^k$ に対し, φ の割り当て $a\colon X \to \Sigma$ を

$$a(x_i) = a'(x_i)_1 \quad (i \in [n])$$

と定める．割り当て a' が頂点 y_I に接続する全ての辺 $\{x_i, y_I\}$ $(\forall i \in I)$ に対応する G の制約を充足するならば, 割り当て a は制約 c_I を満たす．対偶をとると, a が c_I を満たさないならば, a' は少なくとも 1 つの辺 $\{\{x_i, y_I\}\}_{i \in I}$ に対応する制約が充足されない．すなわち $\mathrm{UNSAT}(a'; G) \geq \frac{1}{k} \cdot \mathrm{UNSAT}(a; \varphi)$ である．従って, 割り当て a' を $\mathrm{UNSAT}(a'; G) = \mathrm{UNSAT}(G)$ となるように選ぶと

$$\mathrm{UNSAT}(G) = \mathrm{UNSAT}(a'; G) \geq \frac{\mathrm{UNSAT}(a; \varphi)}{k} \geq \frac{\mathrm{UNSAT}(\varphi)}{k}$$

より, 主張を得る．

□

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 7

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, $1 = \lambda_1 \geq \cdots \geq \lambda_n \geq -1$ に対して P^k の固有値は $1 = \lambda_1^k \geq \cdots \geq \lambda_n^k \geq -1$ となる. 今, $\max\{|\lambda_2|, |\lambda_n|\} \leq \lambda$ であるため, $\max\{|\lambda_2^k|, |\lambda_n^k|\} \leq \lambda^k$ 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 $n \geq 2$ 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.

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．

3.3　制約グラフの定数次数エクスパンダー化

ここでは，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できることを示す．

補題 3.3.1 （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．

3.3.1　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 3.3.2 （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.

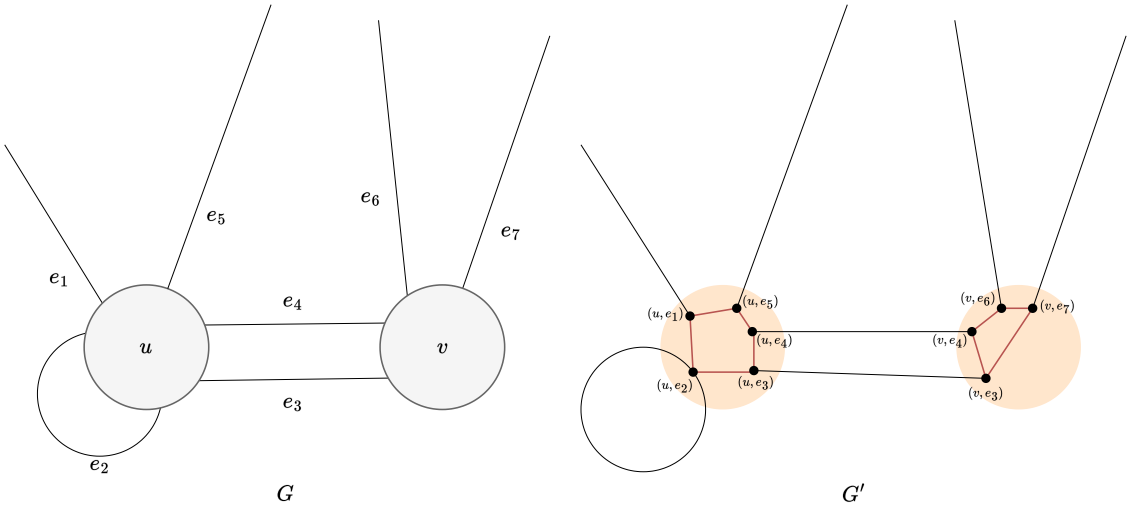


図 3.3　次数削減変換の例. 橙色の内部の頂点集合がクラウドであり, クラウド内の辺は定理 3.2.8 によって構成されるが, ここでは図の簡単のため X_d を長さ d の閉路としている. G' における黒い辺の制約は対応する元のグラフの辺と同一の制約とし, クラウド内の赤辺の制約は等式制約とする.

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 $\sum_{u \in V} \frac{d_0 \deg_G(u)}{2} \leq d_0 |E|$ 本である (ここで式 (3.1) を用いた). 従って

$$|E'| = |E_{\text{inner}}| + |E_{\text{outer}}| \leq (d_0 + 1)|E| \tag{3.2}$$

を満たす. また, V' の要素数は次数の総和に等しいため, $|V'| \leq 2|E|$ 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 $\mathrm{UNSAT}(G') \geq c' \cdot \mathrm{UNSAT}(G)$ となる定数 $c' > 0$ が存在すること示す. 定数 $c' > 0$ を

$$c' = \min \left\{ \frac{(1 - \lambda_0) d_0}{8(d_0 + 1)}, \frac{1}{2(d_0 + 1)} \right\} \tag{3.3}$$

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 3.4). 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.

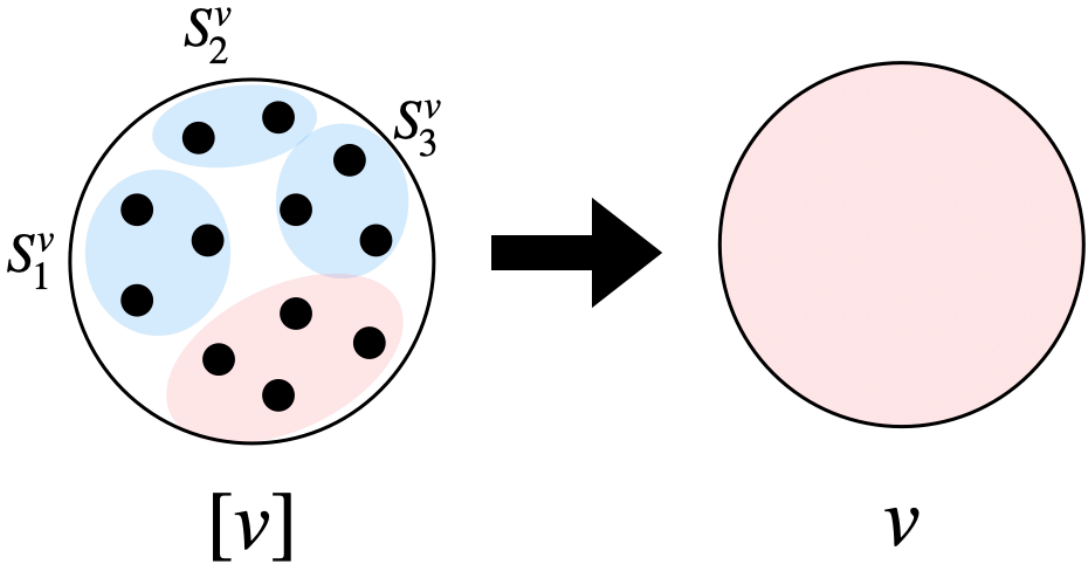


図 3.4　多数決によって選ばれなかった v-クラウド内の頂点の集合を $S_v \subseteq [v]$ とする.

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 $S_\sigma^v \subseteq [v]$ に対するエクスパンダー混交補題 (補題 3.2.4) より,

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

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_σ^v と $[v] \setminus S_\sigma^v$ 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

$$\begin{aligned}
|F'| &\geq \frac{1}{2} \sum_{v,\sigma} W(S_\sigma^v, [v] \setminus S_\sigma^v) \\
&\geq \frac{(1-\lambda_0)d_0}{4} \sum_{v,\sigma} |S_\sigma^v| \\
&= \frac{(1-\lambda_0)d_0}{4} |S| \\
&\geq \frac{(1-\lambda_0)d_0}{8} \mathrm{UNSAT}(a;G) \cdot |E| && \because \text{ケース 1 の仮定} \\
&\geq \frac{(1-\lambda_0)d_0}{8(d_0+1)} \mathrm{UNSAT}(a;G) \cdot |E'|. && \because \text{式 (3.2)}
\end{aligned}$$

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

$$\begin{aligned}
|F'| &\geq |F| - |S| \\
&\geq |E|\mathrm{UNSAT}(a;G) - \frac{\mathrm{UNSAT}(a;G)}{2}|E| && \because \text{ケース 2 の仮定} \\
&= \frac{\mathrm{UNSAT}(a;G)}{2}|E| \\
&\geq \frac{\mathrm{UNSAT}(a;G)}{2(d_0+1)}|E'| && \because \text{式 (3.2)}
\end{aligned}$$

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 (3.3) 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. □

3.3.2 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 3.3.3 （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

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

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

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

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 3.3.1 を示す準備が整った.

補題 *3.3.1* 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 3.3.2 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 3.3.2 の主張より, d-正則である. 次に, G_1 を入力として補題 3.3.3 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 3.3.3 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 3.3.2 と 3.3.3 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. □

第 4 章

PCP 定理の証明

本チャプターは PCP 定理の証明を与える.

4.1 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. 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）

任意のアルファベット Σ に対し, ある二つの定数 $c = c(|\Sigma|) > 0, \alpha = \alpha(|\Sigma|) \in (0,1)$, アルファベット Σ_0, 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 $G' = \langle (V',E'), \Sigma_0, \mathcal{C}' \rangle$ 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 3.1.4) の証明は, まず NP 完全問題である 3-QuadEq を補題 3.1.9 を用いて制約グラフに変換し, その制約グラフに対して補題 4.1.1 を $O(\log s)$ 回だけ繰り返し適用することによって UNSAT の値を定数まで増幅することで得られる.

4.2　制約グラフのべき乗

補題 3.3.1 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, そのような定数次数エクスパンダー性を持つ制約グラフに対し, 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 4.2.1 （制約グラフの冪乗）

パラメータ $\ell \in \mathbb{N}$ 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 $G^\ell = \langle (V, \boldsymbol{E}), \Sigma^{d^\ell}, \mathcal{C}^\ell \rangle$ を, 以下のようにして定義する:

- 頂点集合は同じ V とする. なお, V の元は $v_1 < \cdots < v_n$ のように順序づけられているとする.
- グラフ $(V, \boldsymbol{E})$ をべき乗 (定義 3.2.6) によって得られるグラフ $(V, E)^\ell$ とする. すなわち, 各二頂点 $u, v \in V$ に対し, 長さ ℓ の uv-路の個数と同じだけ uv 間に多重辺を用意する (uv-路と vu 路の個数は一致するため, 無向グラフとして定義できる). 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.
- アルファベット集合は Σ^{d^ℓ} とする. ここで, 頂点 u に $\vec{\sigma} = (\sigma_1, \ldots, \sigma_{d^\ell}) \in \Sigma^{d^\ell}$ 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 ℓ 以内の頂点の集合を $\Gamma(u)$ とし[a], その元を頂点番号の大小順で並べて $\Gamma(u) = \{v_1, \ldots, v_\ell\}$ とする (ここで G の d-正則性より $\ell \leq d^\ell$ である). このとき, $\vec{\sigma}$ は各 $v \in \Gamma(u)$ に $\vec{\sigma}_u \in \Sigma$ を割り当てる写像とみなし, $\vec{\sigma}_v$ を u **の** v **に対する意見**と呼ぶ. なお, $\ell < d^\ell$ の場合は相異なる二つの $\vec{\sigma}, \vec{\sigma}' \in \Sigma^{d^\ell}$ が同一の写像としてみなされることもある (これらは制約を充足するか否かにおいては区別されない).
- 辺 $\boldsymbol{e} = \{u, v\} \in \boldsymbol{E}$ に対応する制約 $\boldsymbol{c_e} \in \mathcal{C}^\ell$ は次の二つを満たす割り当て $(\vec{\sigma}, \vec{\sigma}') \in \Sigma^{d^\ell} \times \Sigma^{d^\ell}$ によって満たされる: 元のグラフ G の全ての辺 $e = \{s, t\} \in E$ (ただし $s < t$) に対し,
  - $s \in \Gamma(u), t \in \Gamma(v)$ ならば $(\vec{\sigma}(s), \vec{\sigma}'(t)) \in c_e$ が成り立つ.
  - $s \in \Gamma(v), t \in \Gamma(u)$ ならば $(\vec{\sigma}'(s), \vec{\sigma}(t)) \in c_e$ が成り立つ.

[a] 頂点 v が頂点 u から距離 i 以内であるというのは, 頂点 u から開始し頂点 v に至る長さ i 以下の路が存在することをいう.

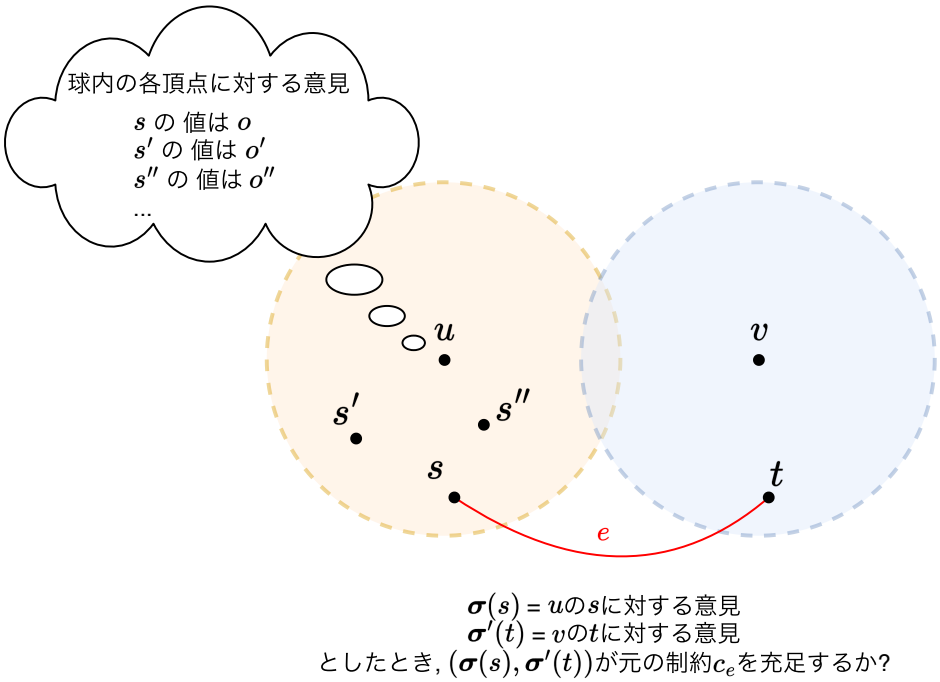


図 4.1　制約グラフ G のべき乗 G^ℓ の図例. 頂点 u, v にそれぞれ $\vec{\sigma}, \vec{\sigma}' \in \Sigma^{d^\ell}$ が割り当てられているとき, それぞれ関数 $\vec{\sigma}\colon \Gamma(u) \to \Sigma$ と $\vec{\sigma}'\colon \Gamma(v) \to \Sigma$ とみなす. なお, 元のグラフ G が自己ループを持つことから, 距離 ℓ 以内の全ての頂点に対して意見が定義される.

パラメータ ℓ 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. べき乗によって得られる制約グラフは元の制約グラフに対して UNSAT の値が増幅する.

補題 4.2.2 （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: 十分大きいパラメータ $\ell = \ell(d, |\Sigma|, \lambda) \leq |E|$ 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 $G^\ell = \langle (V, \boldsymbol{E}), \Sigma^\ell, \mathcal{C}^\ell \rangle$ を, 定義 4.2.1 で定義する. このとき

- $\mathrm{size}(G^\ell) \leq \mathrm{size}(G) \cdot d^\ell$ である.
- $\mathrm{UNSAT}(G) = 0$ ならば $\mathrm{UNSAT}(G^\ell) = 0$ である.

- $\mathrm{UNSAT}(G^\ell) \geq \beta\sqrt{\ell} \cdot \min\left\{\mathrm{UNSAT}(G), \frac{1}{\ell}\right\}$ である.

注釈 4.2.3 （「定数」の意味）

主張や証明の中でしばし「定数」という言葉を多用するが, ここでは $d, \lambda, |\Sigma|$ をユニバーサルな定数で固定し, これらの値に応じて定まる値 β, ℓ なども定数と呼ぶ. ただし, 定数はグラフの頂点数や辺数には依存しない.

証明. 定義より $|\boldsymbol{E}| \leq |V|d^\ell$ であるから, $\mathrm{size}(G^\ell) \leq \mathrm{size}(G) \cdot d^\ell$ である.

また, $\mathrm{UNSAT}(G) = 0$ ならば, 全制約を満たす割り当て $a\colon V \to \Sigma$ に対し, $a'(u)$ に

$$\vec{\sigma}(u)\colon w \mapsto a(w)$$

で定まる関数 $\vec{\sigma}\colon \Gamma(w) \to \Sigma$ とみなせる値 $\vec{\sigma} \in \Sigma^{d^\ell}$ を割り当てる (候補が複数存在する場合はどれでもよい). このようにして定まる $a'\colon V \to \Sigma^\ell$ を考えると, これは全ての制約を満たす割り当てであるため, $\mathrm{UNSAT}(G^\ell) = 0$ である.

最後に $\mathrm{UNSAT}(G^\ell)$ の下界を証明する. 簡単のため ℓ は偶数であるとする (奇数の場合は $\ell/2$ を $\lfloor \ell/2 \rfloor$ に置き換えれば同じ証明が成立する). 頂点 $u \in V$ に対し, V 値をとる確率変数 $\mathsf{RW}_j(u)$ を, グラフ G 上で頂点 u から開始する長さ j のランダムウォークの最終到達頂点とする (長さとは辿った辺の本数のことである). べき乗の制約グラフ G^ℓ における割り当て $a'\colon V \to \Sigma^\ell$ に対し, 元の制約グラフに対する任意の割り当て $a\colon V \to \Sigma$ を

$$a(u) = \operatorname*{argmax}_{\tau \in \Sigma} \left\{\Pr\left[a'(\mathsf{RW}_{\ell/2}(u))_u = \tau\right]\right\}. \tag{4.1}$$

によって定める. 常に $\mathsf{RW}_{\ell/2}(u) \in \Gamma(u)$ であるから $a'(\mathsf{RW}(u))_u$ は必ず存在する. すなわち, u から開始したランダムウォークが最も「拾いやすい」意見を $a(u)$ としている. このようにして定めた割り当て $a\colon V \to \Sigma$ は, 適当な定数 $\beta = \beta(\lambda, d, |\Sigma|) > 0$ に対して

$$\mathrm{UNSAT}(a'; G^\ell) \geq \beta\sqrt{\ell} \cdot \min\left\{\mathrm{UNSAT}(a; G), \frac{1}{\ell}\right\}. \tag{4.2}$$

を満たすことを後で示す. $\mathrm{UNSAT}(a'; G^\ell) = \mathrm{UNSAT}(G^\ell)$ を満たす a' に対して式 (4.2) を適用し, さらに $\mathrm{UNSAT}(a; G) \leq \mathrm{UNSAT}(G)$ を代入すると $\mathrm{UNSAT}(G^\ell)$ の所望の下界が得られて証明は完了する.

■式 (4.2) の証明.　割り当て $a' \colon V \to \Sigma^{d^\ell}$ を固定し, a を式 (4.1) によって定める. 辺部分集合 $F_0 \subseteq E$ を, a によって充足されない辺の集合とし, さらにその部分集合 $F \subseteq F_0$ を,

- $|F_0| \leq \frac{|E|}{\ell}$ ならば $F_0 = F$,
- そうでない場合は $|F| = \left\lfloor \frac{|E|}{\ell} \right\rfloor$ となるように F を任意に固定する.

このとき, 任意の $x \geq 1$ に対し $\lfloor x \rfloor \geq x/2$ であることから, $\frac{|E|}{\ell} \geq 1$ の仮定を用いて

$$\begin{aligned} |F| &\geq \min\left\{ |F_0|, \left\lfloor \frac{|E|}{\ell} \right\rfloor \right\} \\ &\geq \frac{1}{2} \min\left\{ |F_0|, \frac{|E|}{\ell} \right\} \\ &\geq \frac{|E|}{2} \min\left\{ \mathrm{UNSAT}(a; G), \frac{1}{\ell} \right\} \end{aligned} \tag{4.3}$$

を得る (なお, $|E| \geq \ell$ の仮定を用いるのはここのみである).

べき乗グラフの辺 $\boldsymbol{e} \in \boldsymbol{E}$ は G 上の長さ ℓ の路に対応する. 従って $\boldsymbol{e} = (u_0, \ldots, u_\ell)$ (ただし全ての $i \in [\ell]$ に対し $\{u_{i-1}, u_i\} \in E$) と表し, 元のグラフの辺と区別するため $e \in E$ を G 辺, $\boldsymbol{e} \in \boldsymbol{E}$ を G^ℓ 辺と呼ぶことにする.

G^ℓ 辺 $\boldsymbol{e} = (u_0, \ldots, u_\ell)$ の i 番目の辺 $e_i := \{u_{i-1}, u_i\}$ は

- $e_i \in F$
- $a'(u_0)_{u_{i-1}} = a(u_{i-1})$ かつ $a'(u_\ell)_{u_i} = a(u_i)$

をどちらも満たすとき, **悪い**ということにする. 一様ランダムに $\boldsymbol{e}$ を選んだとき, $B_i \in \{0, 1\}$ を辺 e_i が悪いならば $B_i = 1$, そうでないならば $B_i = 0$ とする (これらは確率変数である). さらに

$$I = \left[\frac{\ell}{2} - \sqrt{\ell}, \frac{\ell}{2} + \sqrt{\ell} \right] \cap \mathbb{Z}$$

に対し, $B = \sum_{i \in I} B_i$ とする. もしも $B > 0$ ならば, 辺 $\boldsymbol{e}$ の制約 $c'_{\boldsymbol{e}}$ は a' によって充足されない. 従って, $\mathrm{UNSAT}(a'; G^\ell) \geq \Pr_{\boldsymbol{e} \sim \boldsymbol{E}}[B \geq 1]$ である.

Paley-Zygmund の不等式 (補題 A.1.3) より, $\mathbb{E}[B]$ および $\mathbb{E}[B^2]$ を評価することで $\Pr[B \geq 1]$ の下界を得ることができる. これらは以下のように評価できる.

主張 4.2.4 （B の期待値）

ある十分小さな定数 $C = C(d, \lambda, |\Sigma|) > 0$ が存在して, B の期待値は

$$\mathbb{E}[B] \geq C\sqrt{\ell} \cdot \frac{|F|}{|E|}.$$

主張 4.2.5 （B^2 の期待値）

ある十分大きな定数 $C' = C'(d, \lambda, |\Sigma|) > 0$ が存在して, B^2 の期待値は

$$\mathbb{E}[B^2] \leq C'\sqrt{\ell} \cdot \frac{|F|}{|E|}.$$

これらの主張の証明は後で与える. ここでは, これらを用いて式 (4.2) を示す. 実際, 補題 A.1.3 と主張 4.2.4 と 4.2.5 より, 定数 $\beta = \frac{C^2}{2C'} > 0$ に対して

$$\begin{aligned}
\mathrm{UNSAT}(a'; G^\ell) &\geq \Pr[B \geq 1] \\
&\geq \frac{\mathbb{E}[B]^2}{\mathbb{E}[B^2]} \\
&\geq \frac{C^2}{C'}\sqrt{\ell} \cdot \frac{|F|}{|E|} \\
&\geq \beta\sqrt{\ell} \cdot \min\left\{\mathrm{UNSAT}(a; G), \frac{1}{\ell}\right\} \qquad \because \text{式 (4.3)}
\end{aligned}$$

となり主張を得る.

□

最後に残された二つの主張の証明を与える. なお, 制約グラフのエクスパンダー性を用いるのは主張 4.2.5 の証明のみである.

主張 *4.2.4* の証明. 各 $i \in I$ に対して $\mathbb{E}[B_i] = \Pr_{\boldsymbol{e} \sim \boldsymbol{E}}[e_i \text{ が悪い}]$ を下から抑えればよい. 一様ランダムな $\boldsymbol{e} = (u_0, \ldots, u_\ell) \sim \boldsymbol{E}$ は G 上の長さ ℓ のランダムウォークであるから, 各 $i \in I$ に対して $\boldsymbol{e}$ は以下のプロセスによって生成される:

1. 辺 $\{u_{i-1}, u_i\} \sim E$ を一様ランダムに選ぶ.
2. 頂点 u_{i-1} から開始して長さ $i-1$ のランダムウォークを行い, 辿った頂点を順番に $u_{i-1}, u_{i-2}, \ldots, u_0$ とする.
3. 同様に, 頂点 u_i から開始して長さ $\ell - i$ のランダムウォークを行い, 辿った頂点を順番に $u_i, u_{i+1}, \ldots, u_\ell$ とする.
4. 頂点列 $(u_0, \ldots, u_\ell)$ に対応する $\boldsymbol{e} \in \boldsymbol{E}$ を出力する.

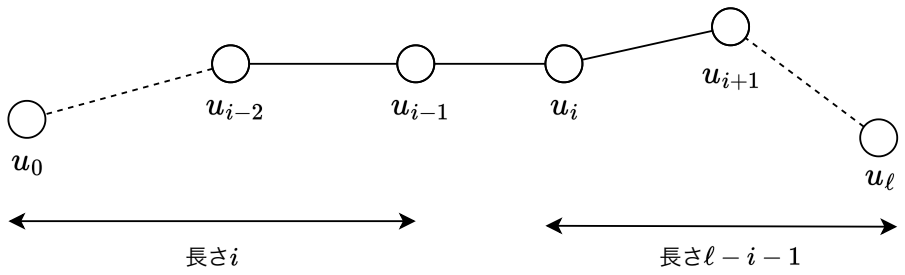


図 4.2 ランダムウォークの図例. 頂点 u_0 から開始して長さ $i-1$ のランダムウォークを行い, 辿った頂点を順番に $u_{i-1}, u_{i-2}, \ldots, u_0$ とする. 同様に, 頂点 u_i から開始して長さ $\ell - i$ のランダムウォークを行い, 辿った頂点を順番に $u_i, u_{i+1}, \ldots, u_\ell$ とする. この二つのランダムウォークを組み合わせて, 長さ ℓ のランダムウォークを生成する.

このプロセスに基づいて, $\Pr[B_i = 1]$ を評価する. まず, ステップ 1 で選ばれた辺が $\{u_{i-1}, u_i\} \in F$ である確率は $\frac{|F|}{|E|}$ である. 次にステップ 1 で選ばれた辺 $\{u_{i-1}, u_i\} \in F$ で条件つけたとき, ステップ 2 と 3 は独立なランダムネスに基づく試行であるため, それぞれの両端点 u_0, u_ℓ は独立な確率変数である. 従って

$$\begin{aligned}
&\Pr[B_i = 1] \\
&\quad = \mathop{\mathbb{E}}_{\text{step 1-3}}\left[\{u_{i-1}, u_i\} \in F \text{ and } a'(u_0)_{u_{i-1}} = a(u_{i-1}) \text{ and } a'(u_\ell)_{u_i} = a(u_i)\right] \\
&\quad = \frac{|F|}{|E|} \cdot \mathop{\mathbb{E}}_{\{u_{i-1}, u_i\} \sim F}\left[\underbrace{\Pr_{\text{step 2}}[a'(u_0)_{u_{i-1}} = a(u_{i-1})]}_{=:p_0} \cdot \underbrace{\Pr_{\text{step 3}}[a'(u_\ell)_{u_i} = a(u_i)]}_{=:p_\ell}\right]
\end{aligned} \tag{4.4}$$

を得る. 二つ目の等式では, $\{u_{i-1}, u_i\} \in F$ で条件つけて $\{u_{i-1}, u_i\} \sim E$ をランダムに選ぶというのは, F 内の辺を一様ランダムに選ぶことに他ならないことに留意する.

次に, 任意の辺 $\{u_{i-1}, u_i\} \in F$ を固定して p_0 と p_ℓ を評価する. 頂点 u_{i-1} を任意に固定して p_0 を考える (p_ℓ も同様に示せる). 頂点 $u \in V$ および非負整数 $m \geq 0$ に対し, 確率変数 $X_{u,m} \in \Sigma$ を, (a' における)$\mathsf{RW}_m(u)$ の u に対する意見, すなわち $X_{u,m} = a'(\mathsf{RW}_m(u))_u$ とする. 式 (4.1) より, $m = \ell/2$ のとき,

$$\Pr\left[X_{u_{i-1}, \ell/2} = a(u_{i-1})\right] \geq \frac{1}{|\Sigma|} \tag{4.5}$$

である. ステップ 2 で選ばれた端点 u_0 は $u_0 = \mathsf{RW}_i(u_{i-1})$ であるから, $X_{u_{i-1}, i} = a'(\mathsf{RW}_i(u_{i-1}))_{u_{i-1}} = a'(u_0)_{u_{i-1}}$ である. 従って,

$$p_0 = \Pr\left[X_{u_{i-1}, i} = a(u_{i-1})\right] \tag{4.6}$$

である．任意に頂点 $u \in V$ と $i \in I$ を固定して $X_{u,\ell/2}$ と $X_{u,i}$ の分布を比較する．$i \in I$ なので，$|i - \ell/2| \leq \sqrt{\ell}$ である．

元のグラフ G の各頂点に対し，その頂点が持つ自己ループの一つを赤く塗る (仮定より元のグラフ (V, E) は d-正則かつ各頂点が自己ループを少なくとも一つ持つ)．頂点 u から開始する長さ $\ell/2$ のランダムウォークを考え，確率変数 $Y_{\ell/2}$ を，このランダムウォークが辿った赤**以外**の辺の個数とする (Y_i も同様に定義する)．各頂点がちょうど一つの赤い自己ループを持つため，確率変数 $Y_{\ell/2}$ の分布は始点 u に依存せず二項分布 $\mathrm{Bin}(\ell/2, 1 - 1/d)$ である (同様に Y_i の分布は $\mathrm{Bin}(i, 1 - 1/d)$ である)．また，$\mathsf{RW}'_k(u)$ を，頂点 u から開始する赤い辺を選ばない長さ k のランダムウォークの最終到達頂点とする (すなわち，G の各頂点から赤い辺を除いて得られるグラフ上の長さ k のランダムウォークである)．従って，任意の頂点部分集合 $A \subseteq V$ に対し

$$\begin{aligned} \Pr\left[\mathsf{RW}_i(u) \in A\right] &= \sum_{k=0}^{i} \Pr\left[\mathsf{RW}_i(u) \in A \mid Y_i = k\right] \cdot \Pr\left[Y_i = k\right] \\ &= \sum_{k=0}^{i} \Pr\left[\mathsf{RW}'_k(u) \in A\right] \cdot \Pr\left[Y_i = k\right] \end{aligned} \tag{4.7}$$

が成り立つ．ここで，以下の主張を後で示す:

主張 4.2.6

十分大きな定数 $C_1 = C_1(d, |\Sigma|) > 0$ と十分小さな定数 $C_2 = C_2(d, |\Sigma|) > 0$ が存在して以下が成り立つ: $K := \left\{k \in \mathbb{Z}_{\geq 0} \colon \left|k - \left(1 - \frac{1}{d}\right)\frac{\ell}{2}\right| \leq C_1\sqrt{\ell}\right\}$ としたとき，

1. $\Pr\left[Y_{\ell/2} \notin K\right] \leq \frac{1}{2|\Sigma|}$.
2. 全ての $k \in K$ と全ての $i \in I$ に対し，$\Pr\left[Y_i = k\right] \geq C_2 \cdot \Pr\left[Y_{\ell/2} = k\right]$.

主張 4.2.6 を仮定し，その定数 C_1, C_2 を用いて主張 4.2.4 を示す．パラメータ ℓ は d に依存して十分大きくとる．このとき，任意の $i \in I$ に対して $K \subseteq [0, \ell/2 - \sqrt{\ell}] \subseteq$

$[0, i]$ が成り立つから

$$\begin{aligned}
\Pr\left[\mathsf{RW}_i(u) \in A\right] &= \sum_{k=0}^{i} \Pr\left[\mathsf{RW}'_k(u) \in A\right] \Pr\left[Y_i = k\right] \\
&\geq \sum_{k \in K} \Pr\left[\mathsf{RW}'_k(u) \in A\right] \underbrace{\Pr\left[Y_i = k\right]}_{\geq C_2 \Pr\left[Y_{\ell/2} = k\right]} \\
&\geq C_2 \sum_{k \in K} \Pr\left[\mathsf{RW}'_k(u) \in A\right] \Pr\left[Y_{\ell/2} = k\right] \\
&\geq C_2 \left(\sum_{0 \leq k \leq \ell/2} \Pr\left[\mathsf{RW}'_k(u) \in A\right] \Pr\left[Y_{\ell/2} = k\right] - \Pr\left[Y_{\ell/2} \notin K\right] \right) \\
&\geq C_2 \left(\Pr\left[\mathsf{RW}_{\ell/2}(u) \in A\right] - \frac{1}{2|\Sigma|} \right) \qquad (4.8)
\end{aligned}$$

を得る．ここで頂点部分集合 $A \subseteq V$ を $A = \{v \in V \colon a'(v)_u = a(u)\}$ として式 (4.8) を適用すると

$$\Pr\left[X_{u,i} = a(u)\right] = \Pr\left[a'(\mathsf{RW}_i(u))_u = a(u)\right] = \Pr\left[\mathsf{RW}_i(u) \in A\right] \geq \frac{C_2}{2|\Sigma|}$$

となるため，$u = u_{i-1}$ とすると，式 (4.6) より

$$p_0 = \Pr\left[X_{u_{i-1},i} = a(u_{i-1})\right] \geq \frac{C_2}{2|\Sigma|}$$

を得る．同様に，p_ℓ についても，適当な定数 $C'_2 > 0$ に対して $p_\ell \geq \frac{C'_2}{2|\Sigma|}$ が成り立つため，式 (4.4) より，適当な定数 $C_3 = C_3(d, |\Sigma|) > 0$ に対して

$$\Pr[B_i = 1] \geq \frac{|F|}{|E|} \cdot \mathop{\mathbb{E}}_{\{u_{i-1},u_i\} \sim F} [p_0 \cdot p_\ell] \geq C_3 \cdot \frac{|F|}{|E|}$$

を得る．最後に $B = \sum_{i \in I} B_i$ より，$\mathbb{E}[B] \geq C_3 \sqrt{\ell} \cdot \frac{|F|}{|E|}$ を得る． □

主張 *4.2.6* の証明．一つ目の主張は，$Y_{\ell/2}$ の分布が二項分布 $\mathrm{Bin}(\ell/2, 1 - 1/d)$ であることと Chebyshev の不等式 (補題 A.1.2) を用いれば示せる (演習問題 8)．

二つ目の主張を示す．すなわち，二つの二項分布 $\mathrm{Bin}(i, 1-1/d)$ と $\mathrm{Bin}(\ell/2, 1-1/d)$ を比較する．一つ目の主張を満たすように任意に定数 $C_1 > 0$ を固定する．このとき，二つ目の主張が成り立つようにうまく $C_2 > 0$ を選べることを示せばよい．

一般に，任意の $m \in \mathbb{N}$，定数 $p \in (0, 1)$，$C > 0$ および $k \in \mathbb{N}$ (ただし $|k - mp| \leq C\sqrt{m}$ を満たす) に対し，

$$\Pr\left[\mathrm{Bin}(m, p) = k\right] = \frac{1}{\sqrt{2\pi mp(1-p)}} \exp\left(-\frac{(k-mp)^2}{2mp(1-p)} \right) \cdot \left(1 \pm O_{C,p}\left(\frac{1}{m} \right) \right) \qquad (4.9)$$

が成り立つ (De Moivre-Laplace の定理). 特に, $|k - mp| \leq C\sqrt{m}$ を満たすため, 右辺は $\Theta(1/\sqrt{m})$ で評価できる.

従って, 適当な定数 $D_1, D_2 > 0$ が存在して, ℓ が十分大きいとき,

$$\Pr[Y_i = k] = \Pr[\mathrm{Bin}(i, 1 - 1/d) = k] \geq \frac{D_1}{\sqrt{i}},$$
$$\Pr\left[Y_{\ell/2} = k\right] = \Pr[\mathrm{Bin}(\ell/2, 1 - 1/d) = k] \leq \frac{D_2}{\sqrt{\ell/2}}$$

を満たす. $i \in I$ より, 主張を得る. □

演習問題 8 （主張の証明）

主張 4.2.6 の一つ目の主張, すなわちある十分大きな定数 $C_1 = C_1(d, |\Sigma|) > 0$ が存在して

$$\Pr\left[Y_{\ell/2} \notin K\right] \leq \frac{1}{2|\Sigma|}$$

が成り立つことを示せ.

主張 *4.2.5* の証明. ランダムウォーク $\boldsymbol{e} = (u_0, \ldots, u_\ell) \sim \boldsymbol{E}$ に対し, 確率変数 Z_i を, $\{u_{i-1}, u_i\} \in F$ ならば $Z_i = 1$, そうでなければ $Z_i = 0$ とする. このとき, $B_i \leq Z_i$ であるから, $Z := \sum_{i \in I} Z_i$ に対し $B \leq Z$ であり, 特に $\mathbb{E}[B^2] \leq \mathbb{E}[Z^2]$ である. 以下, $\mathbb{E}[Z^2]$ を評価する. まず, $\mathbb{E}[Z_i] = \Pr[\{u_{i-1}, u_i\} \in F] = \frac{|F|}{|E|}$ である (ランダムウォークの i 番目の辺の周辺分布は E 上の一様分布だから). 定義より

$$\begin{aligned}\mathbb{E}[Z^2] &= \sum_{i,j \in I} \mathbb{E}[Z_i Z_j] = \mathbb{E}[Z] + 2 \sum_{i,j \in I, i<j} \mathbb{E}[Z_i Z_j] \\ &= \frac{|F|}{|E|}|I| + 2 \sum_{i,j \in I, i<j} \mathbb{E}[Z_i Z_j]. \end{aligned} \tag{4.10}$$

ここで, 固定した $i < j$ に対して $\mathbb{E}[Z_i Z_j]$ を上から評価する. すなわち, λ-エクスパンダーグラフ上の長さ ℓ のランダムウォーク $(u_0, \ldots, u_\ell)$ を考えたとき, i 番目の辺 $\{u_{i-1}, u_i\}$ と j 番目の辺 $\{u_{j-1}, u_j\}$ が同時に F に含まれる確率を評価する. 辺部分集合 $F \subseteq E$ の辺に接続している頂点部分集合を $V_F := \bigcup_{e \in F} e$ とすると, ランダムウォーク $(u_{i-1}, \ldots, u_j)$ に関する確率について,

$$\Pr\left[\{u_{i-1}, u_i\} \in F \text{ and } \{u_{j-1}, u_j\} \in F\right] \leq \Pr\left[u_{i-1} \in V_F \text{ and } u_j \in V_F\right] \tag{4.11}$$

となる. ここでは, 頂点 u_{i-1} を一様ランダムに選び, そこから開始する長さ $j - i + 1$ のランダムウォーク $(u_{i-1}, u_i, \ldots, u_{j-1}, u_j)$ に関する確率を考えている. このランダ

ムウォークは, べき乗グラフ G^{j-i+1} 上の長さ 1 のランダムウォークと一致する. 補題 3.2.7 より, G^{j-i+1} は d^{j-i+1}-正則かつ λ^{j-i+1}-エクスパンダーである. 従って, 補題 3.2.4 より, 式 (4.11) の右辺は

$$\begin{aligned}\Pr\left[u_{i-1} \in V_F \text{ and } u_j \in V_F\right] &= \frac{W(V_F, V_F)}{nd^{j-i+1}} \\ &\leq \left(\frac{|V_F|}{n}\right)^2 + \lambda^{j-i+1} \cdot \frac{|V_F|}{n} \\ &\leq \left(\frac{2|F|}{n}\right)^2 + \lambda^{j-i+1} \cdot \frac{2|F|}{n} \\ &= \frac{d|F|^2}{|E|^2} + \frac{\lambda^{j-i+1} d|F|}{|E|}\end{aligned}$$

を得る. ここで, $|F|/|E| \leq 1/\ell$ および $|I| \leq 2\sqrt{\ell}$ を用いると, ある定数 $C = C(d, \lambda) > 0$ が存在して

$$\begin{aligned}\sum_{i,j \in I, i<j} \mathbb{E}[Z_i Z_j] &\leq d\,\underbrace{|I|^2 \left(\frac{|F|}{|E|}\right)^2}_{\leq 4|F|/|E|} + \frac{d|F|}{|E|} \sum_{i,j \in I, i<j} \lambda^{j-i+1} \\ &\leq 4d\frac{|F|}{|E|} + d|I| \cdot \frac{|F|}{|E|} \cdot \sum_{i \geq 0} \lambda^i \\ &\leq C|I|\frac{|F|}{|E|}.\end{aligned}$$

これと式 (4.10) を組み合わせると, 主張を得る. □

4.3　アルファベット削減

ギャップ増幅補題 (補題 4.1.1) では変換前後で制約グラフのアルファベットは同一であることを主張しているが, べき乗をとることによって, UNSAT の値は $\sqrt{\ell}$ 倍に増幅する一方でアルファベットは Σ^ℓ になってしまう. そこで, アルファベット Σ^ℓ 上の制約グラフ G を, UNSAT や size をそれほど損なわずアルファベット Σ 上の制約グラフ G' に変換する必要がある.

定理 4.3.1 （アルファベット削減）

ある定数 $\gamma > 0$, アルファベット Σ_0, および多項式時間アルゴリズム A が存在して以下が成り立つ: アルゴリズム A は制約グラフ $G = \langle (V, E), \Sigma, C \rangle$ を入力として受け取り, 別の制約グラフ $G' = \langle (V', E'), \Sigma_0, C' \rangle$ を出力する. た

だし G' は, ある定数 $c = c(|\Sigma|) > 0$ に対して

- $\mathrm{size}(G') \leq c \cdot \mathrm{size}(G)$
- $\mathrm{UNSAT}(G) = 0$ ならば $\mathrm{UNSAT}(G') = 0$
- $\mathrm{UNSAT}(G) > 0$ ならば $\mathrm{UNSAT}(G') \geq \gamma \cdot \mathrm{UNSAT}(G)$

定理 4.3.1 の証明では, まず, あるアルファベット Σ_1 と定数 $k \in \mathbb{N}$ が存在して, 与えられた制約グラフ $G = \langle (V, E), \Sigma, \mathcal{C} \rangle$ を, アルファベット Σ_1 上の k-CSP $\varphi = \langle (X, \Sigma_1, \mathcal{I}, \mathcal{C}') \rangle$ に変換する多項式時間アルゴリズム A' が存在することを示す. 次にこの k-CSP を補題 3.1.9 を用いて $\Sigma_0 := \Sigma_1^k$ 上の制約グラフ $G' = \langle (V', E'), \Sigma_0, \mathcal{C}' \rangle$ に変換することによって証明は完了する.

補題 4.3.2

ある定数 $q_0 \in \mathbb{N}$ と多項式時間アルゴリズム A' が存在して以下が成り立つ: 任意のアルファベット Σ に対し, ある定数 $c = c(|\Sigma|) > 0$ が存在し, アルゴリズム A' は制約グラフ $G = \langle (V, E), \Sigma, C \rangle$ を入力として受け取り, q_0-CSP $\varphi = (X, \{0,1\}, \mathcal{I}, \mathcal{C}')$ を出力する. ただし, φ は

- $|X| \leq c \cdot \mathrm{size}(G)$, $|\mathcal{I}| \leq c \cdot \mathrm{size}(G)$.
- $\mathrm{UNSAT}(G) = 0$ ならば $\mathrm{UNSAT}(\varphi) = 0$.
- $\mathrm{UNSAT}(G) > 0$ ならば $\mathrm{UNSAT}(\varphi) \geq 0.99 \cdot \mathrm{UNSAT}(G)$.

証明. $b = \lceil \log_2 |\Sigma| \rceil$ とし, アルファベット Σ を $\{0,1\}^b$ と同一視する ($|\Sigma|$ が 2 べキでない場合はダミーの値を追加して $|\Sigma| = \{0,1\}^b$ とし, 各制約はダミーの値を読み込んだ場合は常に拒否するようにすればよい). 制約グラフ G の各辺 $e \in E$ に対応する制約 $c_e \subseteq \Sigma^2$ を指示関数 $c_e \colon \Sigma^2 \to \{0,1\}$ と同一視し, これを二入力の論理回路として表現したものを $C_e \colon \{0,1\}^b \times \{0,1\}^b \to \{0,1\}$ とする (任意の論理関数 $f \colon \{0,1\}^\ell \to \{0,1\}$ はサイズ $O(\ell 2^\ell)$ の論理回路で表現できるため, この回路はサイズが高々 $2^{O(b)}$ であるが, $b = \log_2 |\Sigma|$ は定数として扱うのでこの回路のサイズも定数である).

各辺 $e = \{u, v\} \in E$ に対し, このようにして得られた各 C_e に対して定理 2.3.1 を適用する. この定理から得られる検証者 $V^{\pi_u, \pi_v, \pi_e}(C_e)$ に対し, ランダムシード $r \in \{0,1\}^{\mathsf{poly}(b)}$ を固定したときの決定的な検証者を $V^{\pi_u, \pi_v, \pi_e}(C_e; r)$ とする. なお, V の時間計算量は高々 $\mathsf{poly}(b) \leq T$ であるとする. これは以下の性質を持つ:

- $C_e(\sigma_u, \sigma_v) = 1$ を満たす $\sigma_u, \sigma_v \in \{0,1\}^b$ に対して $\pi_u = \mathrm{Had}_b(\sigma_u)$ およ

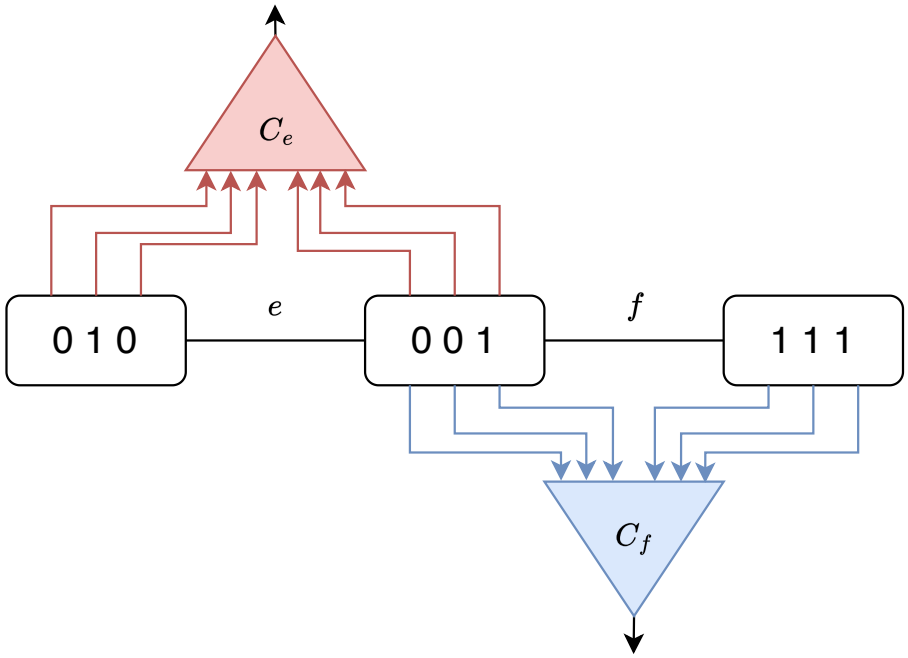


図 4.3　2-CSP の各辺の制約を論理回路として表現したもの.

び $\pi_v = \mathrm{Had}_b(\sigma_v)$ であるならば, ある $\pi_e \in \mathbb{F}_2^{2^{2b}+2^{4b^2}}$ が存在して, 全ての $r \in \{0,1\}^T$ に対して $V^{\pi_u,\pi_v,\pi_e}(C_e; r)$ は受理する.

- $C_e(\sigma_u, \sigma_v) = 1$ を満たす任意の $\sigma_u, \sigma_v \in \{0,1\}^b$ に対して

$$\mathrm{dist}(\pi_u, \mathrm{Had}_b(\sigma_u)) \geq 0.01 \text{ or } \mathrm{dist}(\pi_v, \mathrm{Had}_b(\sigma_v)) \geq 0.01$$

を満たす任意の π_u, π_v および任意の $\pi_e \in \mathbb{F}_2^{2^{2b}+2^{4b^2}}$ に対して,

$$\Pr_r \left[V^{\pi_u,\pi_v,\pi_e}(C_e; r) \text{ rejects}\right] \geq 0.99.$$

- ある定数 $q = O(1)$ が存在して検証者 $V^{\pi_u,\pi_v,\pi_e}(C_e)$ は高々 q 回のオラクルアクセスを行う. なお, 最終的に構成する CSP は q-CSP となる.

最終的に出力する新しい q_0-CSP φ を以下のように構成する.

- 各 $u \in V$ と各 $i \in [2^b]$ に対し, $\pi_{u,i} \in \mathbb{F}_2$ を変数とする. また, $\pi_u = (\pi_{u,i})_{i\in[2^b]} \in \mathbb{F}_2^{2^b}$ とする. 同様に, 各 $e \in E$ と各 $j \in \left[2^{2b} + 2^{4b^2}\right]$ に対し, $\pi_{e,j} \in \mathbb{F}_2$ を変数とする. また, $\pi_e = (\pi_{e,j})_{j\in[2^{2b}+2^{4b^2}]} \in \mathbb{F}_2^{2^{2b}+2^{4b^2}}$ とする. 全体の変数集合は $X = \left\{\pi_{u,i} \colon u \in V, i \in [2^b]\right\} \cup \left\{\pi_{e,j} \colon e \in E, j \in [2^{2b} + 2^{4b^2}]\right\}$ となる.
- 各 $e = \{u, v\} \in E$ と各 $r \in \{0,1\}^T$ に対し, 「$V^{\pi_u,\pi_v,\pi_e}(C_e; r) = 1$」という制約を $c'_{e,r}$ とし, $e \in E, r \in \{0,1\}^T$ 全てに対して $c'_{e,r}$ を集めたものを $\mathcal{C}'$ とする.

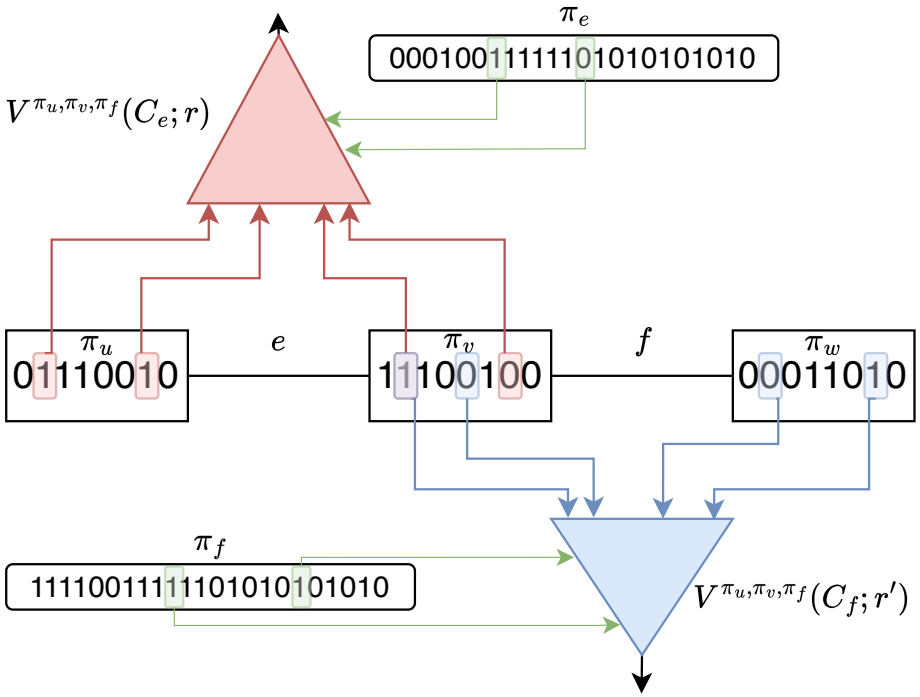


図 4.4　構成する q_0-CSP の図．PCPP 検証者がアクセスするオラクルを文字列とみなした時の各成分が変数に対応する．実際には，各 $r \in \{0,1\}^T$ に対して $V^{\pi_u,\pi_v,\pi_e}(C_e;r)$ を列挙してそれらを全て制約に追加する．

この検証者 $V^{\pi_u,\pi_v,\pi_e}(C_e;r)$ は，変数 π_u, π_v, π_e をオラクルとして実行する．

各制約「$V^{\pi_u,\pi_v,\pi_e}(C_e;r) = 1$」は，検証者が高々 q_0 回のオラクルアクセスを行うことから，高々 q_0 個の変数の値に依存する．すなわち φ は q_0-CSP である．

この q_0-CSP φ が所望の性質を満たすことを以下で確認する．

■φ のサイズ．　頂点 (変数) の個数は $2^b \cdot |V| + |E| \cdot \left(2^{2b} + 2^{4b^2}\right) \leq c \cdot \mathrm{size}(G)$ であり，制約の個数は $2^T \cdot |E| \leq 2^{\mathsf{poly}(b)} \cdot |E|$ である (検証者の時間計算量が b に関して多項式で抑えられることから $T \leq \mathsf{poly}(b)$ である)．

■充足割り当ての存在性．　G の制約全てを満たす制約 $a\colon V \to \Sigma$ が存在すると仮定する．このとき，各 $u \in V$ に対して

$$\pi_u = \mathrm{Had}_b(a(u))$$

とする．このとき，各辺 $e = \{u, v\} \in E$ に対して $(a(u), a(v))$ は辺 e における制約を充足する．従って，定理 2.3.1 より，ある $\pi_e \in \mathbb{F}_2^{2^{2b}+2^{4b^2}}$ が存在して，全ての $r \in \{0,1\}^T$ に対して $V^{\pi_u,\pi_v,\pi_e}(C_e;r)$ は受理する．このように構成された $(\pi_u)_{u\in V}$ と $(\pi_e)_{e\in E}$ から定まる φ の割り当て $a'\colon X \to \{0,1\}$ は φ の制約全てを充足する．

■不満足値の性質. $\mathrm{UNSAT}(\varphi) \geq 0.99 \cdot \mathrm{UNSAT}(G)$ を示す. φ の任意の割り当て $a' \colon X \to \{0,1\}$ を固定する. 各頂点 $u \in V$ に対し, a' が定める変数 π_u に対する部分割り当ての値を $\pi'_u \in \{0,1\}^b$ とする. 元の制約グラフ G の各頂点 $u \in V$ に対し, $a(u) \in \Sigma$ を「π'_u の復号化」, すなわち

- ある $\sigma \in \Sigma$ が存在して $\mathrm{dist}(\pi'_u, \mathrm{Had}_b(\sigma)) \leq 0.01$ ならば, $a(u) = \sigma$ (このような σ は存在するならば一意であることが演習問題 4 から保証されている)
- そうでないならば, $a(u) \in \Sigma$ は任意に選ぶ

で定める. 辺部分集合 $F \subseteq E$ を $\mathrm{UNSAT}(a; G)$ に寄与する辺の集合とする. すなわち, 辺 $e = \{u, v\} \in E$ であって, $(a(u), a(v))$ が辺 e における制約を充足しないようなものの集合とする.

ある $(\sigma_u, \sigma_v) \in c_e$ であって, 任意の $e = \{u, v\} \in E$ に対して $\mathrm{dist}(\pi'_u, \mathrm{Had}_b(\sigma_u)) \leq 0.01$ かつ $\mathrm{dist}(\pi'_v, \mathrm{Had}_b(\sigma_v)) \leq 0.01$ を満たすものが存在するならば, $a(u) = \sigma_u$ かつ $a(v) = \sigma_v$ であるため, $e \notin F$ となる. 対偶をとると, 任意の $e = \{u, v\} \in F$ と任意の $(\sigma_u, \sigma_v) \in c_e$ に対して, $\mathrm{dist}(\pi'_u, \mathrm{Had}_b(\sigma_u)) > 0.01$ または $\mathrm{dist}(\pi'_v, \mathrm{Had}_b(\sigma_v)) > 0.01$ が成り立つことがわかる.

従って, 検証者 $V(C_e)$ の性質 (定理 2.3.1) より, 任意の $e \in F$ に対して

$$\forall \pi'_e \in \{0,1\}^{2^{2b}+2^{4b^2}}, \quad \Pr_{r \sim \{0,1\}^T}\left[V^{\pi'_u, \pi'_v, \pi'_e}(C_e; r) \text{ が拒否する}\right] \geq 0.99$$

が成り立つ. すなわち, $e \in F$ に関する 2^T 個の制約のうち, $0.99 \cdot 2^T$ 個の制約は割り当て $\pi_u \leftarrow \pi'_u$, $\pi_v \leftarrow \pi'_v$, $\pi_e \leftarrow \pi'_e$ に対して充足されない. 従って,

$$\begin{aligned}
\mathrm{UNSAT}(a'; \varphi) &= \Pr_{e \sim E, r \sim \{0,1\}^T}\left[V^{\pi_u, \pi_v, \pi_e}(C_e; r) \text{ が拒否する}\right] \\
&= \Pr_{e \sim E, r \sim \{0,1\}^T}\left[V^{\pi_u, \pi_v, \pi_e}(C_e; r) \text{ が拒否する} \,\middle|\, e \in F\right] \cdot \Pr\left[e \in F\right] \\
&\geq 0.99 \cdot \frac{|F|}{|E|} \\
&= 0.99 \cdot \mathrm{UNSAT}(a; G)
\end{aligned}$$

が成り立つ. □

最後に補題 3.1.9 を用いて, q_0-CSP を制約グラフ $G' = \langle (V', E'), \Sigma_0, \mathcal{C}' \rangle$ に変換して証明が完了する:

定理 *4.3.1* の証明. 入力として受け取った制約グラフ $G = \langle (V, E), \Sigma, \mathcal{C} \rangle$ に対し, 補題 4.3.2 を用いて, アルファベット $\{0,1\}$ 上の q_0-CSP のインスタンス $\varphi = \langle (X, \{0,1\}, \mathcal{I}, \mathcal{C}') \rangle$ を構成する (これは多項式時間で行える). さらにこの q_0-CSP イ

ンスタンス φ を, 補題 3.1.9 を用いて制約グラフ $G' = \langle (V', E'), \{0,1\}^{q_0}, \mathcal{C}'' \rangle$ に変換する. アルゴリズム A はこのようにして得られる制約グラフ G' を出力する.

以下に G' が所望の性質を持つことを確認する: まず, 補題 4.3.2 より, ある定数 $c = c(|\Sigma|) > 0$ が存在して, φ は

- $|X| \leq c \cdot \mathrm{size}(G)$, $|\mathcal{I}| \leq c \cdot \mathrm{size}(G)$.
- $\mathrm{UNSAT}(G) = 0$ ならば $\mathrm{UNSAT}(\varphi) = 0$.
- $\mathrm{UNSAT}(G) > 0$ ならば $\mathrm{UNSAT}(\varphi) \geq 0.99 \cdot \mathrm{UNSAT}(G)$.

が成り立つ. このとき, 補題 3.1.9 より, G' はアルファベット $\{0,1\}^{q_0}$ 上の制約グラフであって, q_0 のみに依存するある定数 c' に対して

- $\mathrm{size}(G') \leq c' q_0 \cdot (|X| + |\mathcal{I}|)$.
- $\mathrm{UNSAT}(\varphi) = 0$ ならば $\mathrm{UNSAT}(G') = 0$.
- $\mathrm{UNSAT}(\varphi) > 0$ ならば $\mathrm{UNSAT}(G') \geq \frac{1}{q_0} \cdot \mathrm{UNSAT}(\varphi)$.

が成り立つ. これらを組み合わせると

- $\mathrm{size}(G') \leq c \cdot c' q_0 \cdot \mathrm{size}(G)$
- $\mathrm{UNSAT}(G) = 0 \Rightarrow \mathrm{UNSAT}(\varphi) = 0 \Rightarrow \mathrm{UNSAT}(G') = 0$.
- $\mathrm{UNSAT}(G') \geq \frac{1}{q_0} \cdot \mathrm{UNSAT}(\varphi) \geq \frac{0.99}{q_0} \cdot \mathrm{UNSAT}(G)$

が成り立つ. □

4.4　ギャップ増幅補題と PCP 定理の証明

ギャップ増幅補題 (補題 4.1.1) は, べき乗によって UNSAT を増幅させ, それに伴い増幅したアルファベットを定理 4.3.1 によって削減することによって得られる.

補題 *4.1.1* の証明. 入力として受け取った制約グラフ G に対し, 以下を適用する:

- 与えられた制約グラフ G を入力として補題 3.3.1 のアルゴリズムを実行し, その出力を G_1 とする (エクスパンダー化)
- 後で定めるパラメータ $\ell \in \mathbb{N}$ に対して $G_2 = G_1^{\ell}$ とする (べき乗操作).
- 定理 4.3.1 のアルゴリズムを G_2 を入力として実行し, その出力 G_3 を出力して終了する (アルファベット削減).

ユニバーサルな定数 d, λ を補題 3.3.1 の定数とする. また, べき乗操作のパラメータ ℓ は $d, \lambda, |\Sigma|$ のみに依存し, 十分大きくとる. 補題 3.3.1 と 4.2.2 と定理 4.3.1 より,

- 最終的な制約グラフ G_3 のサイズは，適当な $c = c(d, \lambda, |\Sigma|, \ell)$ に対して $\mathrm{size}(G_3) \leq c \cdot \mathrm{size}(G)$ である．
- $\mathrm{UNSAT}(G) = 0$ ならば $\mathrm{UNSAT}(G_3) = \mathrm{UNSAT}(G_2) = \mathrm{UNSAT}(G_1) = 0$ である．

が成り立つ．最後に，ℓ を適切に選ぶことによって，定数 $\alpha := \lceil 2/\ell \rceil$ に対して $\mathrm{UNSAT}(G_3) \geq \min\{\alpha, 2 \cdot \mathrm{UNSAT}(G)\}$ にできることを示す．まず，補題 3.3.1 より，制約グラフ G_2 は d-正則 λ-エクスパンダーであり，さらにある定数 β_1 に対して $\mathrm{UNSAT}(G_1) \geq \beta_1 \cdot \mathrm{UNSAT}(G)$ が成り立つ．従って，補題 4.2.2 より，ある定数 $\beta_2 = \beta_2(|\Sigma|) > 0$ に対して

$$\begin{aligned}\mathrm{UNSAT}(G_2) &\geq \beta_2\sqrt{\ell} \cdot \min\left\{\mathrm{UNSAT}(G_1), \frac{1}{\ell}\right\} \\ &\geq \beta_2\sqrt{\ell} \cdot \min\left\{\beta_1 \cdot \mathrm{UNSAT}(G), \frac{1}{\ell}\right\} \\ &\geq \beta_1\beta_2\sqrt{\ell} \cdot \min\left\{\mathrm{UNSAT}(G), \frac{1}{\ell}\right\}\end{aligned}$$

が成り立つ．最後に定理 4.3.1 より，ある定数 $\gamma = \gamma(|\Sigma|) > 0$ に対して

$$\begin{aligned}\mathrm{UNSAT}(G_3) &\geq \gamma \cdot \mathrm{UNSAT}(G_2) \\ &\geq \gamma\beta_1\beta_2\sqrt{\ell} \cdot \min\left\{\mathrm{UNSAT}(G), \frac{1}{\ell}\right\}\end{aligned}$$

が成り立つ．このとき，$\ell = \left\lceil \left(\frac{2}{\gamma\beta_1\beta_2}\right)^2 \right\rceil$，$\alpha = \frac{1}{\ell}$ とすれば，$\mathrm{UNSAT}(G_3) \geq \min\{\alpha, 2 \cdot \mathrm{UNSAT}(G)\}$ が成り立つ．

□

PCP 定理を証明する全ての準備が整った．

定理 *1.2.7* の証明．定理 1.2.14 を示せば，これと 3-QuadEq の NP 完全性を組み合わせると定理 1.2.7 が従う (演習問題 2)．また，補題 3.1.6 より，定理 1.2.14 と 3.1.4 は同値であるため，以降では定理 3.1.4 の証明を与える．すなわち，与えられた 3-QuadEq のインスタンス $\varphi_0 = (f_k)$ に対して，多項式時間で定理 3.1.4 の条件を満たす q-CSP のインスタンス φ を構成する．これは以下のアルゴリズムによって構成される．

アルゴリズム 4.4.1

アルゴリズム A を次のように定める．ただし入力として，3-QuadEq のインスタンス $\varphi_0 = (f_k)$ を受け取る．

1. φ_0 を 3-CSP とみなし, 補題 3.1.9 を用いて制約グラフ G_0 に変換する.
2. 各 $i = 1, \ldots, \lceil \log_2(3s) \rceil$ に対し, 制約グラフ G_i を入力として補題 4.1.1 のアルゴリズムを実行してその出力として得られるグラフを G_{i+1} とする.
3. $G_{\lceil \log_2(3s) \rceil}$ を出力して終了する.

元の問題 3-QuadEq はアルファベット $\mathbb{F}_2$ 上の 3-CSP であるため, G_0 のアルファベットは $\mathbb{F}_2^3$ である. また, 最終的に得られる制約グラフのアルファベットは, 定理 4.3.1 のアルファベット Σ_0 となる. これらのアルファベットをユニバーサルなものとして固定する. このとき, 適当な定数 $c > 0$ が存在して各 $i = 1, \ldots, \lceil \log_2(3s) \rceil$ に対して $\mathrm{size}(G_i) \leq c \cdot \mathrm{size}(G_{i-1})$ が成り立つため, 最終的な制約グラフのサイズは

$$\mathrm{size}(G_{\lceil \log_2(3s) \rceil}) \leq c^{O(\log s)} \cdot \mathrm{size}(G_0) = s^{O(1)}$$

が成り立つ. また, 補題 3.1.9 と 4.1.1 より, $\mathrm{UNSAT}(\varphi_0) = 0$ ならば, 全ての i について $\mathrm{UNSAT}(G_i) = 0$ が成り立つ. 最後に, $\mathrm{UNSAT}(\varphi_0) > 0$ ならば $\mathrm{UNSAT}(\varphi_0) \geq \frac{1}{s}$ であり, 補題 3.1.9 より $\mathrm{UNSAT}(G_0) \geq \frac{1}{3s}$ となり, さらに全ての i に対して $\mathrm{UNSAT}(G_i) \geq \min\{2\mathrm{UNSAT}(G_{i-1}), \alpha\}$ が成り立つため,

$$\mathrm{UNSAT}\left(G_{\lceil \log_2(3s) \rceil}\right) \geq \min\left\{\alpha, 2^{\log_2(3s)} \cdot \mathrm{UNSAT}(G_0)\right\} \geq \alpha$$

が成り立つ. □

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. 13).

[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).

[Bab85] L. Babai. “Trading group theory for randomness”. In: **Proceedings of the seventeenth annual ACM symposium on Theory of computing - STOC ’85**. the seventeenth annual ACM symposium (Providence, Rhode Island, United States). New York, New York, USA: ACM Press, 1985, pp. 421–429. DOI: 10.1145/22145.22192 (cit. on p. 22).

[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. 29).

[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. 16).

[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).

[GMR85] S. Goldwasser, S. Micali, and C. Rackoff. “The knowledge complexity of interactive proof-systems”. In: **Proceedings of the seventeenth annual ACM symposium on Theory of computing - STOC ’85**. the seventeenth annual ACM symposium (Providence, Rhode Island, United States). New York, New York, USA: ACM Press, 1985, pp. 291–304. DOI: 10.1145/22145.22178 (cit. on p. 22).

[GMW86] O. Goldreich, S. Micali, and A. Wigderson. “Proofs that yield nothing but their validity and a methodology of cryptographic protocol design”. en. In: **27th Annual Symposium on Foundations of Computer Science (sfcs 1986)**. 27th Annual Symposium on Foundations of Computer Science (sfcs 1986) (Toronto, ON, Canada). IEEE, Oct. 1986, pp. 174–187. DOI: 10.1109/sfcs.1986.47 (cit. on p. 22).

[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 pp. 16, 17, 21).

[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. 17).

[LFKN02] C. Lund, L. Fortnow, H. Karloff, and N. Nisan. “Algebraic methods for interactive proof systems”. In: **Proceedings [1990] 31st Annual Symposium on Foundations of Computer Science**. [1990] 31st Annual Symposium on Foundations of Computer Science (St. Louis, MO, USA). IEEE Comput. Soc. Press, 2002, 2–10 vol.1. DOI: 10.1109/fscs.1990.89518 (cit. on p. 23).

[ODo] R. O’Donnell. **A history of the PCP Theorem**. URL: https://courses.cs.washington.edu/courses/cse533/05au/pcp-history.pdf (visited on 05/28/2025) (cit. on pp. 21, 22).

[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. 59).

[Sha02] A. Shamir. "IP=PSPACE (interactive proof=polynomial space)". In: **Proceedings [1990] 31st Annual Symposium on Foundations of Computer Science**. [1990] 31st Annual Symposium on Foundations of Computer Science (St. Louis, MO, USA). IEEE Comput. Soc. Press, 2002, 11–15 vol.1. DOI: 10.1109/fscs.1990.89519 (cit. on p. 23).

付録 A

付録

A.1 基本的な確率の不等式

この節では, 証明の中で用いられる確率の不等式とその証明を述べる.

補題 A.1.1 （Markov の不等式）

任意の非負の確率変数 X と任意の $t > 0$ に対し,

$$\Pr[X \geq t] \leq \frac{\mathbb{E}[X]}{t}$$

が成り立つ.

証明. ここでは簡単のため, X の台 $\Omega = \mathsf{supp}(X) \subseteq \mathbb{R}$ は有限集合, すなわち $|\Omega| < \infty$ と仮定する[*1]. このとき,

$$\begin{aligned}
\mathbb{E}[X] &= \sum_{x \in \Omega} x \Pr[X = x] \\
&\geq \sum_{x \in \Omega, x \geq t} x \Pr[X = x] \\
&\geq \sum_{x \in \Omega, x \geq t} t \Pr[X = x] \\
&= t \Pr[X \geq t]
\end{aligned}$$

を整理すると主張を得る. □

[*1] 確率変数 X のとりうる値の集合, すなわち $\{x \colon \Pr[X = x] > 0\}$ を X の台と呼び, $\mathsf{supp}(X)$ と表す.

補題 A.1.2 （Chebyshev の不等式）

期待値 $\mathbb{E}[X]$ と分散 $\mathbf{Var}[X]$ が存在する任意の確率変数 X と任意の $t > 0$ に対し,

$$\Pr[|X - \mathbb{E}[X]| \geq t] \leq \frac{\mathbf{Var}[X]}{t^2}$$

が成り立つ.

証明. 確率変数 $Y = (X - \mathbb{E}[X])^2$ は非負の確率変数であるから, これに Markov の不等式を適用すると

$$\begin{aligned}\Pr\left[|X - \mathbb{E}[X]| \geq t\right] &= \Pr\left[(X - \mathbb{E}[X])^2 \geq t^2\right] \\ &\leq \frac{\mathbb{E}\left[(X - \mathbb{E}[X])^2\right]}{t^2} \\ &= \frac{\mathbf{Var}[X]}{t^2}\end{aligned}$$

を得る. □

補題 A.1.3 （Paley-Zygmund の不等式）

非負整数値をとる任意の確率変数 X に対し,

$$\Pr[X \geq 1] \geq \frac{\mathbb{E}[X]^2}{\mathbb{E}[X^2]}.$$

証明. 非負整数値をとる確率変数 X に対し

$$\begin{aligned}\mathbb{E}[X] &= \mathbb{E}[X \cdot \mathbf{1}_{X \geq 1}] \\ &\leq \sqrt{\mathbb{E}[X^2] \cdot \mathbb{E}[\mathbf{1}_{X \geq 1}]} && \because \text{Cauchy-Schwarz の不等式} \\ &= \sqrt{\mathbb{E}[X^2] \cdot \Pr[X \geq 1]}.\end{aligned}$$

両辺を二乗して整理すると主張を得る. □