Symbolic Retrodictive Execution of Quantum Circuits

Jacques Carette Gerardo Ortiz Amr Sabry

January 12, 2022

Abstract

The quantum circuit model consists of two classes of gates: (i) quantum counterparts to classical reversible gates (e.g., Toffoli gates), and (ii) genuine quantum gates with no classical counterpart (e.g., Hadamard and phase gates). We make the remarkable observation, that, for a number of quantum algorithms, judicious reasoning about the classical components, ignoring all the quantum gates, is sufficient. Put differently, in those cases, the quantum gates serve no fundamental purpose and are actually distracting from an underlying efficient classical algorithm. The result relies on the ability to symbolically execute circuits, especially in a retrodictive fashion, i.e., by making partial observations at the output site and proceeding backwards to infer the implied initial conditions.

1 Introduction

- Retrodictive execution more efficient in some cases. What cases?
- Here are three examples: Deutsch-Jozsa, Simon, Shor when period is close to a power of 2
- Symbolic (retrodictive) evaluation as a broader perspective to classical computation
- Symbolic execution allows you to express/discover interference via shared variables
- When interference pattern is simple symbolic execution reveals solutions faster (and completely classically)
- Symbolic execution as a "classical waves" computing paradigm

2 A Complete Example

Consider the small circuit in Fig. 2. The scenario we are investigating is the following. The initial state of c_2, c_1, c_0 is unknown but both the initial state and final state of q_1, q_0 are known. The initial state is clearly 00 and let's assume for the remaining of this example that the final state is 01. The question is what can we infer about c_2, c_1, c_0? To answer the question, we will symbolically evaluate the circuit starting from the final state $c_0, c_1, c_2, 1, 0$ and going backwards towards the initial state as shown in Fig. 1. In step (1), we encounter a `cx` acting on q_1 and q_0 which are known. In step (2), we are not so lucky: we encounter a `ccx` gate with one unknown control wire but all hope is not lost. The action of `ccx` a b c is to update c to be $a \wedge b \oplus c$ where $\wedge$ is boolean conjunction (often omitted when clear from context) and $\oplus$ is the exclusive-or operation. In step (2), this means that the target wire q_1 should be updated to $1 \wedge c_0 \oplus 0$ which simplifies to c_0.

At the end of the retrodictive execution, we conclude that $q_0 = 1 \oplus c_0 \oplus c_1$ and $q_1 = c_0 \oplus c_2$ which needs to reconciled with the initial condition that $q_0 = q_1 = 0$. Solving these equations gives two possible solutions for $c_2 j, c_1, c_0$: either $c_2, c_1, c_0 = 010$ or $c_2, c_1, c_0 = 101$.

Another perspective on this analysis is the following. Assume c_2, c_1, c_0 started in an equal superposition and that q_1, q_0 were measured to be 01 after applying the circuit to the incoming superposition. The

$$c_0\ c_1\ c_2\ (q_0 = 1)\ (q_1 = 0) \quad (0)$$

$(\texttt{cx}\ q_1\ q_0)$

$$c_0\ c_1\ c_2\ (q_0 = 1)\ (q_1 = 0) \quad (1)$$

$(\texttt{ccx}\ c_0\ q_0\ q_1)$

$$c_0\ c_1\ c_2\ (q_0 = 1)\ (q_1 = c_0) \quad (2)$$

$(\texttt{cx}\ q_1\ q_0)$

$$c_0\ c_1\ c_2\ (q_0 = 1 \oplus c_0)\ (q_1 = c_0) \quad (3)$$

$(\texttt{x}\ q_1)$

$$c_0\ c_1\ c_2\ (q_0 = 1 \oplus c_0)\ (q_1 = 1 \oplus c_0) \quad (4)$$

$(\texttt{ccx}\ c_0\ q_1\ q_0)$

$$c_0\ c_1\ c_2\ (q_0 = 1 \oplus c_0)\ (q_1 = 1 \oplus c_0) \quad (5)$$

$(\texttt{x}\ q_1)$

$$c_0\ c_1\ c_2\ (q_0 = 1 \oplus c_0)\ (q_1 = c_0) \quad (6)$$

$(\texttt{cx}\ q_1\ q_0)$

$$c_0\ c_1\ c_2\ (q_0 = 1)\ (q_1 = c_0) \quad (7)$$

$(\texttt{ccx}\ c_1\ q_0\ q_1)$

$$c_0\ c_1\ c_2\ (q_0 = 1)\ (q_1 = c_0 \oplus c_1) \quad (8)$$

$(\texttt{cx}\ q_1\ q_0)$

$$c_0\ c_1\ c_2\ (q_0 = 1 \oplus c_0 \oplus c_1)\ (q_1 = c_0 \oplus c_1) \quad (9)$$

$(\texttt{cx}\ c_1\ q_1)$

$$c_0\ c_1\ c_2\ (q_0 = 1 \oplus c_0 \oplus c_1)\ (q_1 = c_0) \quad (10)$$

$(\texttt{cx}\ c_2\ q_1)$

$$c_0\ c_1\ c_2\ (q_0 = 1 \oplus c_0 \oplus c_1)\ (q_1 = c_0 \oplus c_2) \quad (11)$$

Figure 1: Retrodictive execution of the circuit in Fig. 2

measurement of $q_1 q_0$ would essentially cause a *phase kickback* collapsing the equal superposition of all possible values of c_2, c_1, c_0 to just the two possible values that are consistent with the measurement.

Fig. 3 shows a larger circuit. The top five wires are the interesting inputs while the bottom five wires serve as ancilla bits initialized to fixed values. The question we are interested in this case is the following: say we learn that at the end of the execution we have $b_2 b_1 b_0 = 001$, what possible input values for $a_1 a_0$ and $b_2 b_1 b_0$ could have produced such a result? Using the same process as above, we calculate $b_0 = 1 \oplus a_0 \oplus a_1$, $b_1 = a_1 \oplus a_0 a_1$, and $b_2 = 0$.

Quantum algorithms typically operate on a *black box* holding a classical function whose properties need to be computed. The general structure of these algorithms is to (i) create a superposition of values to be passed as inputs to the black box, (ii) apply the operation inside the black box, and (iii) post-process the output of the black box. We observe that, in quite a few cases, steps (i) and (iii) are actually unnecessary and that the entire "quantum" algorithm can be executed by forward or backward, full or partial, efficient classical *symbolic execution* of the black box.

typical use: superposition, Uf, measure second register; we only care about which x has f(x) = r

3 Retrodictive Execution: Taking advantage of the future

We need to explain ideas about time-reversal, prediction and retrodiction in physics. The laws of computation and the laws of physics are intimately related. When does knowing something about the future help us unveil the structure or symmetries of the past? It is like a detective story, but one with ramifications in complexity and/or efficiency. Problems involving questions where answers demand a Many(past)-to-one(future) map are at the root of our proposal.... Difference between exploiting or not entanglement in the unitary evolution.

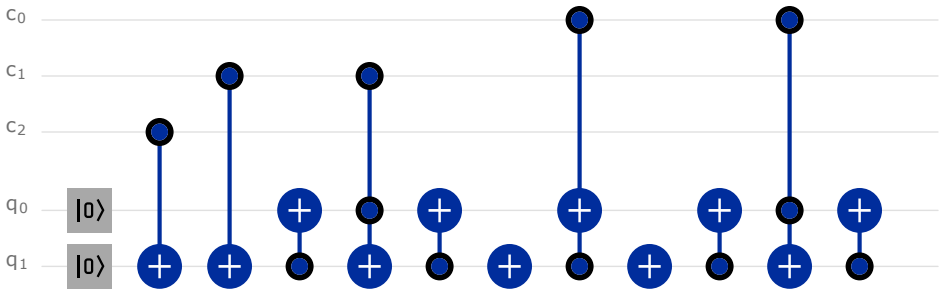


Figure 2: A small example circuit

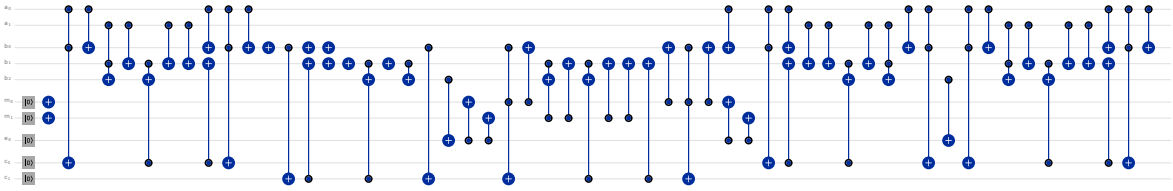


Figure 3: A larger example circuit

4 Partial Symbolic Evaluation with Algebraic Normal Form (ANF)

We should use two prototypical examples to illustrate main ideas before going to the complex ones. The examples I have in mind are: Deutsch-Josza and Simon (precursor of Shor's). There are prior works on de-quantization of the first problem and should make contact with their resolution. Perhaps we can show that they are as efficient classically? That would justify retrodiction alone. The more complex (and important) case of factorization should be the natural follow up.

The idea of symbolic execution is not tied to forward or backward execution. We should introduce it in a way that is independent of the direction of execution. What the idea depends on however is that the wave function, at least in the cases we are considering, can be represented as equations over booleans.

Wave Functions as Equations over Booleans

in the typical scenario for using quantum oracles, we can represent wave function as equations over booleans; equations represent the wave function but the solution is unobservable just like the components of the superposition in the wave function are not observable; just like we don't directly get access to the components of the wave function; we don't directly get access to the solution of the equations; need to "observe" the equations

we can go backwards with an equation (representing a wave function sigma x where f(x) = r and go back towards the present to calculate the wave function (represented as equations again)

Musing: how to explain complementarity when wave function is represented as an equation? Kochen specker;

or contextuality

observer 1 measures wires a,b; obs2 measures wires b,c; not commuting; each obs gives partial solution to equations; but partial solutions cannot lead to a global solution

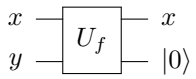
KS suggests that equations do not have unique solutions; only materialize when you measure;

can associate a probability with each variable in a equation: look at all solutions and see the contribution of each variable to these solutions.

5 Classical Algorithms from Quantum Circuits

5.1 Deutsch

The problem is to determine if a function $\mathbb{B} \to \mathbb{B}$ is constant or balanced. Here is the circuit for the Deutsch algorithm after removing all the quantum gates:

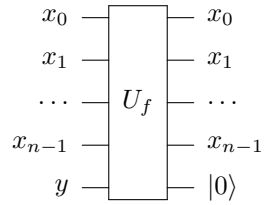


(1)

We fix the ancillary output to a possible boundary condition, say $|0\rangle$, and perform a retrodictive execution of the circuit. This execution produces a formula for y that depends on the function f in the black box. When the function f is a constant function, the formula is the corresponding constant 0 or 1. When the function is balanced the resulting formula is x (when the function is the identity) or $1+x$ (when the function is boolean negation).

5.2 Deutsch-Jozsa

The problem is a generalization of the previous one: the question is to determine if a function $\mathbb{B}^n \to \mathbb{B}$ is constant or balanced. The circuit is identical to above except that x is now a collection of qubits:



(2)

Again, we fix the ancillary output to a possible boundary condition, say $|0\rangle$, and perform a retrodictive execution of the circuit. This execution produces a formula for y that depends on the function f in the black box. When the function f is a constant function, the formula is the corresponding constant 0 or 1. When the function is balanced the resulting formula involves at least one variable x_i.

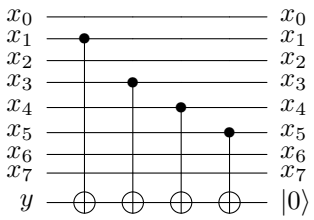
5.3 Bernstein-Vazirani

We are given a function $f_s : \mathbb{B}^n \to \mathbb{B}$ that hides a string $s \in \mathbb{B}^n$. We are promised the function is defined as:

$$f_s(x) = \sum_{i=0}^{n-1} s_i x_i \qquad \bmod 2$$

and the goal is to determine s.

The circuit below demonstrates the situation when $n = 8$ and the hidden string $s = 00111010$:



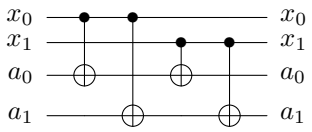
(3)

Symbolically running the circuit in a retrodictive fashion reveals that $y = x_1 \oplus x_3 \oplus x_4 \oplus x_5$ which are exactly the bits that are equal to 1 in the hidden string.

5.4 Simon

We are given a 2-1 function $f : \mathbb{B}^n \to \mathbb{B}^n$ where there exists an a such $f(x) = f(x \oplus a)$ for all x; the goal is to determine a.

The circuit below demonstrates the situation when $n = 2$ and $a = 3$.



(4)

The circuit implements the black box $U_f(x, a) = (x, f(x) \oplus a)$. We first pick a random x, say $x = 3$, fix the initial condition $a = 0$ and run the circuit forward. This execution produces, in the second register, the value of $f(x) = 0$. We now run a symbolic retrodictive execution with $a = 0$ at the output site. That execution produces information on all values of a that are consistent with the observed result. In this case, we get: $a_0 = x_0 + x_1$ and $a_1 = x_0 + x_1$. In other words, when $x_0 = x_1$, we have $a = 0$, and when $x_0 \neq x_1$, we have $a = 3$ which is indeed the desired hidden value.

6 Complexity Analysis

one pass over circuit BUT complexity of normalizing to ANF not trivial; be careful

7 Conclusion