논문 2025-1-3 http://dx.doi.org/10.29056/jsav.2025.03.03

양자 컴퓨팅 소프트웨어 유사도 검증 모델에 대한 연구

안철범*, 김진홍**†

A Study on the Quantum Computing Software Similarity Validation Model

Chulbum Ahn*, Jinhong Kim**†

요 약

오늘날 양자 컴퓨팅은 전통적인 고전 컴퓨터보다 우수한 병렬 처리 능력과 특정 문제에 대해 훨씬 더 높은 계산 효율성이 제공되는 잠재된 기술이다. 하지만, 양자 컴퓨터 소프트웨어의 개발은 아직 초기 단계에 있으며, 이를 검증하고 정확성을 보장하는 것은 중요한 과제 중 하나임은 틀림없다. 본 연구는 양자 컴퓨팅 소프트웨어의 유사성 평가에 대한 모델을 제시하고, 이를 통해 양자 알고리즘과 프로그램의 정확성과 신뢰성을 향상시키는 접근 방식에 대해 향후 연구로 제안하고자 한다. 이를 위해 양자 알고리즘의 동작 분석과 양자 회로와 소프트웨어 및 시스템의 시간적 동작을 위한 Kripke 구조를 사용하여 모델을 생성하고 템포-랄 논리를 사용하여 동작을 지정하고, 크립키 구조(Kripke Structure)는 유한 상태 기계와 유사한 방식으로 상태 간의 전이를 설명함으로써 모델링하고자 한다.

Abstract

Now a day, Quantum computing is a technology with the potential to offer superior parallel processing capabilities and significantly higher computational efficiency for certain problems compared to traditional classical computers. However, the development of quantum computer software is still in its early stages, and verifying it to ensure accuracy has emerged as one of the major challenges. This research presents a model for the similarity evaluation of quantum computing software, which we propose as a future study on approaches to improve the accuracy and reliability of quantum algorithms and programs. To achieve this, the behavior analysis of quantum algorithms, the temporal behavior of quantum circuits and software and systems, creates models using Kripke structures and specifies behavior using tempo-lal logic, and Kripke structures model by describing transitions between states in a manner similar to finite-state machines.

한글키워드 : 양자 컴퓨팅, 병렬 처리, 양자 컴퓨팅 소프트웨어 유사성, 양자 알고리즘, 양자회로

keywords : quantum computing, parallel processing, quantum computing sw similarity, quantum algorithm, quantum circuit

* 서일대학교 정보통신공학과
** 배재대학교 소프트웨어학과
† 교신저자: 김진홍(email: jinhkm@pcu.ac.kr)

접수일자: 2025.02.27. 심사완료: 2025.03.10.
게재확정: 2025.03.20.

1. 서 론

양자 컴퓨팅은 기존의 고전적인 컴퓨터 시스템과는 근본적으로 다른 원리로 작동하는 혁신적인 계산 모델로서 양자 역학의 원리를 기반으로 하여, 정보 처리 및 계산을 전통적인 방식이 아닌 양자 비트(큐비트)라는 새로운 형태로 수행한다 [1-3]. 고전적인 컴퓨터에서 정보는 0과 1로 이루어진 이진수 형태로 처리되는 반면에, 양자 컴퓨터에서는 '중첩(superposition)' 상태를 이용하고, 큐비트들 간의 얽힘(entanglement) 현상을 활용하여, 여러 큐비트가 서로 긴밀히 연결되어 복잡한 계산을 동시에 처리할 수 있다 [4]. 양자 역학의 이러한 특성은 고전적인 컴퓨터가 처리하기 어려운 복잡한 문제를 더 효율적으로 해결할 수 있다 [5]. 예를 들면, 대규모 데이터의 분석, 암호 해독 및 최적화 문제 등에서는 양자 컴퓨터가 기존의 컴퓨터에 비해 훨씬 더 빠르며 강력한 계산력을 발휘할 수 있을 것으로 기대됨과 동시에 이러한 가능성 덕분에 양자 컴퓨팅은 과학, 공학, 금융 등 다양한 분야에서 혁신적인 변화에 기반한 중요한 기술로 자리잡고 있다는 것이다. 하지만, 양자 컴퓨터의 실제 구현과 활용에는 많은 도전이 요구되며, 양자 컴퓨팅의 이론적 장점이 실험실 환경에서 실제로 구현되기까지는 여러 기술적 한계가 있다 [6]. 특히 양자 알고리즘의 개발과 소프트웨어 검증은 여전히 초기 단계에 있다는 것이다. 이러한 문제를 해결함에 있어 양자 컴퓨터의 동작 원리를 이해하고, 이를 기반으로 한 효율적인 알고리즘을 개발하며, 소프트웨어의 정확성과 신뢰성을 보장하는 방법론이 필요하다. 양자컴퓨팅의 발전에 따라, 다양한 양자 알고리즘 및 소프트웨어들이 개발되고 있지만, 양자소프트웨어는 고전적인 컴퓨터 소프트웨어와는 다르게 복잡한 수학적 원리를 바탕으로 동작하므로 그 유사도를 평가하고, 다양한 구현체들 간의 비교를 통해 신뢰성을 확보하는 것이 필수적이다 [7].

따라서 본 논문에서는 양자컴퓨팅 소프트웨어의 유사도를 검증하는 방법론에 대해 논의한다. 이를 통해 양자 알고리즘 개발자들이 신뢰성 있는 소프트웨어를 개발할 수 있도록 도울 수 있다.

2. 관련연구

양자컴퓨터는 고전컴퓨터와는 다르게, 큐비트의 중첩(superposition)과 상태가 얽히는 얽힘(entanglement)이라는 양자역학적 특성을 기반으로 계산을 진행한다 [8]. 이러한 특성은 양자 알고리즘에서 병렬 처리와 효율적인 계산을 가능하게 하여 기존 고전적인 알고리즘보다 더 빠른 성능을 낼 수 있다. 그러나 고전적인 알고리즘을 양자 컴퓨터로 이식하거나 새로운 양자 알고리즘을 설계하는 것은 어려운 작업이다. 이 과정에서 소프트웨어의 유사도를 평가하는 것은 여러 알고리즘이나 구현이 동일한 결과를 도출하는지 확인하는 중요한 절차가 된다. 양자 컴퓨팅에서 유사도 검증은 양자 알고리즘이나 양자 회로가 의도한 대로 동작하는지, 그리고 예상한 결과와 실제 결과가 일치하는지를 확인하는 과정을 보인다. 이러한 양자 시스템의 특성상 오류나 잡음에 민감하기 때문에 매우 중요한 작업으로 인해 양자 알고리즘의 정확성을 보장하고, 신뢰성 있는 양자 컴퓨팅 시스템을 구축하기 위해 다양한 검증 방법이 개발되고 있다 [9]. 이에, 몇 가지 양자 컴퓨팅 유사도 검증 사례를 보여준다.

1) IBM의 양자 회로의 시뮬레이션에 의한 유사도 검증

IBM은 하드웨어 및 클라우드 서비스 개발을

기반으로 활발히 진행되고 있으며, 2016년도 5 큐비트 양자컴퓨터 공개와 더불어 자사의 양자 컴퓨터를 사용하는 양자 알고리즘 기반 개발이 가능할 수 있도록 자사의 클라우드 플랫폼 Q Experience를 제공하고 있다. 이 후 Q Experience의 16 큐비트 프로세서 시스템을 추가함으로서, 서킷 컴포저(circuit composer)와 퀀텀 랩을 통한 퀴스킷(Quantum Information Science Kit, Qiskit) 두 가지 방식으로 양자 회로를 구성할 수 있다. 퀴스킷은 양자 회로 및 알고리즘에 요구되는 개발 도구로 IBM OSS(Open Source Software) 플랫폼이다. 서킷 컴포저는 그래픽 드래그 앤 드랍 방식을 사용하기에 코딩이 아닌 회로를 구성하는 것이 가능하다. 또한, 퀴스킷의 경우 파이썬과 주피터 노트북을 이용한 코드로서 회로를 구성할 수 있다. Qiskit과 Cirq와 같은 플랫폼에서는 양자 회로를 설계하고 이를 고전적인 컴퓨터에서 시뮬레이션할 수 있는 도구를 제공한다. 이러한 도구는 양자 회로가 올바르게 동작하는지, 예상한 대로의 출력을 생성하는지 검증할 수 있다. 예를 들면, 양자 알고리즘을 구현한 후, 이를 고전적인 컴퓨터에서 시뮬레이션하여 기존의 고전적인 알고리즘과 결과를 비교하는 방식으로 유사도를 검증한다 [10].

2) Shor 코드에서의 양자 오류 수정 코드

Shor 코드는 양자 오류 정정 코드를 사용하여 양자 컴퓨터에서의 발생 가능 오류를 감지하고 수정한다. 이러한 오류 정정 코드의 작동 여부를 확인하기 위해, 시뮬레이션을 통해 오류를 주입하고 정정된 결과를 확인하는 방식으로 유사도를 검증한다. 결국, 양자 컴퓨터는 민감하고 오류가 발생하기 쉬운 시스템이므로, 양자 오류 정정 기법이 필수적이기에 오류 정정 기법이 제대로 작동하는지, 즉 오류를 효과적으로 수정하고 양자 시스템이 의도한 대로 동작하는지 검증하는 것이 중요하기 때문이다 [11].

3) 양자 알고리즘에 의한 실험적 검증

이론적인 양자 알고리즘과 실제 양자 컴퓨터에서 실행된 결과를 비교하는 방식으로 이루어지며, 실제 양자 하드웨어에서 양자 알고리즘이 동작하는지 확인하기 위해 실험적으로 검증하는 과정을 기반으로 한다 [12].

- Shor 알고리즘: 양자 컴퓨터가 소인수 분해 문제를 고전적인 컴퓨터보다 효율적으로 해결할 수 있다는 것을 보여주는 알고리즘으로서 작은 숫자에 대해 실험적으로 실행하고, 고전적인 방법으로 계산한 결과와 비교하여 유사도를 검증한다 [13].

- Grover 알고리즘: 데이터베이스 검색 문제에서 고전적인 방법보다 고 성능 기반 알고리즘으로 양자 컴퓨터에서 실행하고, 고전적인 방법으로 동일한 검색을 수행하여 실행 시간을 비교하는 방식으로 유사도를 검증한다 [14].

4) 양자 상태 Tomography

양자 상태 Tomography 기법을 사용하여 실험적으로 측정하고, 이를 이론적인 예측과 비교하는 것으로 예를 들면, 특정 양자 알고리즘을 실행한 후, 측정된 상태가 예상한 상태와 얼마나 유사한지 비교하는 방식으로 유사도를 검증할 수 있다. 또한, 양자 알고리즘 실행 후, 상태 벡터나 밀도 행렬을 계산하여 이론적 모델과 비교함으로써, 양자 시스템이 예상한 대로 동작하는지 평가된다.

3. 양자 컴퓨팅 SW 유사도 모델 제안

양자컴퓨팅 SW 유사도 모델은 IBM의 양자 컴퓨팅 프레임워크의 모델 검사를 통해 소프트웨어를 검증하는 것은 형식적인 재전송을 통해 소프트웨어를 모델링하는 것을 포함한다. 그런 다음 적절한 논리를 통해 원하는 동작을 지정하고, 두 구성 요소가 생성되면 모델 검사 알고리즘을 사용하여 모델이 지정된 동작을 따르는지 확인할 수 있다. 소프트웨어와 시스템의 시간적 동작으로 Kripke 구조를 사용하여 모델을 생성하고 템포-랄 논리를 사용하여 동작을 지정한다. 크립키 구조(Kripke Structure)는 유한 상태 기계와 유사한 방식으로 상태 간의 전이만을 설명함으로써 소프트웨어나 시스템을 모델링한다. 그러나 크립키 구조는 각 상태에서 유지되기에 그 속성도 모델링되기에 이에 대한 알고리즘은 다음과 같다.

$Def\,1.4-tuples\,M=(S,S_0,R,L)\,where$

$-\ S\,is\,a\,finite\,set\,of\,states;$

$-\ S_0\subseteq S\,is\,the\,set\,of\in ial\,states;$

$-\ R\subseteq S\,X\,S\,is\,a\,total\,transition\,\overset{tion,}{\text{a}}$

total transition relation,

$where\,\text{for}\,all\,s\in S,\ there\,\exists\,s'\in S$

$such\,that\,(s,s')\in R;$ ..... (1)

$-\ L:S\rightarrow 2^{AP}$ *is a labelling function that gives the set of propositions* $(p\in AP)$ *that hold within a given state.* ..... (2)

행동(State) 을 지정하는 데 사용되는 일반적인 유형의 논리는 시간 논리로, 시간이 지남에 따라 시스템에 대해 어떤 명제를 가질 수 있는지 설명하는 데 사용할 수 있다. CTL의 형식적 의미론은 먼저 크립케 구조$(M=(S,S_0,R,L))$의 경로를 정의할 수 있다.

$Def\,2.$ A path is a tuple $\sigma=(s_0,s_1,s_2,....)\,where\,s_i\in S\ and\ we\ have\ that\ (s_i,s_{i+1})\in R\,\text{for}\,all\,i.$ ..... (3)

경로는 적절한 전환이 있는 한 무한하거나 유한한 길이를 가질 수 있으며, 경로 양자화기와 결합된 상태로만 존재한다. 이에, 귀납적 정의는 다음과 같다.

$$\theta ::= p\|\neg\theta\|\theta\vee\theta\|ET\|AT$$
$$T ::= X\theta\|F\theta\|G\theta\|\theta\cup\theta,$$ ..... (4)

$P\in AP$는 이 모델의 명제로 $X(\text{“next”}),\ F(\text{“eventually”}),\ G(\text{“always”})\ and\ U(\text{“until”})$ 용어로 기본적인 시간 연산자를 나타낸다. 이에 양자 SW 유사도 검증을 위한 모델링을 도식화하면 다음과 같다.

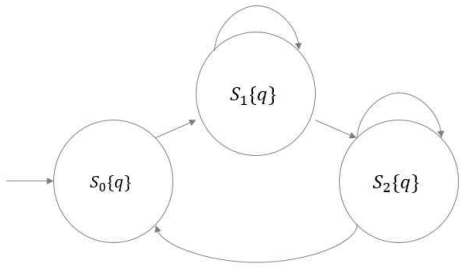

그림 1. 크립케 구조 $S=S_0,S_1,S_2$
Fig. 1. Kripke Structure with $S=S_0,S_1,S_2$

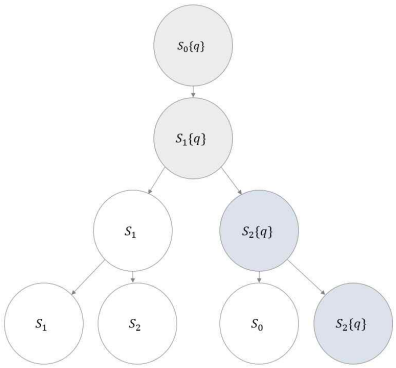

그림 2. 오토마타를 위한 계산 트리
Fig. 2. A computation tree for the Automata

그림 1과 2처럼 크립케 구조는 다음의 식으로 나타낸다.

$$[[P]]_M = s \in S : p \in L(s)$$
$$[[\neg\theta]]_M = S / [[\theta]]_M$$
$$[[\theta_1 \vee \theta_2]]_M = [[\theta_1]] \cup [[\theta_2]]_M$$
$$[[EX\theta]]_M = \{\ s \in S : \exists t \in S \ such\ that$$
$$s = s_0 \text{ and } \forall k \in N, s_k \in [[\theta]]_M\ \} \quad \text{..... (5)}$$

양자 계산 트리는 양자 크립키 구조의 동작을 추론하는데 사용하는 시간 논리로 모델 검증을 제안할 수 있다.

4. 결 론

양자 소프트웨어 검증은 전통적인 검증 기법만으로는 해결이 어렵고, 새로운 검증 패러다임이 필요하다. 이를 위해 시뮬레이션, 정형 검증, 하드웨어 협력 검증, AI 활용 등의 기술이 결합된 다각적인 접근이 요구된다. 양자 SW 검증은 아직 초기 단계이며, 기존 검증 기법을 직접 적용하는 것은 한계가 있다. 따라서 양자-고전 하이브리드 접근법, 시뮬레이션 기반 검증, 에러 보정 기법을 결합하여 실질적인 검증 전략을 구축해야 한다. 또한, 양자 하드웨어 발전과 함께 검증 기법도 점진적으로 개선될 필요가 있다. 즉, 현재로서는 완벽한 검증보다는 확률적, 경험적, 수학적 방법을 조합한 실용적인 검증 방식이 필요하며, 이를 위한 연구와 도구 개발이 지속적으로 이루어져야 한다.

이 논문은 2025학년도 배재대학교 교내학술연구비 지원에 의하여 수행됨

참 고 문 헌

[1] N. Leveson and C. Turner, “An investigation of the therac-25 accidents,” Computer, vol. 26, no. 7, pp. 18 - 41, Jul. 1993, ISSN: 0018-9162. DOI: 10.1109/mc.1993. 274940.

[2] C. W. Barrett, R. Sebastiani, S. A. Seshia, and C. Tinelli, Handbook of Satisfiability (Frontiers in artificial intelligence and applications 0922-6389 v. 185), A. Biere, Ed. Amsterdam: IOS Press, 2011, 966 pp., Master and use copy. Digital master created according to Benchmark for Faithful Digital Reproductions of Monographs and Serials, Version 1. Digital Library Federation, December 2002, ISBN: 160750376X. [Online]. Available: https://api.semanticscholar.org/CorpusID:12067495

[3] H. G. Rice, “Classes of recursively enumerable sets and their decision problems,” Transactions of the American Mathematical Society, vol. 74, no. 2, pp. 358 - 366, 1953, ISSN: 1088-6850. DOI: 10 . 1090 / s0002 - 9947 - 1953 - 0053041-6.

[4] A. M. Turing, “On computable numbers, with an application to the entscheidungs problcm. proceedings of the london mathematical society, 2 S. vol. 42 (1936 - 1937), pp. 230 - 265.,” The Journal of Symbolic Logic, vol. 2, no. 1, pp. 42 - 43, 1937, ISSN: 1943-5886. DOI: 10/d8n42j.

[5] R. Malik, S. Mohajerani, and M. Fabian, “A survey on compositional algorithms for verification and synthesis in supervisory control,” Discrete Event Dynamic Systems, vol. 33, no. 3, pp. 279 - 340, Aug. 2023, ISSN: 1573-7594. DOI: 10/gsnjqn.

[6] A. P. Kaleeswaran, A. Nordmann, T. Vogel, and L. Grunske, “A user study for evaluation of formal verification results and their explanation at bosch,” Empirical Software Engineering, vol. 28, no. 5, Sep.

2023, ISSN: 1573-7616. DOI: 10/gtpnbk.
[7] D. Beyer and A. Podelski, “Software model checking: 20 years and beyond,” in Principles of Systems Design. Springer Nature Switzerland, 2022, pp. 554 - 582, ISBN: 9783031223372. DOI: 10/gtpm9r.
[8] S. Riedmaier, B. Danquah, B. Schick, and F. Diermeyer, “Unified framework and survey for model verification, validation and uncertainty quantification,” Archives of Computational Methods in Engineering, vol. 28, no. 4, pp. 2655 - 2688, Aug. 2020, ISSN: 1886-1784. DOI: 10/gtpm93.
[9] S. A. Cook, “The complexity of theorem-proving procedures,” in Proceedings of the Third Annual ACM Symposium on Theory of Computing, ser. STOC ’71, Shaker Heights, Ohio, USA: Association for Computing Machinery, 1971, pp. 151 - 158, ISBN: 9781450374644. DOI: 10.1145/800157.805047. [Online]. Available: https: //doi.org/10.1145/800157.805047.
[10] E. Clarke, D. Kroening, and F. Lerda, “A tool for checking ansi-c programs,” in Tools and Algorithms for the Construction and Analysis of Systems (tacas 2004), K. Jensen and A. Podelski, Eds., ser. Lecture Notes in Computer Science, vol. 2988, Springer, 2004, pp. 168 - 176, ISBN: 3-540-21299-X. [Online]. Available: https://www.cs.cmu.edu/~svc/papers/view-publicationsckl2004.html
[11] L. Lin and Y. Tong, “Optimal polynomial based quantum eigenstate filtering with application to solving quantum linear systems,” Quantum 4, 361 (2020), vol. 4, p. 361, Oct. 31, 2019, ISSN: 2521-327X. DOI: 10/gtmtzg. arXiv: 1910.14596
[12] J. M. Martyn, Z. M. Rossi, A. K. Tan, and I. L. Chuang, “Grand unification of quantum algorithms,” PRX Quantum, vol. 2, p. 040 203, 4 Dec. 2021. DOI: 10/gnpx8n.
[13] A. Ambainis, D. Leung, L. Mancinska, and M. Ozols, “Quantum Random Access Codes with Shared Randomness,” arXiv, Tech. Rep., Jun. 2009, arXiv:0810.2937 [quant-ph] type: article. DOI: 10/grzghz. arXiv: 0810. 2937.
[14] A. P. Punnen, Ed., The Quadratic Unconstrained Binary Optimization Problem. Springer International Publishing, 2022. DOI: 10/gsjc23.

저 자 소 개

안철범(Chulbum Ahn)

2010.2 단국대학교 전자·컴퓨터공학과 박사
2018.3-현재 : 서일대학교 교수
<주관심분야> 데이터통신응용, 빅데이터, 네트워크 보안, 인공지능

김진홍(Jinhong Kim)

2006.2 성균관대학교 컴퓨터공학과 박사
2022.3-현재 한국SW감정평가학회 부회장
2017.3-2020.02 : 서일대학교 교수
2020.3-현재 : 배재대학교 교수
<주관심분야> 인공지능, 빅데이터, 지능형 소프트웨어