

# 古典・量子ハイブリッドな高機能プログラミング言語の 設計及び処理系の開発 — Qitten —

平田賢吾 寺本幸生 米内貴志

古典・量子ハイブリッドな高級プログラミング言語 Qitten の設計・開発を行った

量子に馴染みがないプログラマにとっても書きやすい 言語で量子ソフトウェアの開発を加速させる

## ハイブリッドな記述

Shorのアルゴリズム、HHLアルゴリズムなどの古典・量子ハイブリッドアルゴリズムを一つの言語で全て記述可能

古典・量子の境界をなくす

## 線形型システム

量子複製不可能定理に反するプログラムを型エラーとして検出する

```
let y = gates::X(x);  
:  
let z = gates::H(x);
```

**Type error:**  
量子ビット x は複製できません

プログラマを予期せぬエラーから守る

## Qitten の特徴

✓記述性

✓安全性

✓実用性

## 量子 if などの高級文法

```
fn main() {  
  ...  
  qif flag1 {  
    gates::X(targ);  
    qif flag2 {  
      gates::Z(targ)  
    }  
  } else {  
    gates::Y(targ)  
  }  
  ...  
}
```

制御付き量子操作を直感的に

## コンパイル可能

IBM Quantum, IonQ などの実機上でプログラムを実行できる

実際に計算を実行できる



IonQ

# 開発内容の全体像

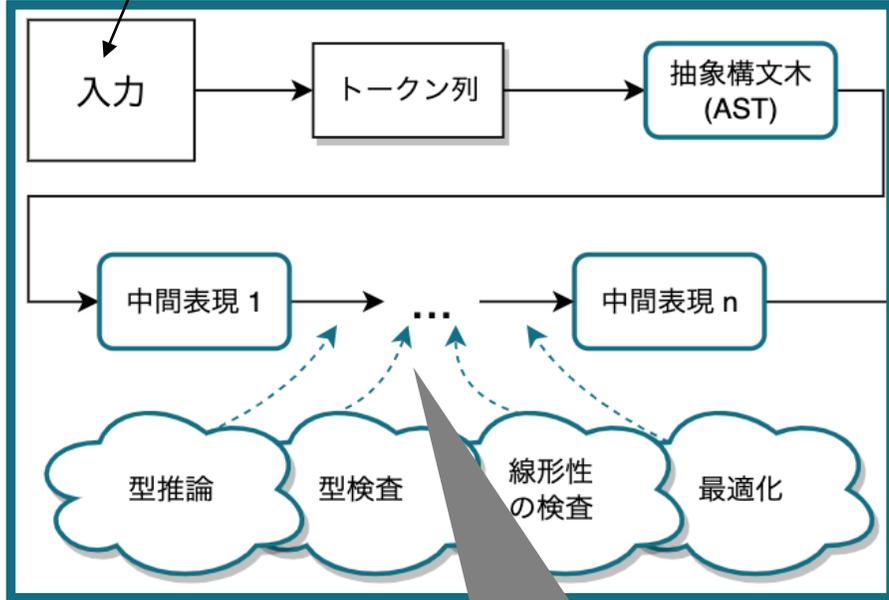
```
fn main() -> i32 {  
  let first = toss();  
  if first == 1 {  
    1 + toss()  
  } else {  
    0  
  }  
}  
  
fn toss() -> i32 {  
  let a = qfalse;  
  let b = H(a);  
  let c = measure(b);  
  if c {  
    1  
  } else {  
    0  
  }  
}
```

Qitten プログラム

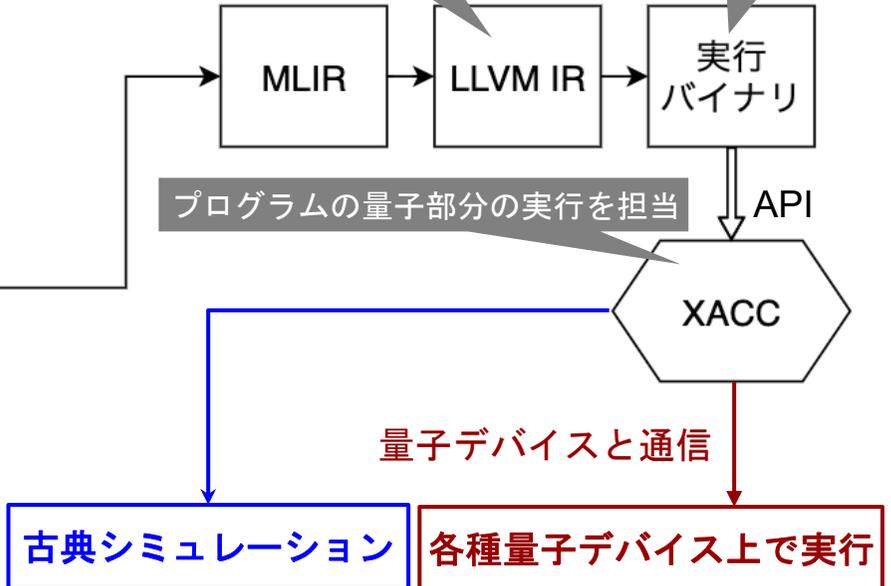
Rust 言語ライクな文法

デモ動画  
<https://www.youtube.com/watch?v=Dlt1ufYQZws>

古典プログラムについては LLVM の最適化を利用  
プログラムの古典部分は古典コンピュータ上で実行  
量子部分は XACC を使用



プログラムから量子回路を合成



量子デバイスと通信



## Qitten コンパイラ