線型ホモトピー型理論を動機付けする
トポロジカル量子コンパイルの
形式検証に向けて

322301073 伊藤 賢世
アドバイザー：Jacques Garrigue

2025 年 1 月 30 日

修論内容：サーベイ

Hisham Sati らの研究プロジェクト

Quantum Certification via Linear Homotopy Types

を理解するためのトポロジカル量子コンパイルについて

研究概要

主張 信頼性のある量子コンピュータを実現するための検証言語「QS」とその基礎理論「線型ホモトピー型理論（LHoTT)」の提唱.

研究者 Hisham Sati，Urs Schreiber，David J. Meyer，. . .

拠点 ニューヨーク大学アブダビ校
Center for Quantum and Topological Systems (CQTS)

概要

前半 修士論文では，量子コンピュータの誤りを排除するトポロジカル量子コンパイルを主題として捉え，（古典）コンピュータ上で検証するための言語として LHoTT（あるいは QS）を動機付ける．

後半 本発表では，特にトポロジカル量子コンパイルの一般論である量子コンパイルについて説明する．

目次

イントロダクション

重要課題：量子コンピュータの実現

- 量子コンピュータ $=$ 量子力学の原理を活用した計算機.
- 普及している（古典）コンピュータの性能を凌駕する量子コンピュータ上のアルゴリズムがいくつか見つかっている.

$\longrightarrow$ 量子コンピュータの実現に対する期待は計り知れない.

イントロダクション

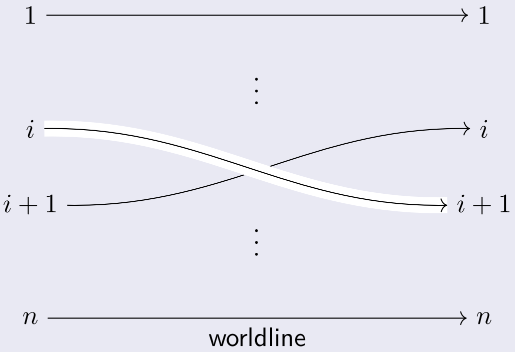
計算の信頼性における課題①：Decoherence

- 量子コンピュータのデータ $=$ 量子系の状態（量子ビット）.
  - n 量子ビット $(\mathbb{C}^2)^{\otimes n} := \underbrace{\mathbb{C}^2 \otimes \cdots \otimes \mathbb{C}^2}_{n\text{個}}$
- 量子状態は環境の影響で壊れやすい（Decoherence 問題）.

$\longrightarrow$ エラーが多く，アルゴリズム通りに計算することは難しい.

アプローチ：エラーの低減

- トポロジカル量子計算
  - トポロジーで量子状態を構成．
  - 組み紐で計算を実行．



- Decoherence に影響されにくく，エラーが減る．

イントロダクション

コンパイル

- あるプログラミング言語で書かれたコードを，数学的な意味を保ちつつ，別のコードに変換することをコンパイルという．
- そのプログラムをコンパイラという．
- 特に，コンピュータが実行可能なコードに変換したりする．
- また，コードの実行に関する最適化が施される．

例

C コンパイラ C 言語 → アセンブリ言語

アセンブラ アセンブリ言語 → 機械語

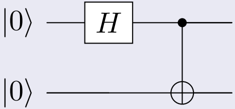
古典コンピュータは最終的に機械語を実行している．

イントロダクション

量子回路

- 計算は量子ゲートによって実行される．
  - 量子系のユニタリ変換 $U\colon (\mathbb{C}^2)^{\otimes n} \longrightarrow (\mathbb{C}^2)^{\otimes n}$.
- 量子回路は量子ゲートで構成され，アルゴリズムを記述．

$$\mathrm{CNOT}(H \otimes I)\left|00\right\rangle,$$



量子コンパイル：計算したい量子回路 $\rightarrow$ 実行可能な量子回路

- 任意の量子ゲートを準備し実行可能にするのは現実的でない．
- 有限個のみ準備（基本ゲート）し，その量子回路で実行する．

イントロダクション

量子コンパイルの特徴

基本ゲートの組み合わせで，任意の量子アルゴリズムを正確にシミュレートすることはできない．

$\longrightarrow$ 任意の精度で近似的にシュミレートする．
(cf. Solovay・Kitaev アルゴリズム)

注意

後半では，Solovay・Kitaev アルゴリズムを説明する．

課題①の先：トポロジカル量子コンパイル

トポロジカル量子コンピュータの量子コンパイル．

- 量子ゲート = 組み紐ゲート

イントロダクション

コンパイルの形式検証

古典コンピュータでは，コンパイルが正しいことの検証が研究されている．

- コンパイラの仕様を検証言語で記述し，コードの数学的な意味が保たれることを証明する．
- cf. C コンパイラの 1 つ CompCert の正しさは，Coq で検証．

計算の信頼性における課題②：コンパイルの正しさ

トポロジカル量子コンパイルの形式検証

- 量子回路の記述
- トポロジカル量子ゲートの記述

イントロダクション

Satiらの提案：線型ホモトピー型理論（LHoTT）

- LHoTT = Linear + HoTT.
- 量子回路の記述 →QS ⊆ LHoTT
- トポロジカル量子ゲートの記述 → HoTT ⊆ LHoTT

⟶ LHoTT は信頼性のある量子プログラミングに対する最初の包括的なパラダイムになる.

目次

量子コンパイル

近似するには，距離の概念を要する．以下，$d = 2^n$ とする．

定義 (ユニタリ変換の間の距離)

任意の $U, V \in U(d)$ に対し，その距離を次で定義する．

$$d(U,V) := ||U - V|| = \sup_{||\psi||=1} ||(U - V) \left|\psi\right\rangle ||$$

量子コンパイル

定義 (Universal set)

ゲートの集合 $\mathcal{G}$ が $U(d)$ に対して万能 *(universal)* であるとは，任意の $U \in U(d)$ と $\varepsilon > 0$ に対し，ある $l \in \mathbb{N}$ と $G_1, \ldots, G_l \in \mathcal{G}$ が存在して，$d(U, G_l \cdots G_1) < \varepsilon$ となることである．有限列 $G_1, \ldots, G_l$ を U の ε 近似列といい，l を長さという．

量子コンパイル

- $U(d)$ の元は $U(2) \cup \{\, \mathrm{CNOT} \,\}$ の元の合成で表せる．
- $U(2)$ の元は $SU(2)$ の元の合成で表せる．ただし，物理的な意味を持たない大域的な相を無視する．

$\longrightarrow$ よって，$SU(2)$ の元が近似できればよい．

定義 (Instruction set)

ゲートの有限集合 $\mathcal{G}$ が *instruction set* であるとは，次の 3 つの条件を満たすことである．

1. $\mathcal{G} \subseteq SU(2)$．
2. $\mathcal{G}$ は逆行列で閉じている．
3. $\mathcal{G}$ は $SU(2)$ に対して *universal* である．

量子コンパイル

定理 (Solovay・Kitaev アルゴリズム)

$\mathcal{G} \subseteq SU(2)$ を *instruction set* とする．ある正数列 (ε_n) であって，

$$\varepsilon_n < \varepsilon_{n-1}, \quad \varepsilon_n \to 0$$

となるものが存在し，次を満たす．任意のゲート $U \in SU(2)$ と深さ n を入力して，ε_n 近似列 $G_1, \ldots, G_{l_n} \in \mathcal{G}$ 並びに量子回路 $G_{l_n} \cdots G_1$ を出力する *SK* アルゴリズム $\mathrm{SK}(U, n)$ が帰納的に定義できる．（続く）

量子コンパイル

定理 (Solovay・Kitaev アルゴリズム)

…このとき，任意の ε に対し n を適当に選ぶことで近似列が得られ，近似列の長さ l_n と *SK* アルゴリズムの実行時間 t_n が次のように評価できる．

$$n = \left\lceil \frac{\ln[\frac{\ln(1/\varepsilon c^2)}{\ln(1/\varepsilon_0 c^2)}]}{\ln(3/2)} \right\rceil,$$

$$l_n = O(\ln^{\ln 5/ln(3/2)}(1/\varepsilon)),$$

$$t_n = O(\ln^{\ln 3/ln(3/2)}(1/\varepsilon)).$$

量子コンパイル：SK アルゴリズムの証明

（証明）次の Step0〜Step2 で SK アルゴリズムを帰納的に定義する．その前に Step2 で登場する定数 c に対し，$\varepsilon_0 < 1/c^2$ なる定数をとる．

Step0: 初期近似列の brute-force search

ε_0 に対して，ある長さ l_0 が存在して，次を満たす．$U \in SU(2)$ に対し，ε_0 近似列 $L_0 \in \coprod_{0<l\leq l_0} \mathcal{G}^l$ がとれる．その量子回路を U_0 とおく．SK$(U, 0) := (L_0; U_0)$ と定義する．

量子コンパイル：SK アルゴリズムの証明

Step1: 群交換子による分割

$n \geq 1$ に対し，$(L_{n-1}; U_{n-1}) := \mathrm{SK}(U, n-1)$ とおく．ある分割

$$UU_{n-1}^{\dagger} = \Delta = VWV^{\dagger}W^{\dagger}, \quad d(I,V), d(I,W) < c_0\sqrt{\varepsilon_{n-1}}$$

によって，$V, W \in \mathcal{G}$ を得る．ただし，$c_0 \approx \frac{1}{\sqrt{2}}$ で定数．

量子コンパイル：SK アルゴリズムの証明

Step2: 交換因子に SK アルゴリズム

V, W に対して SK アルゴリズムを適用する．

$$(L_V; V_{n-1}) := \mathsf{SK}(V, n-1), \quad (L_W; W_{n-1}) := \mathsf{SK}(W, n-1).$$

このとき，$\Delta_{n-1} := V_{n-1}W_{n-1}V_{n-1}^{\dagger}W_{n-1}^{\dagger}$ とおくと，

$$d(\Delta, \Delta_{n-1}) < \varepsilon_n := c\varepsilon_{n-1}^{\frac{3}{2}}, \tag{1}$$

$$\varepsilon_{n-1} < \frac{1}{c^2} \tag{2}$$

を満たす．ただし，$c \approx 8c_0 \approx 4\sqrt{2}$ で定数．

$$U_n := \Delta_{n-1}U_{n-1}, \quad \mathsf{SK}(U, n) := (L_{n-1}L_W^{\dagger}L_V^{\dagger}L_W L_V; U_n)$$

量子コンパイル：SK アルゴリズムの証明

この手続きにより，$U_{n-1}U^\dagger = \Delta_{n-1}U_{n-1}U^\dagger = \Delta_{n-1}\Delta^\dagger$ となるから，

$$d(U, U_{n-1}) = d(I, U_{n-1}U^\dagger) = d(I, \Delta_{n-1}\Delta^\dagger) = d(\Delta, \Delta_{n-1}) < \varepsilon_n$$

と評価できる．また，不等式（2）により $\varepsilon_n < \varepsilon_{n-1}$ を満たす．ε_0 は，不等式（2）を満たすようにとっていたので，帰納的定義がうまくいっている．
定理の後半は，手続きの定め方から

$$\begin{aligned} l_n &= 5l_{n-1}, \\ t_n &\leq 3t_{n-1} + \mathrm{const} \end{aligned}$$

となり，示せる．（証明終わり）

目次

まとめ

振り返り

- 修士論文では，量子コンピュータの誤りを排除するトポロジカル量子コンパイルを主題として捉え，（古典）コンピュータ上で検証するための言語として LHoTT（あるいは QS）を動機付けることがねらいであった．
- 本発表では，特にトポロジカル量子コンパイルの一般論である量子コンパイルについて，具体的に説明した．