

格子手術プログラムのためのコン-positionalな量子ビット割り当て手法

脇坂 遼（京都大学） 鈴木 泰成（日本電信電話 コンピュータ&データサイエンス研究所）

概要 大規模な格子手術プログラムに対するスケーラブルな量子コンパイラ

大規模格子手術プログラムをモジュール化されたまま扱えるコン-positionalな量子ビット割り当て手法を提案

```
// 外部ライブラリ
import exlib.f2

fn f1(x1,x2){...}
q0 = init();
q1 = init();
q2 = init();
cx(q0,q1);
f1(q0,q1);
f2(q0,q1);
f1(q2,q0);
```

入力量子プログラム
(実機制約などは未考慮)

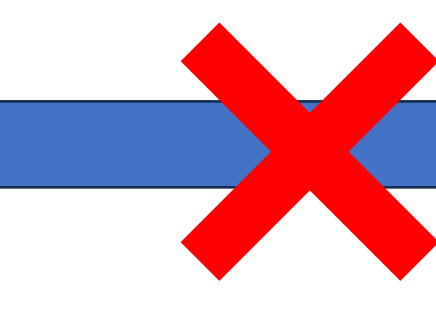
インライン展開
ループ展開

```
q0 = init();
q1 = init();
cx(q0,q1);
...body_exp_f1[q0,q1]...;
...body_exp_f2[q0,q1]...;
...body_exp_f1[q2,q1]...;
```

既存手法

プログラムが極めて巨大に → スケーラビリティに課題

量子ビット割り当て



スケールしない

```
...
q0 = init(loc0);
q1 = init(loc1);
q2 = init(loc2);
tmp = init(loc3);
v = meas[XX](q0,tmp);
...
relocate(q0,q1);
f1(q0,q1);
...
relocate(q2,q0);
f1(q2,q0);
```

格子手術プログラム
(実機で動作)

関数呼出し時の制約
(呼び出し規約)を
マルチエージェント
経路計画問題を
解くことで解決

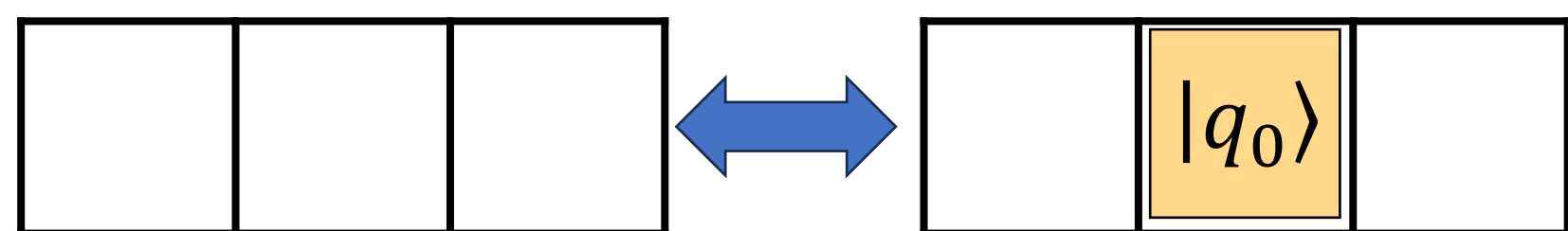
→ 大規模プログラム
に対する現実的な
時間でのコンパイルを実現

本研究：コン-positionalな量子ビット割り当て

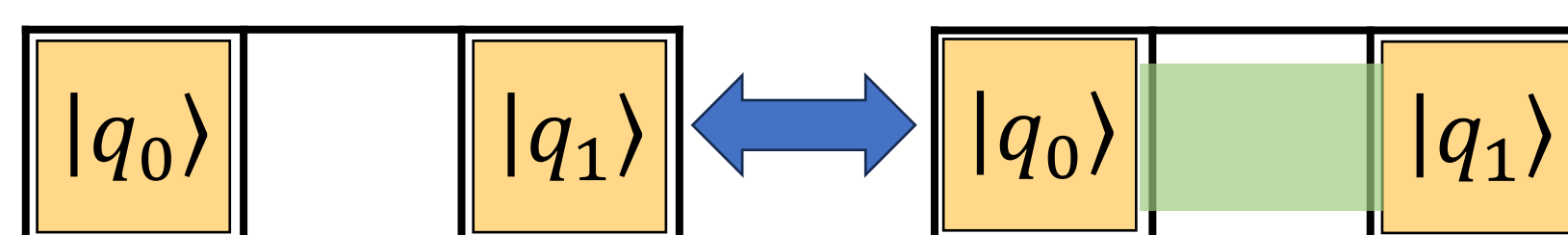
予備知識 格子手術による誤り耐性量子計算

- 誤り耐性量子計算とは、量子誤り訂正によって大規模な量子計算を実現するための技術
- 格子手術とは、誤り耐性のある論理量子ビットに対する論理操作を実現する計算手法の1つ [Horsman+ '12]

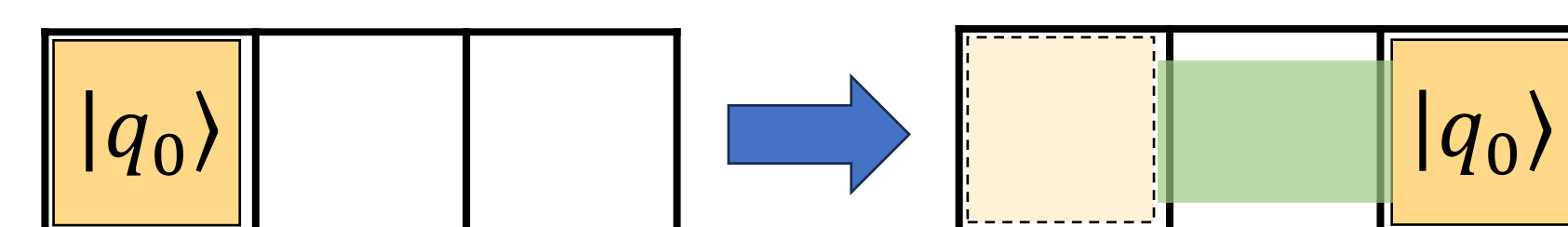
格子手術の各操作は、アーキテクチャグラフ上のグラフ的操作として表現される



(1) 量子ビットの割り当て・解放



(2) 2量子ビット操作（多体測定）



(3) 量子ビットの移動

操作制約 (1) 割り当て先は空でなければならない

(2)(3) パス上に他の論理ビットや一時パスがあってはならない

格子手術プログラムの量子ビット割り当て = 論理ビットの頂点への割り当て + パススケジューリング

アーキテクチャ
グラフ

|q₀> 論理量子ビット

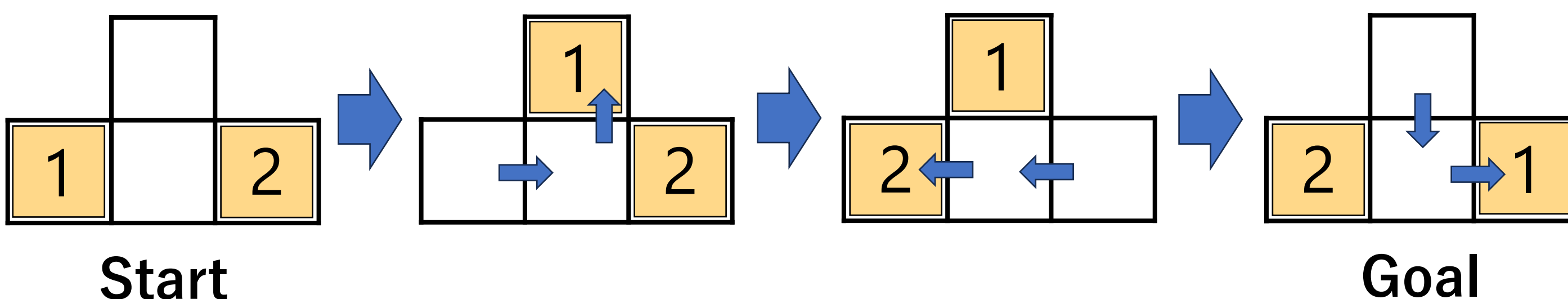
一時的な
パス接続

本研究の貢献 マルチエージェント経路探索によるコン-positionalな量子ビット割り当て

■ マルチエージェント経路探索 (MAPF)

Input: グラフ G , 始点と終点のペア $(s_1, t_1), \dots, (s_n, t_n)$

Output: 互いに衝突しない経路集合 $P_1, \dots, P_n$

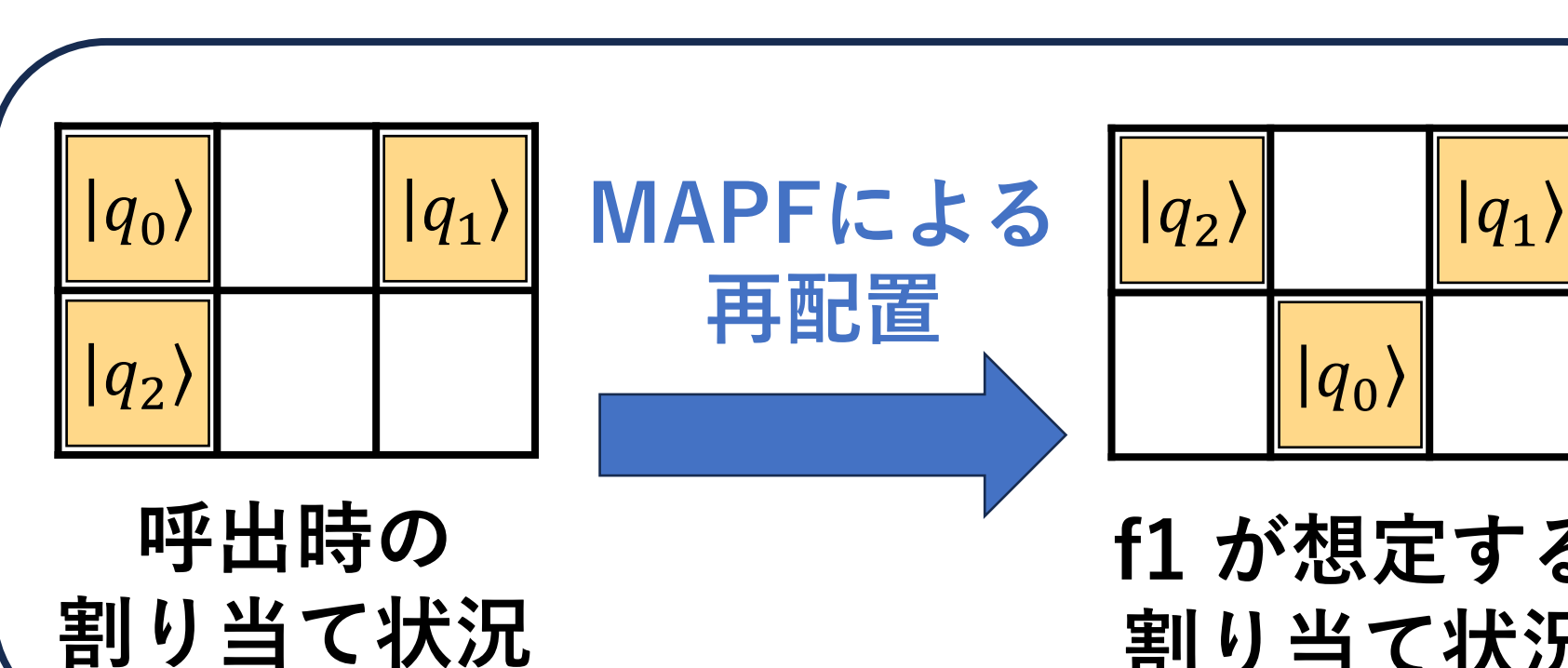


■ MAPF に基づく量子ビット割り当て

```
fn f1(x1,x2,x3){...}
...
q0 = init(loc0);
q1 = init(loc1);
q2 = init(loc2);
...
relocate1(q0,q1,q2);
f1(q0,q1,q2);
relocate2(q0,q1,q2);
f1(q2,q0,q1);
```

基本的に前から順に割り当てていく

関数呼出し時に MAPF で制約を解決



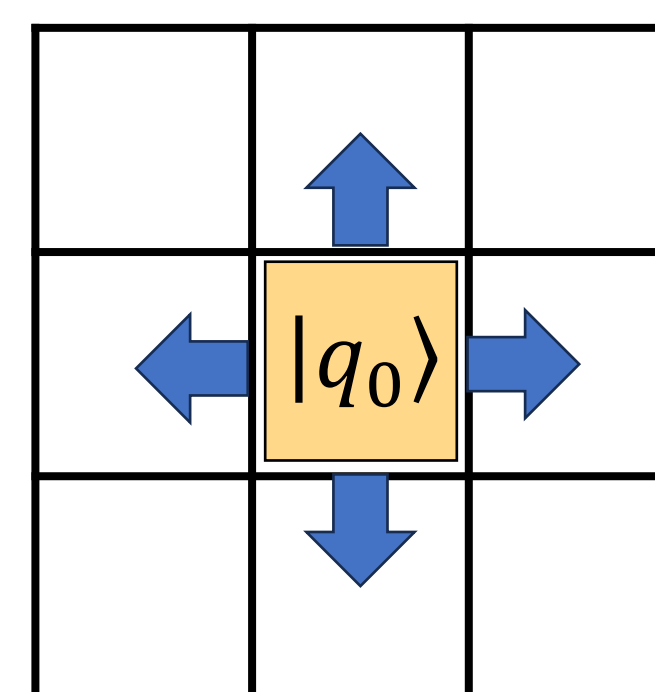
利点1. インライン展開が不要 → モジュラーな最適化が可能

利点2. データ量子ビットのレイアウトに依存しない

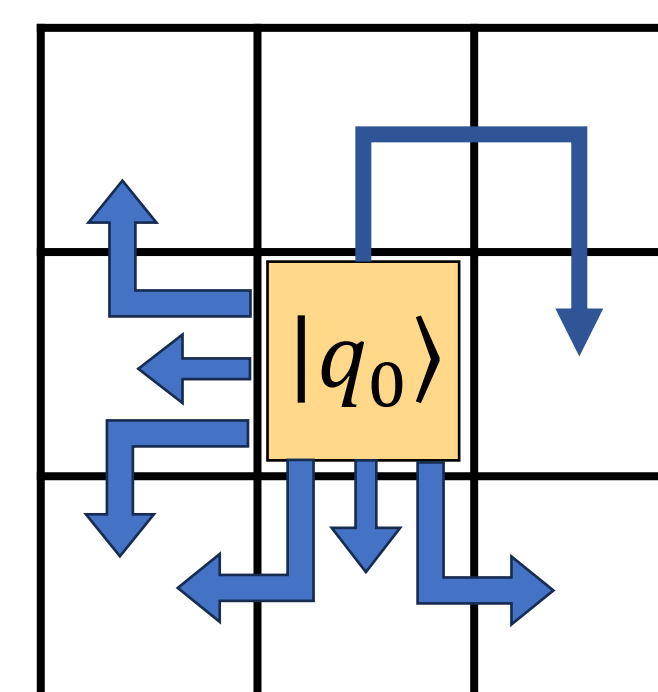
→ 最適化済みの外部ライブラリを簡単に再利用可能

■ 現状の課題：パス候補の組み合わせ爆発

格子手術では1ステップに任意の距離の移動ができる
→ 遷移の候補がチップ面積に比例し探索空間が巨大に



隣接マスへ移動 or 待機
(よくある MAPF)



任意の場所に移動可
(格子手術)

同じ場所に複数の
方法で到達可能

最短経路でなくても
問題ない

- 実際に 5×5 のグリッドで量子ビット数が7個程度でも現実的な時間で解けないインスタンスが存在
- 最適でなくてよいので探索が早く終わるものが欲しい (ただし完全性は保証したい)
- 最終的には 100 論理ビット程度でも動くものを目指す

■ 今後の予定

- 再配置問題が NP-hard であることを証明する
- LaCAM [Okumura '23] などの高速な MAPF 実装を試す
- 量子テレポーテーションの利用を考える
- 部分的なインライン展開の有効化