

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

2025 年 1 月 30 日

イントロダクション

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 信頼性のある量子計算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)

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.
- いくつかの計算について、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見つかったている.
- Microsoft や Google など巨額の投資.

→ 量子コンピュータの実現に対する期待は計り知れない.

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）.
- 量子状態は外部環境の影響で壊れやすい（Decoherence 問題）.

→ エラーが多く、アルゴリズム通りに計算を実行させることは難しい.

イントロダクション

アプローチ①：エラーの訂正

- 古典コンピュータでも物理レベルではエラーが起きている。
 - データに冗長性をもたせて、データを復元（誤り訂正理論）。
- 量子ビットを増やして、量子コンピュータにも応用（量子誤り訂正）。
→ 量子ビットを大規模に用意する問題に至る。

アプローチ②：エラーの低減

- **トポロジカル量子計算** = トポロジカルな情報で量子状態を構成。
- Decoherence に影響されにくく、エラーが減る。
→ 訂正のための冗長な量子ビットは少なくて済む。

①と②を組み合わせて、decoherence に対処。

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別の言語のコードに変換すること。特に、実行可能なコードに変換すること。実行前に最適化が施されている。

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。

量子コンパイル

ソフトウェア側 量子アルゴリズムを量子回路で記述する。

ハードウェア側 実行可能な量子回路。

量子コンパイルのアルゴリズム

- 任意の実行可能な量子回路を用意するのは現実的でない.
- 量子回路を構成する基本要素を準備して、固定する (有限個の**基本ゲート**).
- 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.

→ 任意の精度で**近似的に**シミュレートする.
(cf. Solovay • Kitaev アルゴリズム)

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.

計算の信頼性における課題②：コンパイルの正しさ

古典計算では、コンパイルの正しさが検証されている。

- コンパイルの仕様を検証言語で記述
- cf. C コンパイラは検証言語「Coq」で検証されている。

→ トポロジカル量子コンパイルの近似精度も検証したい。

Sati らの提案：線型ホモトピー型理論 (LHoTT)

- 前提

→ 結論

イントロダクション

タイトル

- 前提

→ 結論

一枚目のスライド

普通に文中で \LaTeX コマンドが使用できます. (そらそう)

ブロックのタイトル

`block` 環境を作成し `begin` の第二引数にタイトルを渡すようになります.

アラート

`alertblock` を指定するとこんな風になります.

例

`exampleblock` を指定するとこうなります.

スライドで左右比較するときなどにこのようにします

内容 1

内容 1 内容 1 内容 1 内容 1 内
容 1 内容 1 内容 1 内容 1 内容
1 内容 1 内容 1 内容 1 内容 1
内容 1

内容 2

内容 2 内容 2 内容 2 内容 2 内
容 2 内容 2 内容 2 内容 2 内容
2 内容 2 内容 2 内容 2 内容 2
内容 2

このとき各 column の幅は合計 100%にしてしまうとデザインの的にあんまよくないです

- 一個目

- 一個目
- 二個目

- 一個目
- 二個目
- 三個目

- 一個目
- 二個目
- 三個目
- 四個目

- 一個目（一画面目以降表示）

- 四個目（二画面目まで表示）

Item 以外の Overlay

常に表示 1

常に表示 2

常に表示 3

常に表示 1

二画面目以降のみ表示 (uncover)

常に表示 2

二画面目以降のみ表示 (only)

常に表示 3