Puente entre la Lógica Clásica y la Computación Cuántica a través del Razonamiento Contradictorio

Kevin Davila (Universidad de los Andes) <analyst1@tradingtech.ai>

Abstract

Este artículo ejemplifica el uso de un novedoso marco semántico para la lógica de primer orden, inspirado en la mecánica cuántica y diseñado para captar algunas sutilezas de la computación cuántica. Demostramos que, para procedimientos cuánticos clave —en concreto el algoritmo de Deutsch y la descomposición de la compuerta Toffoli en compuertas de 2 bits— existen fórmulas que, a pesar de ser inconsistentes en términos clásicos, describen parcialmente el comportamiento algorítmico y exhiben características únicas de consistencia no clásica. Además, mostramos brevemente cómo el abordaje ilustrado en este artículo podría ampliarse para abordar otros algoritmos clave, como el algoritmo de Deutsch–Jozsa y el algoritmo de Estimación de Fase. Una implicación de nuestros hallazgos es la posibilidad de que aceptar contradicciones pueda ser útil para la innovación en el diseño de algoritmos cuánticos.

Palabras Clave: Computación Cuántica, Lógica de Primer Orden, Semántica Cuántica, Lógicas No Clásicas, Tolerancia a la Contradicción

Bridging Classical Logic and Quantum Computation Through Contradictory Reasoning

Abstract

This paper exemplifies the use of a novel semantic framework for first-order logic, informed by quantum mechanics and designed to capture some of the nuances of quantum computation. We demonstrate that, for key quantum procedures—concretely the Deutsch algorithm and the 2-gate decomposition of the Toffoli gate—there exist formulas that, despite being inconsistent in classical terms, partially describe algorithmic behaviour and exhibit unique non-classical consistency features. Moreover, we briefly discuss how the approach illustrated in this article would need to be elaborated to deal with other key algorithms such as the Deutsch–Jozsa algorithm and the Phase Estimation algorithm.
An implication of our findings is the possibility that embracing contradictions might be useful for innovation in quantum algorithm design.

Keywords: Quantum Computation, First-Order Logic, Quantum Semantics, Non-Classical Logics, Contradiction Tolerance

1 Introduction

Quantum computation has transformed our understanding of what can be *efficiently* computed, revealing new complexity-theoretic landscapes and inspiring novel logical frameworks (Deutsch, 1985; Shor, 1997). At the same time, descriptive complexity has shown that many computational problems can be phrased as *first-order* (FO) sentences evaluated over finite structures (Libkin, 2004). Yet classical semantics treats an FO sentence as either true or false, leaving no room for the probabilistic behaviour that characterises quantum devices. This paper addresses the following question: *Can we reinterpret FO sentences so that their truth values emerge from—and are witnessed by—quantum circuits?* Recent surveys further motivate this direction (Bouland and Ozols, 2019; Selinger, 2015).

Building on Deutsch's two-qubit algorithm, we introduce the notion of a *quasi-consistent* sentence: an FO sentence Φ for which the circuit-based evaluation outputs 1 with non-trivial probability $0 < p < 1$. Our approach differs from earlier logical encodings of quantum protocols (Abramsky and Coecke, 2004; van Benthem, 2014) by providing an explicit gate-level translation of quantifiers, instantiation, and relation checking, and exhibiting concrete circuits. Besides clarifying the logical content of quantum subroutines, the framework lays groundwork for analysing algorithms such as Simon's and Grover's (Simon, 1994; Grover, 1996) from a proof-theoretic viewpoint.

The remainder of the article is organised as follows. Section 2 presents a detailed first-order encoding of Deutsch's problem and derives the sentence Φ. The next section, *Semantic Framework and Circuit Verification*, develops the gate-level semantics, introduces the interpreter tuple $(\texttt{Ins}, \texttt{Rec}_{0,1}, \texttt{Rec}_{+,-}, \rho)$, and proves the quasi-consistency of Φ. The section titled *Two-Gate Decomposition of the Toffoli Gate* extends the methodology to the reversible three-qubit Toffoli operation through its decomposition into 2-gates. Finally, Section 5 summarises the main contributions and outlines future work, including applications to the Deutsch–Jozsa and Phase-Estimation algorithms.

2 Deutsch's problem: logical encoding

Deutsch's algorithm solves what we call the *Deutsch Problem*: given a Boolean function defined over a two-element domain, determine—using a single quantum query—whether the function is *constant* or *balanced* (Deutsch, 1985; Deutsch and Jozsa, 1992). By harnessing interference, the algorithm effectively evaluates two counterfactual scenarios simultaneously (Nielsen and Chuang, 2010). We encode the *problem*—rather than the step-by-step workings of the algorithm—in a first-order language with a single binary relation R and represent its core via a four-conjunct first-order sentence

$$\Phi = \Phi_0 \wedge \Phi_1 \wedge \Phi_2 \wedge \Phi_3.$$

We argue that, in a sense, this encoding captures both the classical data and the counterfactual content, analogous to the quantum superposition underlying the algorithm.

2.1 Logical Encoding of the Deutsch Problem

We represent the problem by the following subformulas:

$$\begin{aligned}
\Phi_0 :& \quad \forall x\, \exists y\, [R(x,y) \wedge \forall y'\, (R(x,y') \rightarrow y = y')],\\
\Phi_1 :& \quad \exists x\, \exists y\, [x \neq y \wedge \forall z\, (z = x \vee z = y)],\\
\Phi_2 :& \quad \exists x\, \exists y\, [x \neq y \wedge \exists x'\, \exists y'\, (R(x,x') \wedge R(y,y') \wedge x' = y')],\\
\Phi_3 :& \quad \exists x\, \exists y\, [x \neq y \wedge \exists x'\, \exists y'\, (R(x,x') \wedge R(y,y') \wedge x' \neq y')].
\end{aligned}$$

Here, Φ_0 guarantees that R is a well-defined function (i.e. works as a mapping), Φ_1 restricts the domain to exactly two elements. In conjunction with Φ_1, Φ_2 and Φ_3 encode, respectively, the property of R being constant or balanced. Although Φ_2 and Φ_3 are classically mutually exclusive, their coexistence in Φ reflects quantum superposition—supporting the possibility that contradictory properties coexist, as highlighted by the logical models of Rawling and Selesnick (2000).

We now leverage Deutsch's algorithm and the sentence Φ to showcase the semantic framework introduced above. The guiding principle is *interpretation via circuits*: for any first-order sentence such as Φ we build a quantum circuit whose acceptance statistics encode its truth value. We call the sentence *quasi-consistent* when, under the uniform distribution on inputs, the circuit returns 1 with probability p satisfying $0 < p < 1$.

The sentence Φ defined above is indeed quasi-consistent. We describe next the construction of the circuit for Φ_0. Similar designs are used for subformulas Φ_2 and Φ_3.

3 Semantic Framework and Circuit Verification

In our game–theoretic reading of first-order semantics there are two implicit agents: *Player I*, who assigns values to universally quantified variables, and *Player II*, who must supply witnesses for existentially quantified variables. The partial function ρ introduced below encodes a choice strategy for Player II. The map `Ins` (*insert*) maps classical elements to qubit states,, while the recovery maps $\texttt{Rec}_{0,1}$ and $\texttt{Rec}_{+,-}$ (*recover*) perform the inverse translation by measuring in the computational or Hadamard bases and returning the corresponding classical symbol.

3.1 Semantic Framework

The circuit for Φ operates within a semantic framework consisting of:

1. **A Classical Structure:** We work over the domain $\{a, b, c, d\}$ with binary relation
$$R = \{(a, b), (b, a), (c, d), (d, c)\}.$$
The elements a, b, c, d are identified with quantum states $|a\rangle, |b\rangle, |c\rangle, |d\rangle$ (assumed to be orthonormal).

2. **Mapping Functions:** A tuple of functions connects the classical structure to quantum states:

$$(\texttt{Ins}, \texttt{Rec}_{0,1}, \texttt{Rec}_{+,-}, \rho).$$

Their action is defined by

$$\texttt{Ins}(a) = |0\rangle\,, \quad \texttt{Ins}(b) = H\,|0\rangle\,, \quad \texttt{Ins}(c) = |1\rangle\,, \quad \texttt{Ins}(d) = H\,|1\rangle\,,$$

$$\texttt{Rec}_{0,1}\,|0\rangle = a, \quad \texttt{Rec}_{0,1}\,|1\rangle = c,$$

$$\texttt{Rec}_{+,-}\,|0\rangle = b, \quad \texttt{Rec}_{+,-}\,|1\rangle = d,$$

ρ is a function capturing the second player strategy to choose witnesses for existentially quantified variables given values for the previous universally quantified variables. In this case we define ρ by

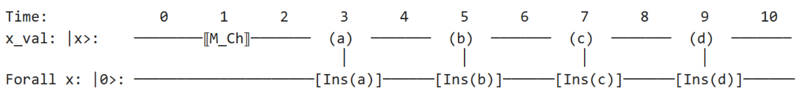
$$\rho(a) = b, \quad \rho(b) = a, \quad \rho(c) = d, \quad \rho(d) = c.$$

3.2 Circuit Design for Φ_0

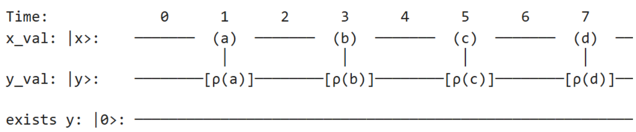
Circuit–diagram conventions. Time flows from left to right, with steps marked along the top (0, 1, 2, …) for each circuit or sub-circuit diagram. Each horizontal line represents a register, labeled on the left. Square brackets indicate operations: $[U]$ applies a unitary U, and $[[M]]$ performs a measurement M. A solid dot "•" connected to $[U]$ denotes a control on $|1\rangle$, i.e., U is applied only if the control qubit is in state $|1\rangle$. For d-dimensional registers, the label "(a)" indicates that U is applied only when the control register is in the basis state $|a\rangle$; otherwise, the identity is applied. Dashed rectangles enclose multi-wire subcircuits, typically used for predicate tests such as $R(x, y)$.

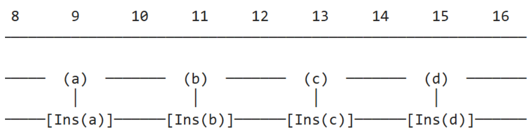
The quantum circuit for Φ_0 proceeds in several stages:

1. **Initialization:** For the universally quantified variable x, a uniform superposition is created over $\{a, b, c, d\}$ and a measurement is performed with respect to the basis $\{a, b, c, d\}$ (mimicking a random choice). The controlled application of `Ins` then maps the chosen classical element into its corresponding quantum state.

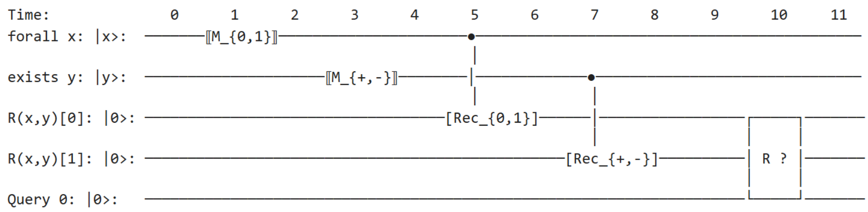


2. **Witness Generation:** Controlled operations employing the strategy ρ instantiate a witness for the existential variable y. Then a corresponding quantum state is prepared using `Ins`.





3. **Verification of $R(x,y)$:** The states are measured: x is measured in the $\{|0\rangle, |1\rangle\}$ basis and y in the $\{|+\rangle, |-\rangle\}$ basis. The functions $\texttt{Rec}_{0,1}$ and $\texttt{Rec}_{+,-}$ recover the classical values, and a three-wire gate then verifies if the pair (x,y) belongs to R.



4. **Secondary Check:** A similar process of universal instantiation and relation checking is applied for the variable y' to enforce the uniqueness condition from Φ_0.

5. **Truth Value Computation:** After designing the modules for the circuit dealing with quantifiers and basic formulae, the propositional connectors (like $\wedge$ and $\rightarrow$) can be handled using classical gates (implemented in the quantum circuit).

The output cable of the overall circuit is the one corresponding to the output of the last propositional connective that was applied (in our case the conjunction). After completing the circuit and computing the probability $\mathbb{P}_i$ that the measured truth value of Φ_i will be 1, it can be seen that for $\mathbb{P}$ (the analogous probability for Φ) one has

$$0 < \mathbb{P} = \prod_{i=0}^{3} \mathbb{P}_i < 1.$$

Thus, Φ is a *proper quasi-consistent sentence*: classically inconsistent yet accepted with strictly positive probability by its witnessing quantum circuit, hinting at a broader landscape of first-order statements whose quantum verifiability transcends classical satisfiability.

4 Two-Gate Decomposition of the Toffoli Gate

A hallmark of quantum computation is the decomposition of the Toffoli gate into two-qubit operations (Toffoli, 1980; Barenco et al., 1995). In our approach, the possibility of such a decomposition is captured by a single first-order sentence $\Psi = \bigwedge_{i=0}^{4} \Psi_i$, where:

$$\Psi_0 := \bigwedge_{i=0}^{4} \forall x, y \left[x \neq y \wedge \exists z_x, z_y \big(R_i(x, z_x) \wedge R_i(y, z_y) \big) \rightarrow z_x \neq z_y \right].$$

This ensures that for each two-gate operation, distinct inputs produce distinct outputs, a property necessary for reversibility.

$$\Psi_1 := \bigwedge_{i=0}^{4} \forall x \Big(x \in \mathrm{Dom}(R_i) \vee x \in \mathrm{Rang}(R_i) \Big).$$

This guarantees that every element appears in either the domain or the range of each relation R_i, ensuring complete coverage of the state space.

$$\Psi_2 := \exists x_0, \ldots, x_7\ \forall x, y \left\{ \Big(\bigwedge_{k<j} x_k \neq x_j \Big) \wedge \Big(\bigvee_{l=0}^{7} (x = x_l \vee y = x_l) \Big) \wedge \Big(\bigwedge_{i=0}^{4} \big(R_i(x, y) \rightarrow R_i\big(f_i(x), f_i(y)\big) \big) \Big) \right\}.$$

This asserts that eight distinct basis states exist and that, if the relation R_i holds for any pair (x, y) (with at least one element being among these basis states), then the corresponding transformation f_i preserves this relation.

$$\Psi_3 := \exists x_0, \ldots, x_7 \left[\Big(\bigwedge_{k<l<8} x_k \neq x_l \Big) \wedge \Big(\bigwedge_{j=0}^{7} f_4 \circ f_3 \circ f_2 \circ f_1 \circ f_0(x_j) = \mathrm{ToffoliGate}(x_j) \Big) \right].$$

This ensures that the composition of functions $f_0, \ldots, f_4$ reproduces the action of the Toffoli gate on the specified basis states.

$$\Psi_4 := \exists x_0, \ldots, x_7 \left[\Big(\bigwedge_{k<l<8} x_k \neq x_l \Big) \wedge \forall z \Big(\bigvee_{j=0}^{6} (z = x_j) \Big) \right].$$

This fixes the universe precisely to the set of eight basis states, ensuring that all elements under consideration are part of the decomposition.

Although one can directly verify via classical methods that the overall sentence is inconsistent, the techniques described in earlier sections allow us to construct a verifier quantum circuit that demonstrates the quasi-consistency of Ψ.

Outlook to Further Algorithms. The gate-level semantics developed for Deutsch's problem and the Toffoli decomposition extend naturally to more sophisticated routines. In particular, by replacing the two-element domain used in Section 2 with 2^n-element superpositions and introducing modular-arithmetic predicates, one can craft quasi-consistent sentences whose witnessing circuits coincide—up to poly-size overhead—with the Deutsch–Jozsa algorithm for constant-versus-balanced promises and with the Phase Estimation algorithm that underlies order-finding and Shor's factoring scheme.

5 Conclusions

By translating each syntactic construct of first-order logic into a concrete, depth-controlled quantum sub-circuit, we have shown that *truth* in classical semantics can be relaxed to a measurable acceptance probability that we term *quasi-consistency*. This bridge between descriptive complexity and quantum query computation clarifies the logical content of well-known protocols such as Deutsch's algorithm and opens a pathway for analysing richer algorithms through proof-theoretic lenses. Consequently, classical sentences can be evaluated on quantum hardware, and their probabilistic outcomes quantify an intermediate notion of logical satisfiability. We believe this gate-level semantics can become a useful tool for both quantum program verification and the complexity-theoretic classification of logical theories.

Books

Libkin, L. (2004). *Elements of finite model theory.* Springer.

Nielsen, M. A., & Chuang, I. L. (2010). *Quantum computation and quantum information.* Cambridge University Press.

van Benthem, J. (2014). *Logic in games.* MIT Press.

Journal Articles and Proceedings

Abramsky, S., & Coecke, B. (2004). A categorical semantics of quantum protocols. In *Proceedings of the 19th IEEE Symposium on Logic in Computer Science* (pp. 415–425).

Barenco, A., Bennett, C. H., Cleve, R., DiVincenzo, D. P., Margolus, N., Shor, P., Sleator, T., Smolin, J. A., & Weinfurter, H. (1995). Elementary gates for quantum computation. *Physical Review A, 52*(5), 3457–3467.

Bouland, A., & Ozols, M. (2019). Quantum structural complexity: A survey. *SIGACT News, 50*(2), 55–73.

Deutsch, D. (1985). Quantum theory, the Church–Turing principle and the universal quantum computer. *Proceedings of the Royal Society A, 400*, 97–117.

Deutsch, D., & Jozsa, R. (1992). Rapid solution of problems by quantum computation. *Proceedings of the Royal Society A, 439*, 553–558.

Grover, L. K. (1996). A fast quantum mechanical algorithm for database search. In *Proceedings of the 28th Annual ACM Symposium on Theory of Computing* (pp. 212–219).

Rawling, J. P., & Selesnick, S. A. (2000). Orthologic and quantum logic: Models and computational elements. *Journal of the ACM, 47*(4), 721–751.

Selinger, P. (2015). Quantum circuits of T-depth one. In *Proceedings of the 30th Annual ACM/IEEE Symposium on Logic in Computer Science* (pp. 487–496).

Shor, P. W. (1997). Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing, 26*(5), 1484–1509.

Simon, D. R. (1994). On the power of quantum computation. In *Proceedings of the 35th Annual Symposium on Foundations of Computer Science* (pp. 116–123).

Toffoli, T. (1980). Reversible computing. In J. W. de Bakker & J. van Leeuwen (Eds.), *Automata, Languages and Programming* (Vol. 85, pp. 632–644). Springer.