Symbolic Evaluation of Hadamard-Toffoli Quantum Circuits

Jacques Carette, Amr Sabry, Gerardo Ortiz

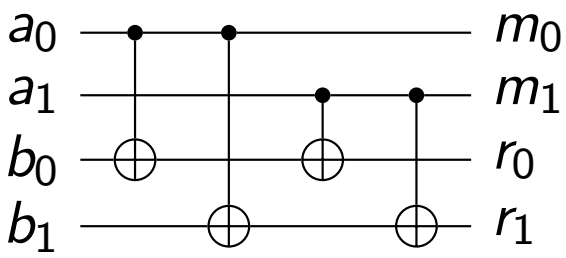
January 16, 2023

McMaster University

The Idea

Traditional partial evaluation and symbolic evaluation techniques apply to an interesting class of quantum circuits with surprising effectiveness!

Reversible gates



$$\begin{aligned} \mathrm{CX}(0, b) &= (0, b) \\ \mathrm{CX}(1, b) &= (1, \overline{b}) \end{aligned}$$

So, writing inputs/outputs horizontally:

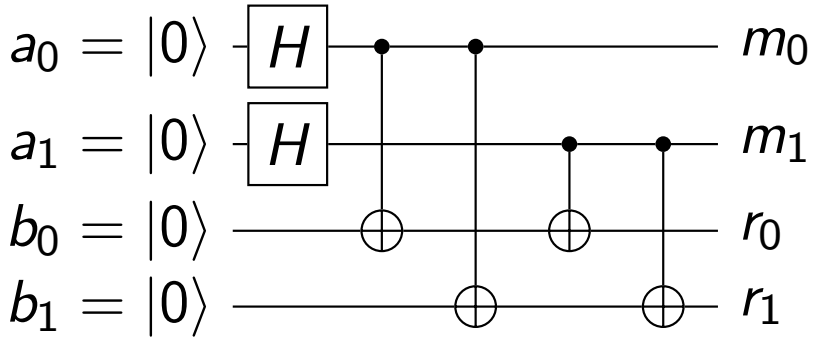
$$\begin{aligned} |0111\rangle &\Rightarrow |0111\rangle \\ &\Rightarrow |0111\rangle \\ &\Rightarrow |0101\rangle \\ &\Rightarrow |0100\rangle \end{aligned}$$

$$\begin{aligned} |1010\rangle &\Rightarrow |1000\rangle \\ &\Rightarrow |1001\rangle \\ &\Rightarrow |1001\rangle \\ &\Rightarrow |1001\rangle \end{aligned}$$

$$\begin{aligned} |1100\rangle &\Rightarrow |1110\rangle \\ &\Rightarrow |1111\rangle \\ &\Rightarrow |1101\rangle \\ &\Rightarrow |1100\rangle \end{aligned}$$

$$\begin{aligned} |1111\rangle &\Rightarrow |1101\rangle \\ &\Rightarrow |1100\rangle \\ &\Rightarrow |1110\rangle \\ &\Rightarrow |1111\rangle \end{aligned}$$

Superpositions

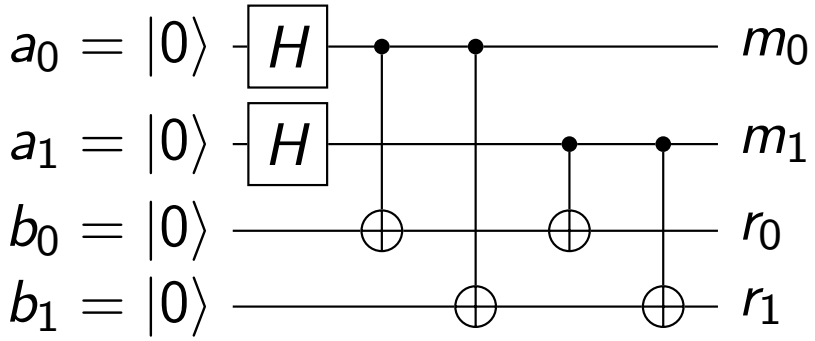


$$H|0\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

So the top two wires are in the state

$$\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = \frac{1}{2}(|00\rangle + |10\rangle + |01\rangle + |11\rangle)$$

Superpositions



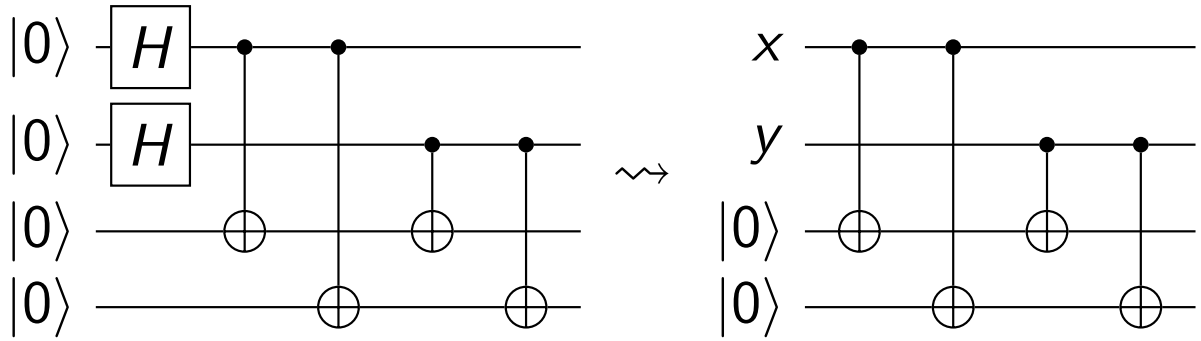
$$H|0\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

Thus for the whole circuit:

$$\begin{aligned}
|0000\rangle &\Rightarrow \frac{1}{2}(|0000\rangle + |0100\rangle + |1000\rangle + |1100\rangle) \\
&\Rightarrow \frac{1}{2}(|0000\rangle + |0100\rangle + |1010\rangle + |1110\rangle) \\
&\Rightarrow \frac{1}{2}(|0000\rangle + |0100\rangle + |1011\rangle + |1111\rangle) \\
&\Rightarrow \frac{1}{2}(|0000\rangle + |0110\rangle + |1011\rangle + |1101\rangle) \\
&\Rightarrow \frac{1}{2}(|0000\rangle + |0111\rangle + |1011\rangle + |1100\rangle)
\end{aligned}$$

Symbolic Execution (new)

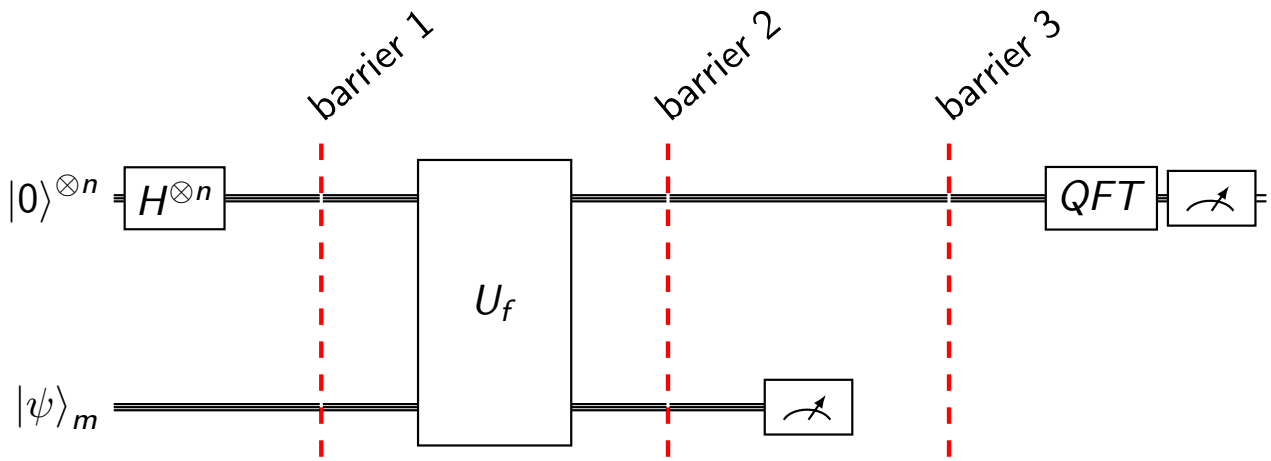
Replace $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ by a symbolic variable:



Maintain formula in algebraic normal form (ANF).
So

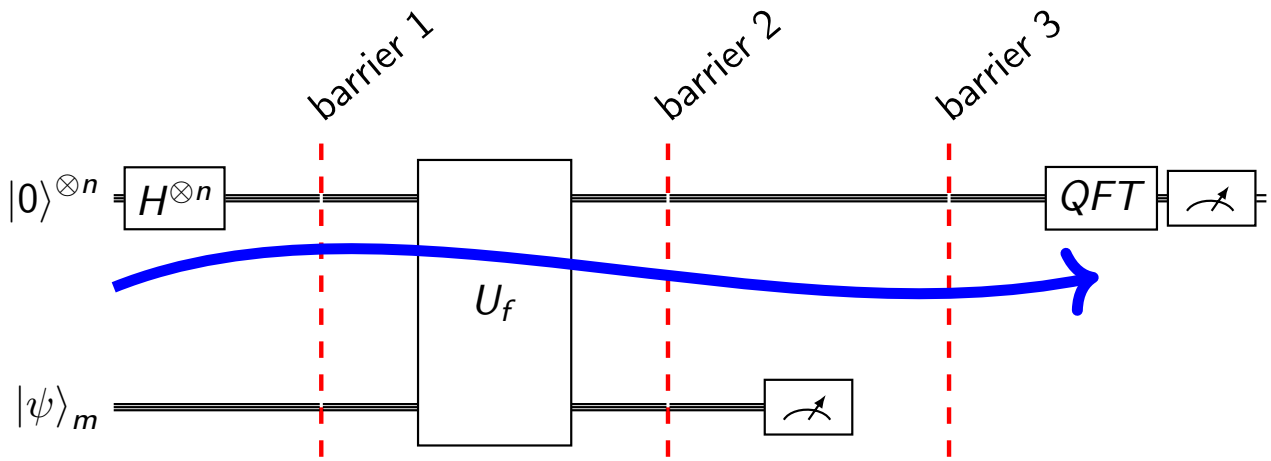
$$\begin{aligned} |xy00\rangle &\Rightarrow |xyx0\rangle \\ &\Rightarrow |xyxx\rangle \\ &\Rightarrow |xy(x \oplus y)x\rangle \\ &\Rightarrow |xy(x \oplus y)(x \oplus y)\rangle \end{aligned}$$

Quantum Circuits



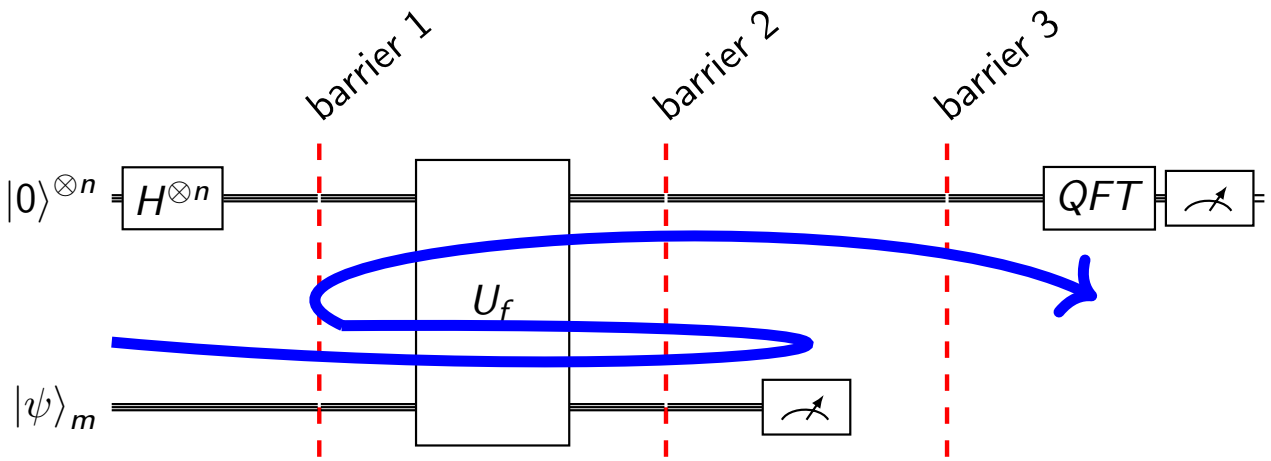
- U_f is a *classical reversible* circuit representing f, i.e. $U_f(\mathbf{x}, \mathbf{y}) = |\mathbf{x} \quad (\mathbf{y} \oplus \mathbf{f}(\mathbf{x}))\rangle$.
- H is the Hadamard gate; introduces *quantum parallelism* to evaluate U_f for many inputs simultaneously.
- QFT is the Quantum Fourier Transform used to analyze the spectral properties of the output.

Quantum Circuits



- U_f is a *classical reversible* circuit representing f, i.e. $U_f(\mathbf{x}, \mathbf{y}) = |\mathbf{x} \quad (\mathbf{y} \oplus \mathbf{f}(\mathbf{x}))\rangle$.
- H is the Hadamard gate; introduces *quantum parallelism* to evaluate U_f for many inputs simultaneously.
- QFT is the Quantum Fourier Transform used to analyze the spectral properties of the output.

Quantum Circuits Retrodictive Evaluation



- U_f is a *classical reversible* circuit representing f, i.e. $U_f(\mathbf{x}, \mathbf{y}) = |\mathbf{x} \quad (\mathbf{y} \oplus \mathbf{f}(\mathbf{x}))\rangle$.
- H is the Hadamard gate; introduces *quantum parallelism* to evaluate U_f for many inputs simultaneously.
- QFT is the Quantum Fourier Transform used to analyze the spectral properties of the output.

McMaster University

Examples

1. Deutsch
2. Deutsch-Jozsa
3. Bernstein-Varizani
4. Simon
5. Grover
6. Shor

McMaster University

Examples

1. Deutsch
2. Deutsch-Jozsa
3. Bernstein-Varizani
4. Simon
5. Grover
6. Shor

Examples: Deutsch and Deutsch-Josza

Definition

A boolean function is **balanced** *if it outputs the same number of 0/1 outputs.*

Deutsch:

Problem

Given $f : \mathbb{B} \to \mathbb{B}$, *decide if* f *is constant or balanced.*

Deutsch-Josza:

Problem

Given $f : \mathbb{B}^n \to \mathbb{B}$, *where* f *is known to be constant or balanced, decide which one it is.*

Examples: Grover and Shor

Grover:

Problem

Given $f : \mathbb{B}^n \to \mathbb{B}$ *where there exists a unique* u *such that* $f(u) = 1$. *Find* u.

Shor:

Problem

Factor a given N. *Do this by using* $f(x) = a^x \mod N$ *for suitable* a *and* $f : \mathbb{B}^Q \to \mathbb{B}^n$ *with* $Q = \lceil \log_2 (N^2) \rceil$, $n = \lceil \log_2 N \rceil$.

Requirements

Variabilities:

1. multiple representations of boolean values,
2. multiple representations of boolean formulae,
3. different evaluation means (directly, symbolically, forwards, backwards, retrodictive).

Possible to implement:

4. a reusable representation of our circuits,
5. a reusable representation of the inputs, outputs and ancillae,
6. a synthesis algorithm for $\mathbb{B}^n \to \mathbb{B}$ functions
7. a reusable library of circuits

Also, non-functional characteristics to hold:

8. evaluation of reasonably-sized circuits should be reasonably efficient.

Design


```
class (Show v, Enum v) => Value v where
  zero :: v
  one  :: v
  snot :: v -> v
  sand :: v -> v -> v
  sxor :: v -> v -> v

  -- has a default implementation
  snand :: [v] -> v -- n-ary and
  snand = foldr sand one
```

Implemented four times.

Design


```
class (Show v, Enum v) => Value v where
  zero :: v
  one  :: v
  snot :: v -> v
  sand :: v -> v -> v
  sxor :: v -> v -> v

  -- has a default implementation
  snand :: [v] -> v -- n-ary and
  snand = foldr sand one
```

Implemented four times.

```
data VarInFormula f v = FR
  { fromVar  :: v -> f
  , fromVars :: Int -> v -> [ f ]
  }
```

Design

First, naïve implementation:

```
newtype Ands = Ands { lits :: [String] }
  deriving (Eq, Ord)

(&&&) :: Ands -> Ands -> Ands
(Ands lits1) &&& (Ands lits2) =
  Ands (lits1 ++ lits2)

newtype Formula = Formula { ands :: [Ands]}
```

Design

Second, using sets of ints and occurence maps:

```
type Literal = Int
newtype Ands = Ands { lits :: IS.IntSet }

{-# INLINABLE compAnds #-}
compAnds :: Ands -> Ands -> Ordering
compAnds (Ands a1) (Ands a2) =
  compare (IS.toAscList a1) (IS.toAscList a2)

(&&&) :: Ands -> Ands -> Ands
(Ands lits1) &&& (Ands lits2) =
  Ands (IS.union lits1 lits2)

-- raw XOR formulas
type XORFU = Map.Map Ands Int
-- Normalized XOR formulas, i.e occur 0 or 1 time
newtype Formula = Formula { ands :: MS.MultiSet Ands }
```

Design

Third, using bitmaps and occurence maps:

```
type Literal = Natural
newtype Ands = Ands { lits :: Literal }

{-# INLINABLE compAnds #-}
compAnds :: Ands -> Ands -> Ordering
compAnds (Ands a1) (Ands a2) = compare a1 a2

(&&&) :: Ands -> Ands -> Ands
(Ands lits1) &&& (Ands lits2) = Ands (lits1 .|. lits2)

type Occur = Int
-- Raw XOR formulas
type XORFU = Map.Map Ands Occur
-- Normalized XOR formulas, i.e occur 0 or 1 time
type XORF = MS.MultiSet Ands
```

Running Deutsch-Josza

- Retrodictive once on output measurement 0
- Output symbolic formula that decides $f(|\mathbf{x}\rangle) = 0$.
- f constant $\Rightarrow 0 = 0$ or $1 = 0$ regardless of circuit size.

Running Deutsch-Josza

- Retrodictive once on output measurement 0
- Output symbolic formula that decides $f(|\mathbf{x}\rangle) = 0$.
- f constant $\Rightarrow 0 = 0$ or $1 = 0$ regardless of circuit size.

Sample outputs:

- $x_0 = 0$,
- $x_0 \oplus x_1 \oplus x_2 \oplus x_3 \oplus x_4 \oplus x_5 = 0$, and
- $1 \oplus x_3x_5 \oplus x_2x_4 \oplus x_1x_5 \oplus x_0x_3 \oplus x_0x_2 \oplus x_3x_4x_5 \oplus x_2x_3x_5 \oplus x_1x_3x_5 \oplus x_0x_3x_5 \oplus x_0x_1x_4 \oplus x_0x_1x_2 \oplus x_2x_3x_4x_5 \oplus x_1x_3x_4x_5 \oplus x_1x_2x_4x_5 \oplus x_1x_2x_3x_5 \oplus x_0x_3x_4x_5 \oplus x_0x_2x_4x_5 \oplus x_0x_2x_3x_5 \oplus x_0x_1x_4x_5 \oplus x_0x_1x_3x_5 \oplus x_0x_1x_3x_4 \oplus x_0x_1x_2x_4 \oplus x_0x_1x_2x_4x_5 \oplus x_0x_1x_2x_3x_5 \oplus x_0x_1x_2x_3x_4 = 0$.

How to decide? If it mentions a variable, it's balanced.
We tested all 12872 functions $\mathbb{B}^6 \to \mathbb{B}$.

Running Deutsch-Josza

- Retrodictive once on output measurement 0
- Output symbolic formula that decides $f(|\mathbf{x}\rangle) = 0$.
- f constant $\Rightarrow 0 = 0$ or $1 = 0$ regardless of circuit size.

Sample outputs:

- $x_0 = 0$,
- $x_0 \oplus x_1 \oplus x_2 \oplus x_3 \oplus x_4 \oplus x_5 = 0$, and
- $1 \oplus x_3x_5 \oplus x_2x_4 \oplus x_1x_5 \oplus x_0x_3 \oplus x_0x_2 \oplus x_3x_4x_5 \oplus x_2x_3x_5 \oplus x_1x_3x_5 \oplus x_0x_3x_5 \oplus x_0x_1x_4 \oplus x_0x_1x_2 \oplus x_2x_3x_4x_5 \oplus x_1x_3x_4x_5 \oplus x_1x_2x_4x_5 \oplus x_1x_2x_3x_5 \oplus x_0x_3x_4x_5 \oplus x_0x_2x_4x_5 \oplus x_0x_2x_3x_5 \oplus x_0x_1x_4x_5 \oplus x_0x_1x_3x_5 \oplus x_0x_1x_3x_4 \oplus x_0x_1x_2x_4 \oplus x_0x_1x_2x_4x_5 \oplus x_0x_1x_2x_3x_5 \oplus x_0x_1x_2x_3x_4 = 0.$

How to decide? If it mentions a variable, it's balanced.
We tested all 12872 functions $\mathbb{B}^6 \to \mathbb{B}$.

Running Grover

Recall: guess the position of the single 1 bit.

$n = 4$, w in the range $\{0..15\}$

$u = 0$ $1 \oplus x_3 \oplus x_2 \oplus x_1 \oplus x_0 \oplus x_2x_3 \oplus x_1x_3 \oplus x_1x_2 \oplus x_0x_3 \oplus x_0x_2 \oplus x_0x_1 \oplus x_1x_2x_3 \oplus x_0x_2x_3 \oplus x_0x_1x_3 \oplus x_0x_1x_2 \oplus x_0x_1x_2x_3$

$u = 1$ $x_0 \oplus x_0x_3 \oplus x_0x_2 \oplus x_0x_1 \oplus x_0x_2x_3 \oplus x_0x_1x_3 \oplus x_0x_1x_2 \oplus x_0x_1x_2x_3$

$u = 2$ $x_1 \oplus x_1x_3 \oplus x_1x_2 \oplus x_0x_1 \oplus x_1x_2x_3 \oplus x_0x_1x_3 \oplus x_0x_1x_2 \oplus x_0x_1x_2x_3$

$u = 3$ $x_0x_1 \oplus x_0x_1x_3 \oplus x_0x_1x_2 \oplus x_0x_1x_2x_3$

$u = 4$ $x_2 \oplus x_2x_3 \oplus x_1x_2 \oplus x_0x_2 \oplus x_1x_2x_3 \oplus x_0x_2x_3 \oplus x_0x_1x_2 \oplus x_0x_1x_2x_3$

$u = 5$ $x_0x_2 \oplus x_0x_2x_3 \oplus x_0x_1x_2 \oplus x_0x_1x_2x_3$

$u = 6$ $x_1x_2 \oplus x_1x_2x_3 \oplus x_0x_1x_2 \oplus x_0x_1x_2x_3$

$u = 7$ $x_0x_1x_2 \oplus x_0x_1x_2x_3$

$u = 8$ $x_3 \oplus x_2x_3 \oplus x_1x_3 \oplus x_0x_3 \oplus x_1x_2x_3 \oplus x_0x_2x_3 \oplus x_0x_1x_3 \oplus x_0x_1x_2x_3$

$u = 9$ $x_0x_3 \oplus x_0x_2x_3 \oplus x_0x_1x_3 \oplus x_0x_1x_2x_3$

$u = 10$ $x_1x_3 \oplus x_1x_2x_3 \oplus x_0x_1x_3 \oplus x_0x_1x_2x_3$

$u = 11$ $x_0x_1x_3 \oplus x_0x_1x_2x_3$

$u = 12$ $x_2x_3 \oplus x_1x_2x_3 \oplus x_0x_2x_3 \oplus x_0x_1x_2x_3$

$u = 13$ $x_0x_2x_3 \oplus x_0x_1x_2x_3$

$u = 14$ $x_1x_2x_3 \oplus x_0x_1x_2x_3$

$u = 15$ $x_0x_1x_2x_3$

Shor

Problem

Factor a given N. Do this by using $f(x) = a^x \mod N$ for suitable a and $f : \mathbb{B}^Q \to \mathbb{B}^n$ with $Q = \lceil \log_2 (N^2) \rceil$, $n = \lceil \log_2 N \rceil$.

Factoring 15:

Base	Equations			Solution
$a = 11$	$x_0 = 0$			$x_0 = 0$
$a = 4, 14$	$1 \oplus x_0 = 1$	$x_0 = 0$		$x_0 = 0$
$a = 7, 13$	$1 \oplus x_1 \oplus x_0 x_1 = 1$	$x_0 x_1 = 0$	$x_0 \oplus x_1 \oplus x_0 x_1 = 0$	$x_0 \oplus x_0 x_1 = 0$	$x_0 = x_1 = 0$
$a = 2, 8$	$1 \oplus x_0 \oplus x_1 \oplus x_0 x_1 = 1$	$x_0 x_1 = 0$	$x_1 \oplus x_0 x_1 = 0$	$x_0 \oplus x_0 x_1 = 0$	$x_0 = x_1 = 0$

Auto-generated circuits: 56,538 generalized Toffoli gates.

Shor

Problem

Factor a given N. Do this by using $f(x) = a^x \mod N$ for suitable a and $f : \mathbb{B}^Q \to \mathbb{B}^n$ with $Q = \lceil \log_2 (N^2) \rceil$, $n = \lceil \log_2 N \rceil$.

Factoring 15:

Base	Equations			Solution
$a = 11$	$x_0 = 0$			$x_0 = 0$
$a = 4, 14$	$1 \oplus x_0 = 1$	$x_0 = 0$		$x_0 = 0$
$a = 7, 13$	$1 \oplus x_1 \oplus x_0 x_1 = 1$	$x_0 x_1 = 0$	$x_0 \oplus x_1 \oplus x_0 x_1 = 0$	$x_0 \oplus x_0 x_1 = 0$	$x_0 = x_1 = 0$
$a = 2, 8$	$1 \oplus x_0 \oplus x_1 \oplus x_0 x_1 = 1$	$x_0 x_1 = 0$	$x_1 \oplus x_0 x_1 = 0$	$x_0 \oplus x_0 x_1 = 0$	$x_0 = x_1 = 0$

Auto-generated circuits: 56,538 generalized Toffoli gates.

For 3*65537=196611 (4,328,778 gates), 16 small equations that refer to just the four variables x_0, x_1, x_2, and x_3 constraining them to be all 0, i.e., asserting that the period is 16.

Timings: Deutsch-Jozsa

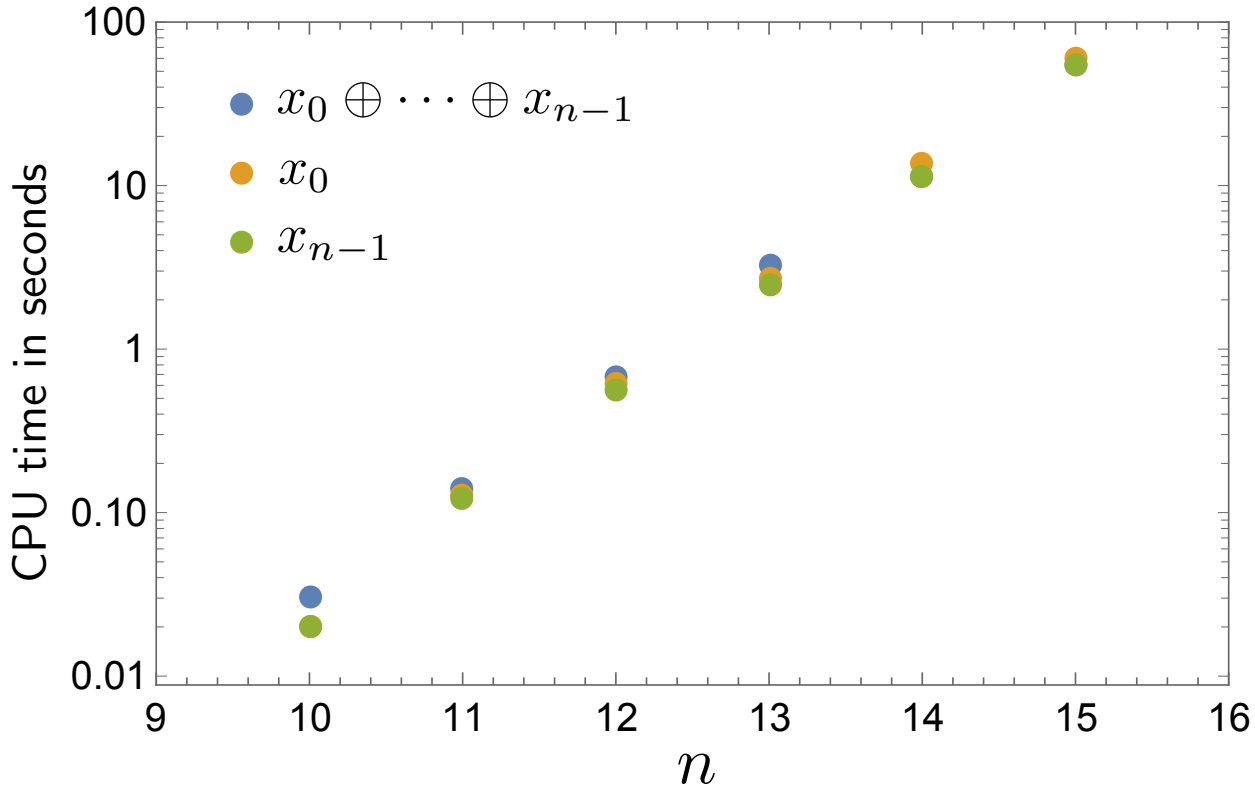


Figure: Retrodictive execution on 3 balanced functions at different sizes

Timings: Grover on different values

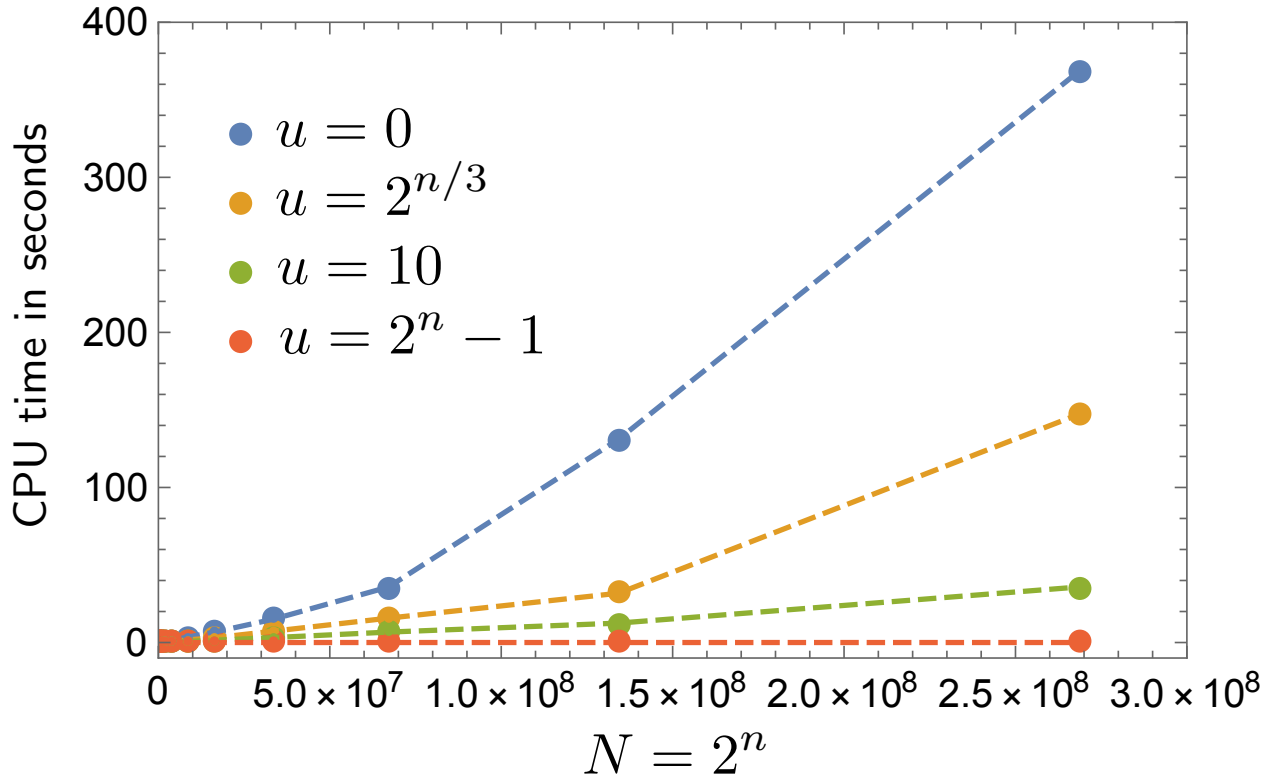


Figure: Retrodictive execution of the Grover algorithm on different "secret" values u at different sizes

Timings: Grover with different representations

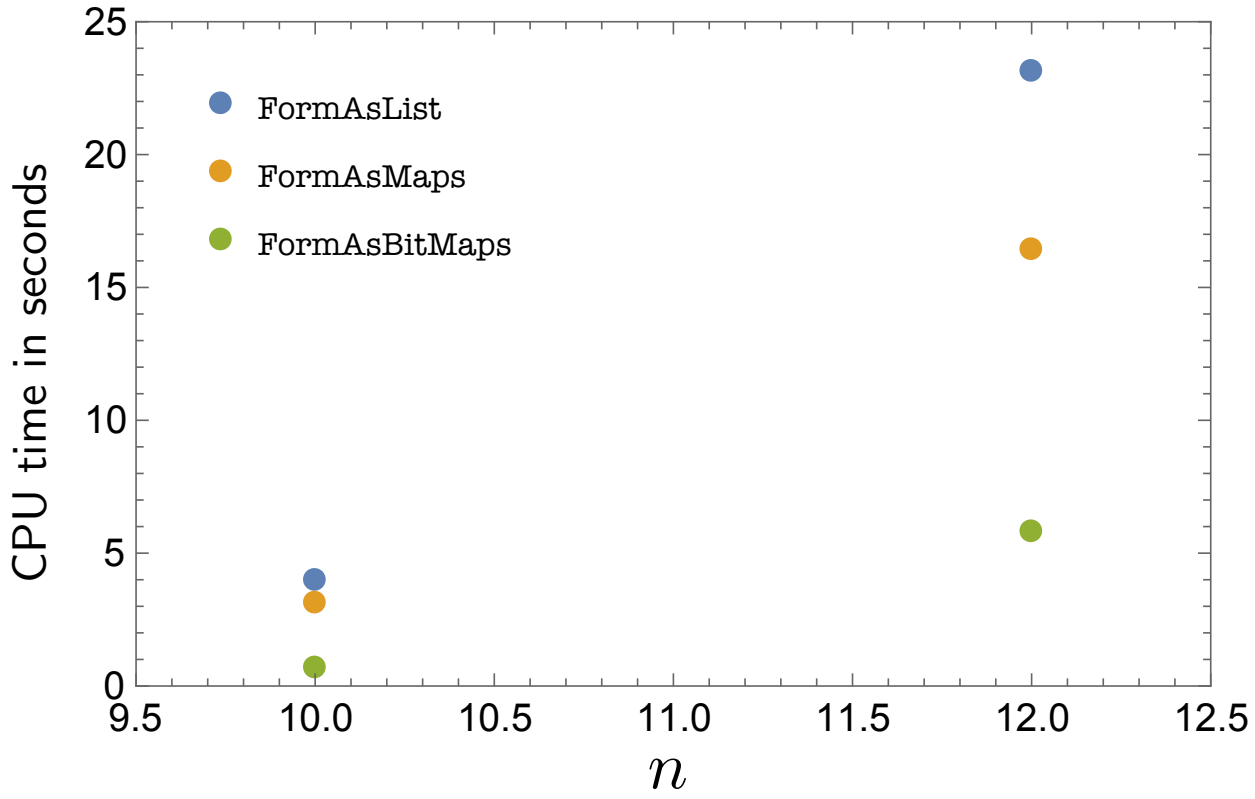


Figure: Retrodictive execution of the Grover algorithm on secret value $u = 0$ at two sizes but using different ANF representations

Complexity

Problem: circuit with T generalized Toffoli gates over $n+m$ qubits split into two registers (A,B).

1. Design oracle / circuit.
2. Let $A = |00\ldots0\rangle$ and $B = |00\ldots0\rangle$ and run classically. $\mathcal{O}(T)$. Leaves A intact, produce b in B.
3. Run backwards (symbolically) with $A = |x_{n-1}\ldots x_1 x_0\rangle$ and $B = |b\rangle$. Worst case, equations sized $\mathcal{O}(2^n)$, so $\mathcal{O}(Tm2^n)$.
4. Answer by inspecting/solving resulting m equations.

Complexity

Problem: circuit with T generalized Toffoli gates over $n+m$ qubits split into two registers (A,B).

1. Design oracle / circuit.
2. Let $A = |00\ldots0\rangle$ and $B = |00\ldots0\rangle$ and run classically. $\mathcal{O}(T)$. Leaves A intact, produce b in B.
3. Run backwards (symbolically) with $A = |x_{n-1}\ldots x_1 x_0\rangle$ and $B = |b\rangle$. Worst case, equations sized $\mathcal{O}(2^n)$, so $\mathcal{O}(Tm2^n)$.
4. Answer by inspecting/solving resulting m equations.

Bottlenecks:

- step (3) at worst case $\mathcal{O}(Tm2^n)$,
- step (4) "solve", which is NP-complete.

However:

- Sometimes 'typical' case has expected run-time depending on bit-size of the information contained in the answer.

McMaster University

Conclusion

- Different algorithms need different analyses to extract answer.
- (Not shown in slides) Shor seems to need qutrits/qudits in general.
- For a number of classical quantum algorithms, classical but symbolic execution works just as well.
- Even better: symbolic execute once instead of "enough" times to get probabilistic answer.
- Symbolic execution sometimes is really bad.

McMaster University

Conclusion

- Different algorithms need different analyses to extract answer.
- (Not shown in slides) Shor seems to need qutrits/qudits in general.
- For a number of classical quantum algorithms, classical but symbolic execution works just as well.
- Even better: symbolic execute once instead of "enough" times to get probabilistic answer.
- Symbolic execution sometimes is really bad.

- Speculation: some common quantum algorithms might not need quantum after all.

Cost Models

- White-box model: *We* implement U_f and the cost of implementing it and the cost of using it is counted as part of the overall complexity
- Black-box model: "Someone" implements U_f and gives us access to it; the complexity analysis only counts the number of times U_f is used. There are different cases based on what kind of access we are given:
  - Query U_f on a classical input
  - Query U_f on a quantum superposition
  - Query U_f on a *symbolic formula* (**NEW!**)