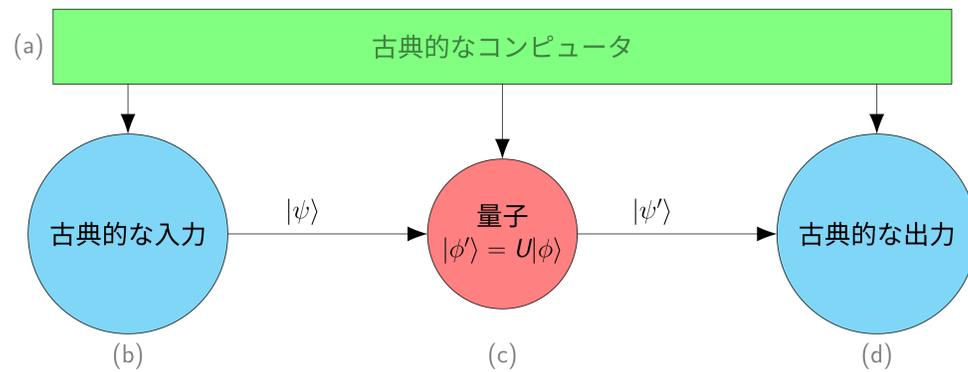


の概要

- 量子コンピューティングは、本格的な量子コンピュータ上で特定の ion 古典的な問題を高速化します
- イオントラップ、超伝導ループ、量子フォトニクス、トポロジカル量子コンピューティング、中性原子、ダイヤモンド空孔などは、実験的には小規模で実験的な規模でしか利用できないが、量子コンピューティングはムーアの法則に対する潜在的な解決策である。
- 量子コンピューティングの各技術には一定の制約と限界がある
- しかし、現在の量子コンピュータの応用が不十分である主な理由の1つは、この新しい技術を効率的に活用する方法についての統一された理解がないことです。
- 私たちの一般的な関心は、より多くの利点を示す新しいアルゴリズムを見つけることで、量子コンピューティングをより有用にする方法を理解することです

量子設計問題を理解する



- 量子初期化: 入力値は、大きさベース: $|0\rangle$ 、位相ベース: $\begin{pmatrix} 1 \\ 1 \end{pmatrix}$ 、 $\begin{pmatrix} 1 \\ -1 \end{pmatrix}$ など、さまざまな方法でエンコードできます
- 量子モデリング: 古典的な問題を解くための量子アルゴリズムを設計するには、与えられた古典的なアルゴリズムが量子コンピュータによって加速される可能性がある場所を理解する必要があります。
- 量子進化は、量子計算を説明するために使用される用語であり、次のように入量子状態を出力量子状態に変換する一連のユニタリ演算子を使用して実行されます。 $|\phi'\rangle = U|\phi\rangle$
- 量子処理: ただし、量子ビット状態のデコヒーレンスにより計算時間が非常に短いため、量子アルゴリズムは計算可能である必要があります。
- 演算子 U の構築または実際の計算は、実現が困難であったりコストがかかったりすることがあります。
- 結果の取得: 量子コンピュータから取得された結果は、取得するか、または全体的なアルゴリズムのさらに別の部分で使用する必要があることを意味します。
- 基本的な要素を理解することで、量子コンピューティングの機能をより有効に活用し、高速量子コンピューティングを実行できるようになります。

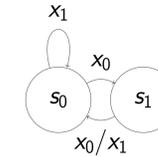
古典的問題に対する量子アルゴリズムを設計する前に尋ねるべき質問

- 特定の形式化で古典的な問題を単純化できますか?
- 古典的なアルゴリズムでは、アプローチはツリー検索に基づいています。量子コンピュータはどうでしょうか? 表現にはどのような同等性が存在するのでしょうか?
- どうすれば実現できるでしょうか? 何が必要ですか? どのような問題表現が必要ですか? どのような量子アルゴリズムを使用すればよいでしょうか?
- 速度の向上はどの程度でしょうか? 問題は最終的に価値のあるものでなければなりません。

有限オートマトン (FA) のテストシーケンスの検索

Table 1. An example of a FA

| State | x_0 | x_1 |
|-------|-------|-------|
| s_0 | s_1 | s_0 |
| s_1 | s_0 | s_0 |



オートマトンとは、 $A = (S, X, \delta)$ の3つの要素です。ここで、 S と X は、それぞれ n 個の状態と k 個の入力シンボルの有限集合です。 $\delta: S \times X \rightarrow S$ は、 A の遷移関数です。

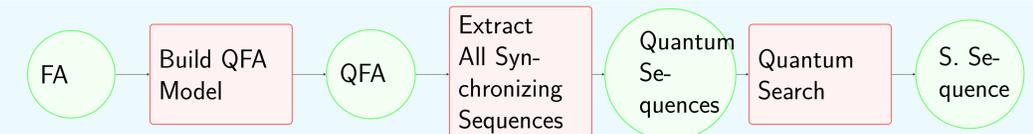
シーケンスのテスト

- ホーミング - 長さ l のシーケンス \bar{x} で、任意の初期状態 s に対して、 $\lambda(x_i, s)^{ol}$ から得られる出力シーケンス \bar{o} を観察すると、 $\delta(x_i, s)^{ol} = s'$ で与えられる最後の状態がわかります。
- 識別 - 長さ l のシーケンス \bar{x} で、2つの初期状態 $s \neq s'$ が与えられたときに、 \bar{o} と \bar{o}' を結果の出力シーケンスとすると、 $\bar{o} \neq \bar{o}'$ となるもの
- 同期 - 長さ l のシーケンス \bar{x} で、任意の2つの初期状態 $s \neq s'$ に対して、 $\delta(x_i, s)^{ol} = \delta(x_i, s')^{ol}$ となるもの
- より具体的で使用法に固有のもの: ペアごとの区別、セットのテストなど。

特定のシーケンスを見つける複雑さは、特定のケースでは NP 困難です:

| Type of Automata | Checking the existence | Upper bound | Finding shortest one |
|-------------------|------------------------|-----------------|----------------------|
| Partial Automata | PSPACE Complete | $n^2 * 4^{n/3}$ | PSPACE Complete |
| Complete Automata | $O(n^2)$ | $(n-1)^2$ | NP Hard |

量子アルゴリズムを用いたテストシーケンスの探索



例

Example

情報の表現

$$|S\rangle = |aw\rangle = |a\rangle \otimes |w\rangle = \alpha_0 \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} + \alpha_1 \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} + \alpha_2 \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} + \alpha_3 \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \alpha_0|0s_0\rangle + \alpha_1|0s_1\rangle + \alpha_2|1s_0\rangle + \alpha_3|1s_1\rangle$$

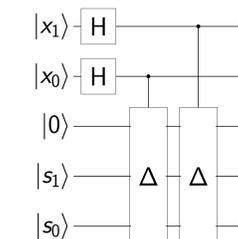
モデリング

状態遷移関数はユニタリ演算子 Δ で表される。

$$\Delta = \begin{pmatrix} \Delta_{x_0} & 0 & \dots & 0 \\ 0 & \dots & \dots & \vdots \\ \vdots & \dots & \dots & 0 \\ 0 & \dots & 0 & \Delta_{x_k} \end{pmatrix}$$

複数の QFA の処理

$$|Q\rangle = |X\rangle|S\rangle = \begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix} \otimes \begin{pmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{pmatrix} = \begin{pmatrix} \beta_0\alpha_0 \\ \beta_0\alpha_1 \\ \beta_0\alpha_2 \\ \beta_0\alpha_3 \\ \beta_1\alpha_0 \\ \beta_1\alpha_1 \\ \beta_1\alpha_2 \\ \beta_1\alpha_3 \end{pmatrix}$$



- 複数の QFA を並列に処理すると、生成されたすべての入力シーケンスに対するすべての出力を取得できます。
- 結果を抽出するには、何を検索するかを知る必要があります。識別シーケンスを探していますか?