# 量子コンピュータによる公平な抽選システムの開発

Yoshimura Hikaru（吉村 優）

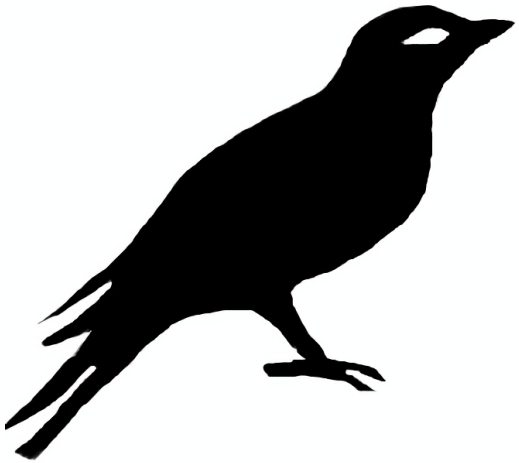
yyu@mental.poker
藤井 啓祐 PM

未踏ターゲット事業成果報告会
February 8, 2020

(y-yu/mitou-final-slide@defd6b3)

# 目次

# 自己紹介



- 筑波大学 情報学群情報科学類卒（学士）
  - 関数型言語の型システムの研究
- 株式会社リクルートマーケティングパートナーズ
  - 古典コンピュータを使ってサーバーサイドを作っている

| | |
|---|---|
| Twitter | @_yyu_ |
| Qiita | yyu |
| GitHub | y-yu |
| Facebook | h1karuy |

# プロジェクトの背景

# 抽選とは？

## 抽選

参加者が抽選の運営に対してお金を支払うと、**確率**で景品を入手できる構造

- 宝くじやソーシャルゲームのマネタイズ手段など社会で広く普及している

誰が確率を計算するの？

運営の古典コンピュータ？

運営の実装が正しい確率に従っているのか？

# 抽選とは？

- 「宝くじ」といった抽選では景品が 1 億円など高額になる
- ソーシャルゲームのガチャの不正疑惑により、運営会社の株価がストップ安になったり、本当に確率が実際と異っていたことがある😢

抽選の公平性への疑惑が会社の信用に影響することもある

- 抽選を公平にすることは社会的に非常に重要である

# 公平な抽選

## 公平な抽選

- 参加者にとっても運営にとっても、抽選による景品の出現確率が実装に基づいて明らかである
- 悪意を持つ参加者や、悪意を持つ運営による片方にとって有利な確率操作ができない

- 公平な抽選の方法として古典コンピュータではたとえば次のやり方がある
  1. **ブロックチェーン**を利用する方法
  2. **コミットメント**を利用する方法

# 1bit の抽選 — コイントス

1. 今、“紙” と “封筒” がある
2. アリスが紙に**表**または**裏**を書き込みそれを封筒に入れる
3. ボブがコイントスをする
4. アリスとボブは封筒を開封する
   - 紙に書かれた表裏とコイントスの結果が等しければ、アリスは景品を得る
   - そうでなければ景品を得ない

- このとき次の 2 つが重要となる
  1. ボブは封筒の中の紙に書かれた文字を読むことができない
  2. アリスは紙に書いた裏表を変更することができない
- 1 が隠蔽、2 が束縛となる

# 隠蔽と束縛

- コイントスが公平になるためには隠蔽・束縛が完全である必要がある
- ところが両方を完全にすることは、たとえ量子コンピュータを利用したとしてもできない*
- 古典コンピュータでは片方を完全に、もう片方を計算の複雑さに依存させることで解決している

つまり計算資源が多い方が有利！

完全な公平ではない

そこで**量子コンピュータ**を利用する

*このことは背理法で証明できる。

# プロジェクトの目的・成果

# プロジェクトの目的・成果

- 量子コンピュータのシミュレーターを利用し、それを用いて量子コンピュータを用いる抽選システムソフトウェアの開発
- 開発したソフトウェアの無償公開
  - https://github.com/y-yu/qrand

この後のスライドで一部を解説！

- 開発した抽選システムのドキュメント執筆、ソフトウェアの性能や課題の考察（今後の予定）

# 量子コイントス

- 量子コンピュータの確率的な性質を利用するコイントス

- 量子コンピュータのコイントスは不正が**等しい確率**で行える
  - たとえば 40%の確率で運営が不正できるならば、40%の確率で参加者も不正できる
  - 古典コンピュータではどちらか側しか不正できない
- あとは何度もコイントスしていくと任意の景品に対応させられる
- 量子コイントスはいくつかのプロトコルが提案されている [1, 2] などがある
- このスライドでロス耐性プロトコル [2] とデモ実装を紹介する

# 実装の説明

- 量子シミュレーターとして Qulacs[3] を利用
- Web サーバー・JSON 通信部分は Python と Flask を利用


- JSON の部分が実際には量子通信となる
- 今回はエミュレーターなので、1 量子ビットを表現する 2 つの複素数を JSON 形式でやりとりする
- やりとりするのは次のような 4 パターンの量子ビット

$$|\text{天国}\rangle \equiv \sqrt{0.9}\,|0\rangle + \sqrt{0.1}\,|1\rangle\,,\quad |\text{地獄}\rangle \equiv \sqrt{0.1}\,|0\rangle - \sqrt{0.9}\,|1\rangle$$

$$|\text{生}\rangle \equiv \sqrt{0.9}\,|0\rangle - \sqrt{0.1}\,|1\rangle\,,\quad |\text{死}\rangle \equiv \sqrt{0.1}\,|0\rangle + \sqrt{0.9}\,|1\rangle$$

# プロトコル

| $a$ \ $x$ | 0 | 1 |
|---|---|---|
| 0 | $\|$天国$\rangle$ | $\|$地獄$\rangle$ |
| 1 | $\|$生$\rangle$ | $\|$死$\rangle$ |

クライアント（参加者）

サーバー（運営）

❶ それぞれ 1 ビットの $a, x$ を用いて 4 つから
1 量子ビットを選びサーバーへ送る

❷ 1 ビット乱数 $\hat{a}, b$ を生成し
$\hat{a} = 0$ なら $\{|$天国$\rangle, |$地獄$\rangle\}$ で
$\hat{a} = 1$ なら $\{|$生$\rangle, |$死$\rangle\}$ で測定し
結果を $\hat{x}$ として保存し $b$ を送信

❸ $a, x$ を公開

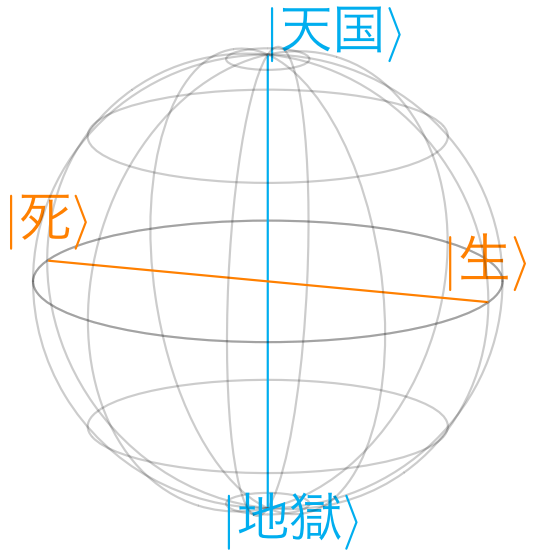
❹ $a = \hat{a}$ ならば $x = \hat{x}$ を検証

❺ $x$ XOR $b$ を結果とする

# 公平性の説明

- 参加者は送信する量子ビットを次の中から選択する

| $a$ \ $x$ | 0 | 1 |
|---|---|---|
| 0 | $\|$天国$\rangle$ | $\|$地獄$\rangle$ |
| 1 | $\|$生$\rangle$ | $\|$死$\rangle$ |




- 図のように $a$ が同じ（= 色が同じ）なら直交する
- たとえば量子ビット $|$地獄$\rangle$ の測定を考える

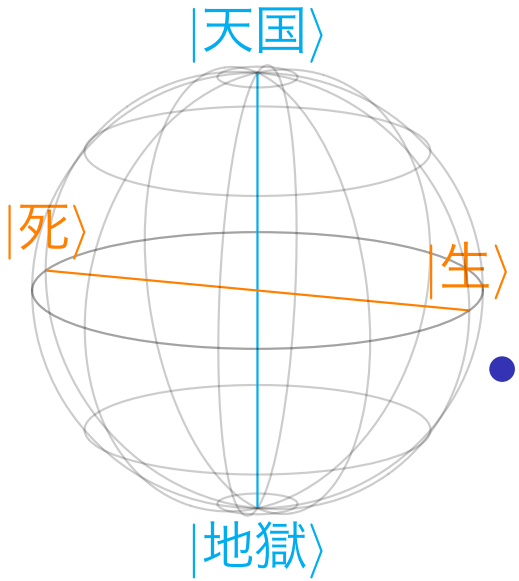
  $\{|$天国$\rangle, |$地獄$\rangle\}$ で測定した場合

  常に $|$地獄$\rangle$ が出力

  $\{|$生$\rangle, |$死$\rangle\}$ で測定した場合

  $|$生$\rangle, |$死$\rangle$ のいずれかが確率で出力

# 公平性の説明




- たとえば参加者が $|地獄\rangle (a = 0)$ を選んだとして、運営は $\{|天国\rangle, |地獄\rangle\}$ か $\{|生\rangle, |死\rangle\}$ のいずれで測定するか決める
  1. 運営が $\{|天国\rangle, |地獄\rangle\}$ を選べば**検証ができる**
  2. 一方で $\{|生\rangle, |死\rangle\}$ を選べば**検証ができない**
- つまり、運営は測定結果 $\hat{x}$ を得るが、これはコミットメントとなるのかランダムな出力なのか参加者の $a$ を知るまで分からない
- 参加者がこの後、どの量子ビットを利用したか公開するとき運営の得た測定結果 $\hat{x}$ を知らないため、コミットメントされているのかどうかが分からない
  - つまり確率でズルできる😈が、確率で失敗する
- このように量子コイントスでは古典のコミットメントのように計算量でどちらかが有利になるということはない

# デモ動画

- ここまで紹介したプロトコルを実際に試して、またチート（不正）してみることもできる機能のデモ
- https://github.com/y-yu/qrand

# まとめ

- なんらかの計算速度を向上させるという利用用途以外にも量子コンピュータにはこのような抽選への利用など様々な応用がある
- 今回の発表では時間の関係で割愛したが、参加者・運営が互いにチートする場合や、あえて運営が途中まで勝たせる戦略などがあり、それは近々文書にまとめて公開したい
- 運営 *vs* 参加者の抽選は公平に不正ができたとしても本当に公平であるのかについて引き続き考えたい
  - 抽選の景品に完全な順序がつく（たとえばお金）なら簡単だが、ゲームの景品などは必ずしも順序がつくとは限らない
  - したがって運営は参加者にとってコインの「裏か表のどちらが有利か？」ということ知らないことがある？
- 今回の実装は量子通信回線を前提としているのが実用化への課題である。また抽選の多人数化などについても今後検討したい

# 参考文献 I


[1] Charles H. Bennett and Gilles Brassard.
Quantum cryptography: Public key distribution and coin tossing.
*Theor. Comput. Sci.*, Vol. 560, pp. 7–11, 2014.

[2] Guido Berlín, Gilles Brassard, Félix Bussières, and Nicolas Godbout.
Fair loss-tolerant quantum coin flipping.
*Phys. Rev. A*, Vol. 80, p. 062321, Dec 2009.

[3] Qulacs(GitHub), 2018.

# Thank you for your attention!

# 質疑用のスライド

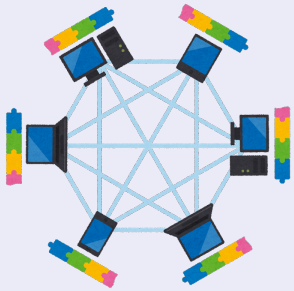
5. 古典コンピュータによる実装
   - ブロックチェーン
   - コミットメント

6. アリス・ボブの不正

# ブロックチェーン

## ブロックチェーン

- マスターが複数あるデータベース
- ブロックチェーンを管理する者（マイナー）は時間のかかるハッシュ計算をし報酬を得る
  - マイナーは 1 つ前のブロックのハッシュ値とマイナーが作成した乱数をハッシュ関数へ投入し、先頭 $n$ ビットが $0$ な場合に報酬を得る



- マイナーが計算する先頭 $n$ ビットが $0$ なハッシュ値は残りのビットが予測困難なため、それを公平な抽選へ利用する

# ブロックチェーン

- ブロックチェーンを利用した公平な抽選はマイナーが公平なことに依存する
  - マイナーが計算したハッシュ値による抽選の結果がマイナーにとって不都合な場合、ブロックチェーンの報酬を放棄することで有利になる
- マイナーは合理的な存在であると考えられるから抽選の景品の価値がブロックチェーンの報酬を越える場合、マイナーは抽選の景品を優先する

景品の価値がいくらのとき、マイナーは景品を優先するのか？

- 記事によると、景品の価値が**約 150 万円**以上のとき†
- 1 億円など高額が動く抽選へ利用した場合に完璧とは言えない

†この計算の当時とは為替が違うため、現在はもう少々必要である

# コミットメント

- 公平なコイントスは次のようなゲームを例に説明できる

1. 今、“紙” と “封筒” がある
2. アリスが紙に**表**または**裏**を書き込みそれを封筒に入れる
3. ボブがコイントスをする
4. アリスとボブは封筒を開封する
   - 紙に書かれた表裏とコイントスの結果が等しければ、アリスは景品を得る
   - そうでなければ景品を得ない

- このとき次の 2 つが重要となる
  1. ボブは封筒の中の紙に書かれた文字を読むことができない
  2. アリスは紙に書いた裏表を変更することができない
- 1 が隠蔽、2 が束縛となる

# コミットメント

- コイントスが公平になるためには隠蔽・束縛が完全である必要がある
- ところが両方を完全にすることは、たとえ量子コンピュータを利用したとしてもできない
  - 古典コンピュータでは片方を完全に、もう片方を計算の複雑さに依存させることで解決している

つまり計算資源が多い方が有利！

- これは完全な公平ではない

そこで**量子コンピュータ**を利用する

# アリス・ボブの不正

- プロトコルを意図的に無視すれば不正することができる
  - これは確率で不正をしようとしたことがバレてしまう
- アリスの最適な不正もボブの最適な不正も同じ割合で成功し同じ割合で失敗する

どうしてそうなるの？

そうなるように $\sqrt{0.9}$ とか $\sqrt{0.1}$ を選んだから！