

# QUANTUM COMPUTATION AS A RECURSION

KEITH KENEMER

ABSTRACT. Quantum computation can be recast as a recursion by identifying multi-variate recursive functions which reconstruct the quantum state amplitude tensor at a computation time step from the previous time step and which adaptively compress the information associated with the quantum state. This proposed method is distinct from Tensor Network approaches which assume some factorization based on the physical structure of a problem. Casting the quantum computation in this way highlights where computational intractability arises in the computation by identifying where the recursion complexity becomes intractable and so provides some insight into the boundary between classical and quantum computation. Understanding this boundary is useful both theoretically as well as for practical applications including optimizing quantum compilers. We utilize the (Clifford +T) universal gate set and see how broad classes of quantum circuits can be efficiently simulated due to a symmetry which arises in the structure of the computation and identify additional classes of circuits outside the Clifford group which can be efficiently simulated.

## 1. INTRODUCTION

The nature of quantum computing is subtle and the boundary between classical and quantum computing is an interesting area of research for both the theoretical aspects as well as practical applications such as optimizing quantum compilers. As shown by the Gottesmann-Knill theorem, there are restricted classes of states which exhibit superposition and high degrees of entanglement but which are nevertheless able to be efficiently simulated on classical machines. These classes of states are generated by the Clifford gates: H (Hadamard), S (Phase gate), and CNOT gates. Universal quantum computation becomes possible by simply adding the T gate, i.e. using the Clifford+T gate set, which is well-known universal gate set.

An alternative method for representing the state of a quantum computation is presented which uses a recursion in place of the standard method of computing updates from unitary transformations of the state vector. The advantage of representing the computation in this way is that it highlights where and how the intractability of the computation arises with respect to classical simulation. The recursive functions construct the amplitude tensor at time step  $n$  from the amplitude tensor at time step  $n - 1$  and can be considered to be a compression of the state vector. At each stage of the computation, a new amplitude state is obtained by a gate-dependent transformation of the previous one.

## 2. QUANTUM COMPUTATION AS A RECURSION

In this section we will consider the Clifford+T gate set and develop a set of recursive functions which represent the amplitudes. These functions will then be used to explore classes of circuits which can be simulated classically. In each of the following sections we examine the effect of applying various gates on the k-th qubit to see how the amplitude tensor is transformed.

**2.1. S,T Gate Recursion.** The S and T gates introduce a conditional phase factor based on the qubit state i.e.  $|0\rangle \rightarrow |0\rangle$  and  $|1\rangle \rightarrow e^{i\theta}|1\rangle$ . Decomposing the basis into the constituent bitfields, the initial state of the computation is:

$$\psi = \sum_m \alpha_m |\phi_m\rangle = \sum_b \alpha(b) \bigotimes_q |b_q\rangle \quad (1)$$

where  $b$  is a multi-dimensional qubit index and  $\alpha$  is the amplitude tensor. The sum can be partitioned into groups based on whether the state of the k-th qubit in the basis is  $|0\rangle$  or  $|1\rangle$ :

$$\begin{aligned} \psi = & \sum_{b \in B_0} \alpha(b) |b_k=0\rangle \bigotimes_{h>k} |b_h\rangle \otimes |0\rangle \otimes \bigotimes_{l<k} |b_l\rangle \\ & + \sum_{b \in B_1} \alpha(b) |b_k=1\rangle \bigotimes_{h>k} |b_h\rangle \otimes |1\rangle \otimes \bigotimes_{l<k} |b_l\rangle \end{aligned} \quad (2)$$

Applying the Phase (S) or T gate results in a phase factor showing up on basis states where the k-th qubit is in the state  $|1\rangle$ :

$$\begin{aligned} \psi = & \sum_{b \in B_0} \alpha(b) |b_k=0\rangle \bigotimes_{h>k} |b_h\rangle \otimes |0\rangle \otimes \bigotimes_{l<k} |b_l\rangle \\ & + \sum_{b \in B_1} e^{i\theta} \alpha(b) |b_k=1\rangle \bigotimes_{h>k} |b_h\rangle \otimes |1\rangle \otimes \bigotimes_{l<k} |b_l\rangle \end{aligned} \quad (3)$$

The result can be expressed in the original full set of indices, with a bitfield-dependent phase factor:

$$\psi = \sum_b e^{i\theta b_k} \alpha(b) \bigotimes_q |b_q\rangle \quad (4)$$

Comparing Eq (1) and Eq (4), we see that the effect of the S (or T) gate was to introduce a phase ( $\theta = \pi/2$  for S gate,  $\theta = \pi/4$  for T gate) in the amplitude function which is dependent on the k-th qubit value.

$$\alpha_{n+1}(b) = e^{i\theta b_k} \alpha_n(b) \quad (5)$$

**2.2. Hadamard Gate Recursion.** After decomposing the basis state into bitfields as in Eq (2), we apply the Hadamard operation which generates superpositions, mapping  $|0\rangle \rightarrow \frac{|0\rangle+|1\rangle}{\sqrt{2}}$  and  $|1\rangle \rightarrow \frac{|0\rangle-|1\rangle}{\sqrt{2}}$ .

$$\begin{aligned}
\psi = & \sum_{b \in B_0} \alpha(b) |b_k=0\rangle \bigotimes_{h>k} |b_h\rangle \otimes \frac{|0\rangle + |1\rangle}{\sqrt{2}} \otimes \bigotimes_{l<k} |b_l\rangle \\
& + \sum_{b \in B_1} \alpha(b) |b_k=1\rangle \bigotimes_{h>k} |b_h\rangle \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}} \otimes \bigotimes_{l<k} |b_l\rangle
\end{aligned} \tag{6}$$

Grouping terms which have  $|0\rangle$  in the  $k$ -th qubit and grouping terms with  $|1\rangle$  in the  $k$ -th qubit, the resulting amplitude tensor now shows two terms:

$$\begin{aligned}
\psi = & \sum_{b \in B'_0} \frac{\alpha(b)|b_k=0\rangle + \alpha(b)|b_k=1\rangle}{\sqrt{2}} \bigotimes_{h>k} |b_h\rangle \otimes |0\rangle \otimes \bigotimes_{l<k} |b_l\rangle \\
& + \sum_{b \in B'_1} \frac{\alpha(b)|b_k=0\rangle - \alpha(b)|b_k=1\rangle}{\sqrt{2}} \bigotimes_{h>k} |b_h\rangle \otimes |1\rangle \otimes \bigotimes_{l<k} |b_l\rangle
\end{aligned} \tag{7}$$

Rewriting in terms of the original full set of indices, we can see the amplitudes have been transformed:

$$\psi = \sum_b \frac{\alpha(b)|b_k=0\rangle + (-1)^{b_k} \alpha(b)|b_k=1\rangle}{\sqrt{2}} \bigotimes_q |b_q\rangle \tag{8}$$

Comparing Eq (1) and Eq (8), the effect of the Hadamard gate is to create two terms, where the amplitudes at the current level in the recursion requires two calls to the previous level in the general case (Eq(9)). This is important from a computational standpoint since it means the computation can become intractable depending on the details of the circuit topology.

$$\alpha_{n+1}(b) = \frac{\alpha_n(b)|b_k=0\rangle + (-1)^{b_k} \alpha_n(b)|b_k=1\rangle}{\sqrt{2}} \tag{9}$$

The following table summarizes the Clifford+T gate operations in terms of the transformations the gate operations induce on the amplitude tensor at the previous step in the computation.

Hadamard Gate	$\alpha_{n+1}(b) = \frac{1}{\sqrt{2}}(\alpha_n(b) b_k=0\rangle + (-1)^{b_k} \alpha_n(b) b_k=1\rangle)$
Phase (S) Gate	$\alpha_{n+1}(b) = e^{\frac{i\pi}{2} b_k} \alpha_n(b)$
$\frac{\pi}{8}$ (T) Gate	$\alpha_{n+1}(b) = e^{\frac{i\pi}{4} b_k} \alpha_n(b)$
CNOT Gate	$\alpha_{n+1}(b) = \alpha_n(b) b_t=b_t \oplus b_c$

TABLE 1. Quantum Logic Operations and Associated Amplitude Recursion

From the table, we can make several observations. First, the S,T, and CNOT operations result in a low complexity increase for each operation, adding only a phase factor in the case of S and T gates and simple variable transformation in the case of CNOT operations. We could expect that S-T-CNOT networks should be amenable to efficient classical simulation. The Hadamard operation, however, includes two terms at each level in the recursion. In the general case, this can result in a binary tree with exponential computational complexity leading to intractability. The Hadamard operation therefore will be a central focus and the exploration

of the boundary between classical and quantum computing will be formulated as discovering the class of inputs and operations which have invariance properties such that the Hadamard recursion does not require both the  $b_k = 0$  and  $b_k = 1$  branches of the recursion.

### 3. REDUNDANCY IN CLIFFORD CIRCUITS

In this section we will show some of the properties of Clifford circuits derived from the proposed recursion framework. These properties will be used to identify computational redundancies which makes this class of circuits amenable to efficient classical simulation.

**Definition 3.1.** A composition of recursive functions  $f_{N-1} \circ f_{N-2} \dots \circ f_n \dots \circ f_1 \circ f_0$  is compressible if the composite function at each level of the recursion  $f_n$  can be represented by a single branch (call) to the previous level  $f_{n-1}$ .

**Lemma 3.2.** *The amplitude function  $\alpha(b)$  for a Clifford circuit  $C$  constructed from an initial uniform superposition (Haar distribution) followed by an S-CNOT network will be of the following product form:  $P(b) = \alpha_0 \prod_{n=0}^{N-1} e^{i\theta \oplus_{n,q} b_{n,q}}$  where  $b$  is a set of boolean variables which specify a particular basis state  $b = b_{N-1}, b_{N-2}, \dots, b_k, \dots, b_1, b_0$  and  $b_i \in \{0, 1\}$ .*

*Proof.* The initial Haar state is given by a constant  $\alpha_0$  since it is a uniform superposition. Subsequent S gates will produce phase factors of the form  $e^{i\theta b_k}$  where  $b_k$  corresponds to the qubit being operated on, resulting in products of the form  $P(b) = \alpha_0 \prod_{n=0}^{N-1} e^{i\theta b_n}$ . The effect of CNOT gates will be to transform the target qubits according to  $b_t \rightarrow b_c \oplus b_t$  creating XOR chains within some of the products and resulting in the final form  $P(b) = \alpha_0 \prod_{n=0}^{N-1} e^{i\theta \oplus_{n,q} b_{n,q}}$ .  $\square$

Next, we will examine this product form and identify symmetries which will result in the compressibility of the computation which will ultimately allow for efficient classical simulation of circuits constructed from the Clifford group gates H, S, and CNOT.

**Lemma 3.3.** *The amplitude function  $\alpha(b)$  for a Clifford circuit  $C$  constructed from an initial uniform superposition (Haar distribution) followed by an S-CNOT network exhibits the following symmetry  $\alpha(b)|_{b_k=1} = e^{i\theta N} \overline{\alpha(b)}|_{b_k=0}$  and is compressible in the sense of Definition 1.*

*Proof.* Starting from the product form in Lemma 3.2 (and suppressing  $\alpha_0$  for clarity) we will see how the product form is transformed when the  $k$ -th qubit evaluated at  $b_k = 0$  and again at  $b_k = 1$  and assume (for the moment) that our bitfield-of-interest  $b_k$  is present in all XOR sequences in the phase factors:

$$P(b)|_{b_k=0} = \prod_{n=0}^{N-1} e^{i\theta \oplus_{q \neq k} b_q \oplus 0} = \prod_{n=0}^{N-1} e^{i\theta \oplus_{q \neq k} b_q} \quad (10)$$

$$P(b)|_{b_k=1} = \prod_{n=0}^{N-1} e^{i\theta \oplus_{q \neq k} b_q \oplus 1} = \prod_{n=0}^{N-1} e^{i\theta(1 - \oplus_{q \neq k} b_q)} \quad (11)$$

and so we see there is a type of *complement invariance* whereby the product terms evaluated at  $b_k = 1$  can be inferred from the product terms evaluated at the logical complement  $b_k = 0$  resulting in compressibility of the function:

$$P(b)|_{b_k=1} = (e^{i\theta})^N \prod_{n=0}^{N-1} e^{-i\theta(\oplus_{q \neq k} b_q)} = e^{i\theta N \overline{P(b, \theta)}}|_{b_k=0} \quad (12)$$

The analysis presented above has assumed that a given bitfield-of-interest  $b_k$  was present in all phase factors for clarity, but the results remain valid for a general S-CNOT network which includes factors without  $b_k$  such as  $f(b, \theta) = \alpha(b)\phi(b', \theta)$  such that  $b_k \notin b'$ . In this case  $\phi(b', \theta)$  is effectively a constant with respect to  $b_k$  and the invariance would manifest as  $f(b, \theta)|_{b_k=1} = (e^{i\theta})^N \overline{\alpha(b)}|_{b_k=0} \cdot \phi(b', \theta)$ .

The effect of an S-CNOT network operating on the previous stage of computation is to multiply it by one of the product forms described above. Additionally, the CNOT operation(s) also globally performs a variable substitution whereby  $b_t \rightarrow b_c \oplus b_t$ . Therefore an S-CNOT network  $f(b, \theta)$  acting on the previous state  $\alpha_{n-1}(b)$  will produce the new state  $\alpha_n(b) = f(b, \theta) \cdot \alpha'_{n-1}(b)$  where  $\alpha'_{n-1}(b)$  has been modified such that  $b_t$  variables have been extended to  $b_c \oplus b_t$ . The state  $\alpha'_{n-1}(b)$  will be of the same general form as  $\alpha_{n-1}(b)$  (i.e. product, sum of products, identical  $\theta$ , etc) and so the invariance properties will also remain the same.  $\square$

Composition properties can be derived for tensors (or functions) which possess the invariance symmetry from Lemma 3.3.

**Lemma 3.4.** *If  $F(b) = f(b)g(b)$  is a composite function composed from a product of functions  $f(b)$  and  $g(b)$  which both exhibit the complement invariance property from Lemma 3.3 then the product  $F(b)$  will also exhibit this invariance.*

*Proof.*  $F(b)|_{b_k=1} = f(b)|_{b_k=1}g(b)|_{b_k=1}$  and  $f(b)|_{b_k=1} = e^{i\theta N \overline{f(b)}}|_{b_k=0}$  and  $g(b)|_{b_k=1} = e^{i\theta N' \overline{g(b)}}|_{b_k=0}$  therefore  $F(b)|_{b_k=1} = e^{i\theta(N+N') \overline{F(b)}}|_{b_k=0}$  which requires only the  $b_k = 0$  branch of the recursion and so is compressible.  $\square$

Using these preliminary observations, we can outline why Clifford circuits remain computationally tractable using the recursion framework. Consider a quantum circuit with a structure defined by alternating S-CNOT networks and Hadamard networks and with an initial state in uniform superposition. Assuming this initial state is convenient when using the recursion framework and there is no loss of generality since any arbitrary initial state could be created from this using Hadamard gates. From Lemma 3.2 we know that the state of circuit after the first S-CNOT network will be of the form  $P_N(b) = \alpha_0 \prod_{n=0}^{N-1} e^{i\theta \oplus_{n,q} b_{n,q}}$ . If we subsequently apply a Hadamard gate on the  $k$ -th qubit, the state will be of the form  $Q_N(b') = P_N(b)|_{b_k=0} + (-1)^{b_k} e^{i\theta N} \overline{P_N(b)}|_{b_k=0}$ . We observe that  $Q_N(b')|_{b_l=1} = e^{i\theta N} \overline{Q_N(b')|_{b_l=0}}$  for some arbitrary  $b_l$  and so although  $Q_N(b')$  is a weighted sum of product terms  $P_N(b)$  and  $\overline{P_N(b)}$ , it behaves as if it were a product form itself from Lemma 3.2 and exhibits the complement invariance property. Applying another S-CNOT network  $P_M(b)$  would result in a state of the  $P_M(b)Q_N(b')$  which will also be complement invariant by Lemma 3.4 and so by induction we see that all further applications of S-CNOT and Hadamard networks will only produce functions which

are complement invariant, requiring only calls to one of the Hadamard branches and so remain tractable for classical simulation.

#### 4. CONCLUSION

A new perspective on quantum computing was presented which casts the computation as a recursion in order to highlight how and where the intractability of quantum computing arises. Using the recursion framework it can be verified that Clifford (H,S,CNOT) circuits are amenable to efficient classical simulation, consistent with the Gottesmann-Knill theorem. Additionally, and very importantly, the specific value of  $\theta$  which distinguishes the S gate from the T gate was not important in the recursion framework, which implies that we can now extend to new classes of circuits which can be efficiently simulated, namely H-T-CNOT circuits and H- $R_\theta$ -CNOT circuits where  $R_\theta$  is a phase shifter gate and  $\theta$  is arbitrary but fixed. Future work may include further investigation into gate sequences with mixed  $\theta$  to see if we can further expand the classes of circuits which can be efficiently simulated classically.

#### 5. REFERENCES

1. “Heisenberg Representation of Quantum Computers”, Gottesman, 1998
2. “Improved Simulation of Stabilizer Circuits”, Aaronson, 2008
3. “Classical simulation of quantum computation, the Gottesman-Knill theorem, and slightly beyond”, Maarten Van den Nest, 2009
4. Quantum Computation and Quantum Information, Nielsen, Chuang, 2000

## APPENDIX A. THE UNIVERSAL GATE SET: H,S,T,CNOT

The Hadamard (H)gate, phase (S) gate, controlled-NOT gate, and  $\frac{\pi}{8}$  (T) gate are universal for quantum computation. The Hadamard gate is defined by the following single-qubit operation which transforms the qubit into a superposition with a phase that depends on the state of the original qubit:

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad |b\rangle \rightarrow \frac{1}{\sqrt{2}}(|0\rangle + (-1)^b |1\rangle) \quad (13)$$

The Phase gate is defined by the following single-qubit operation which generates a phase factor dependent on the original qubit state:

$$S = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix} \quad |b\rangle \rightarrow e^{\frac{i\pi}{2}b} |b\rangle \quad (14)$$

Similarly, the  $\frac{\pi}{8}$ -gate is defined by the single-qubit operation which also generates a phase factor dependent on the original qubit state:

$$T = \begin{bmatrix} 1 & 0 \\ 0 & e^{\frac{i\pi}{4}} \end{bmatrix} \quad |b\rangle \rightarrow e^{\frac{i\pi}{4}b} |b\rangle \quad (15)$$

The CNOT (controlled-not) gate performs a conditional negation of the target qubit based on the value of the control qubit. The operation preserves the state of the control qubit and replaces the target qubit with the XOR of the control and target qubits.

$$CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad |...b_c...b_t...\rangle \rightarrow |...b_c...b_c \oplus b_t...\rangle \quad (16)$$

where the bitfields  $b \in \{0, 1\}$  and Dirac notation is used for the basis states:

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad (17)$$

Another relevant gate is the phase shift gate  $R_\phi$  which is a generalization of the T and S gates and has the following matrix operation:

$$R_\phi = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\phi} \end{bmatrix} \quad |b\rangle \rightarrow e^{i\phi b} |b\rangle \quad (18)$$