

量子コンピューティング向けテストフレームワークの開発 - オープンソースライブラリQuantestPyのリリース -

松本光洋 中村純也

インストール方法

```
pip install quantestpy
```

オープンソースで公開済

<https://github.com/QuantestPy/quantestpy>

テストフレームワークが以下のテストの苦労を軽減

苦労① テスト用に余分にコードを書く必要がある

古典コンピュータを含む一般論

苦労② 規約を設けずに複数人でテストをすると書き方がバラバラになり後で確認が困難なテストが散在する

背景

量子コンピュータでは古典コンピュータのテストフレームワークをそのまま利用することができない

目的

課題①

抽象化レイヤが異なる

古典: 高級言語

量子: アセンブリ言語に近い

課題②

複雑な大規模量子回路の出力を事前に知ることは難しい

テストフレームワークの機能

準備が難しい

入力

対応が想定通りかをチェックする

出力

対象はシミュレータ

実機での準備は容易でない

観測値
(実機では確率論的)

観測値は出力として対応しくない

出力

上記の課題をクリアすることを念頭に置いて、量子コンピュータに特化したテストのフレームワークを開発する

量子コンピュータを活用したソフトウェア開発速度の向上

✓ 開発者は開発における本質的な部分に注力(テストが簡単になる)

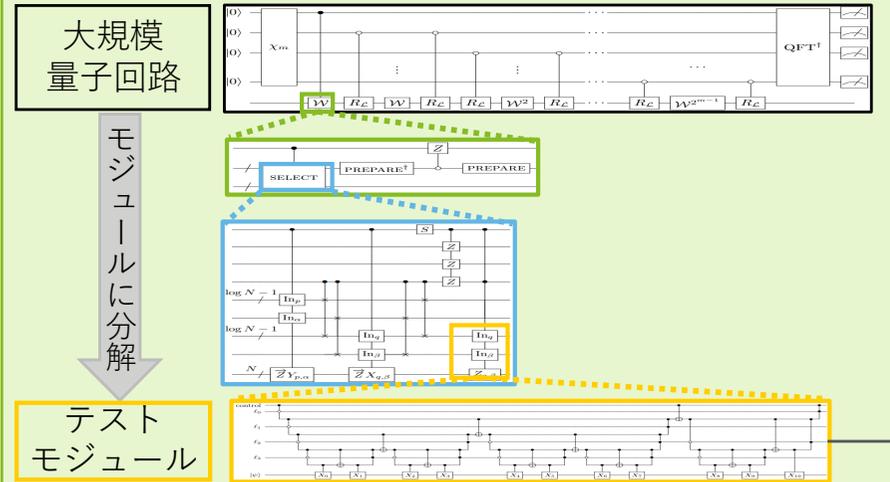
✓ 量子コンピュータのリソースを社会全体として有効活用(誤ったコードを実行する可能性が減る)

意義

QuantestPyはテストの指針、ライブラリ、ドキュメントを包括的に示す初めての量子コンピューティング向けテストフレームワークである

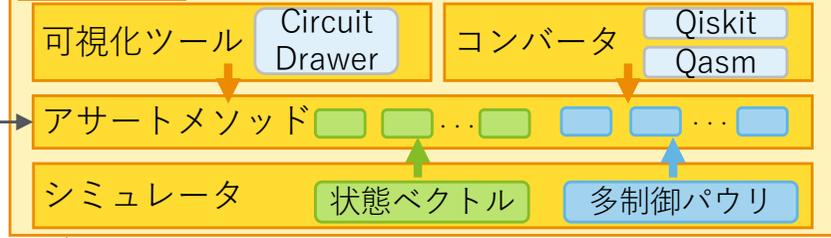
テストの指針

大規模回路を入出力の対応関係が分かるモジュールに分解



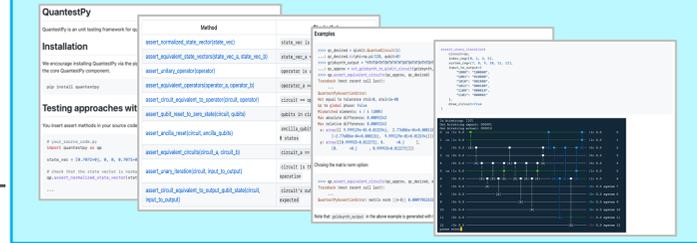
※ 回路図の引用元：Babbush et al., arXiv:1805.03662

ライブラリ



ドキュメント

GitHub上に使用方法・説明・使用例を充実



サポート

アサートメソッド呼出

1. 量子回路作成 (テストモジュール特定)

```
import quantestpy as qp
qc = qp.QuantestPyCircuit(16)
qc.add_gate({"name": "X", "target_qubit": [7], "control_value": [1]})
qc.add_gate({"name": "X", "target_qubit": [2], "control_value": [1, 0, 1]})
...
```

QuantestPy
独自シミュレータで回路を作成した例

QiskitやOpenQASMで作成された回路もアサートメソッドに適用可能(コンバータがQuantestPyCircuitに変換)

2. アサートメソッド適用

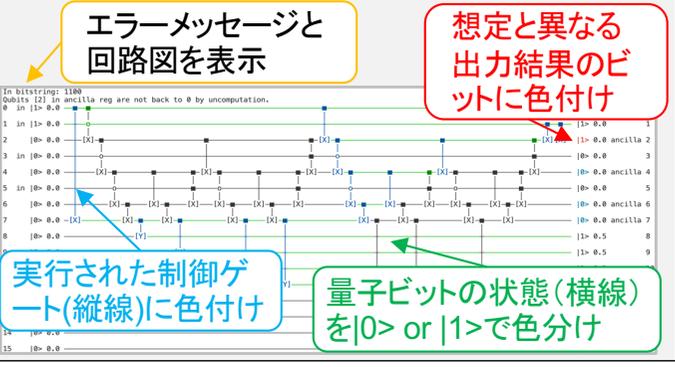
```
qp.assert_unary_iteration(
    circuit=qc,
    index_reg=[0, 1, 3, 5],
    system_reg=[8, 9, 10, 11],
    ancilla_reg=[2, 4, 6, 7],
    input_to_output={
        "1000": "00000000",
        "1001": "10000000",
        "1010": "11000000",
        "1011": "11100000",
        "1100": "11110000",
        "1101": "11111000",
        "1110": "11111100",
        "1111": "11111100"
    },
    draw_circuit=True
)
```

QuantestPyのアサートメソッド

回路とその情報を入れる

想定される入出力をインプットし回路の正しさをチェック

3. プログラム修正



実行例