

## IEEE Spectrum

# Meet Twist: MIT's Quantum Programming Language > Keeping tabs on data entanglement keeps reins on buggy quantum code

BY RINA DIANE CABALLAR | 18 FEB 2022 | 3 MIN READ | 

```
1 fun teleport (q1 : qubit<P>) : qubit<P> =  
2   let (q2 : qubit<M>, q3 : qubit<M>) = bell_pair () in  
3   let (q1 : qubit<M>, q2 : qubit<M>) = CNOT (q1, q2) in  
4   let q1 : qubit<M> = H (q1) in  
5   let q3 = if measure (q2) then X (q3) else q3 in  
6   let q3 = if measure (q1) then Z (q3) else q3 in  
7   cast<P>(q3)
```

CODE: MIT: QUANTUM COMPUTER: GRAHAM CARLOW/IBM

### Meet Twist: MIT의 양자 프로그래밍 언어 데이터 얽힘에 대한 탭 유지하여 버그 있는 양자 코드에 대한 통제 유지

RINA DIANE CABALLAR  
2022년 2월 18일

MIT 컴퓨터 과학 및 인공 지능 연구소(Computer Science and Artificial Intelligence Laboratory, CSAIL)의 연구원 팀이 양자 컴퓨팅을 위한 새로운 프로그래밍 언어인 Twist를 만들었습니다. Twist는 개발자가 어떤 데이터 조각이 얽혀 있는지 쉽게 식별할 수 있도록 설계되어 오류가 적고 디버깅이 쉬운 양자 프로그램을 만들 수 있습니다.

Twist의 기반은 양자 컴퓨터 내부의 두 데이터 조각의 상태가 서로 연결되는 현상인 얽힘을 식별하는 데 있습니다. “얽힌 데이터 조각의 한 조각에 작업을 수행할 때마다 다른 조각에 영향을 줄 수 있습니다. 강력한 양자 알고리즘을 구현할 수는 있지만, 그것은 또

한 작성하는 프로그램에 대해 추론하는 것이 직관적이지 않고 미묘한 버그를 도입하는 것을 쉽게 만들 수 있습니다.”라고 MIT CSAIL에서 컴퓨터 공학을 전공하는 박사과정 학생이자 프로그래밍 언어에 관한 ACM의 회보에 게재된 Twist 에 대한 논문의 주저자인 Charles Yuan 이 말했습니다.

"Twist가 하는 일은 개발자가 어떤 데이터 조각이 얽혀 있고 어떤 데이터가 그렇지 않은 지 말할 수 있는 기능을 제공하는 것입니다."라고 Yuan은 말합니다. "프로그램 내부에 얽힘 정보를 포함시키면, 양자 알고리즘이 올바르게 구현되었는지 확인할 수 있습니다."

이 언어의 특징 중 하나는 개발자가 프로그램 내에서 어떤 표현식과 데이터 조각이 순수한지 지정할 수 있도록 하는 타입 시스템입니다. Yuan에 따르면 순수한 데이터 조각은 얽힘이 없으므로, 얽힘으로 인해 발생할 수 있는 버그와 직관적이지 않은 영향도 없습니다. Twist에는 또한 표현식이 다른 데이터 조각과 얽히지 않는다는 것을 확인하는 순도 주장 연산자와 이러한 주장을 검증하기 위한 정적 분석 및 런타임 검사가 있습니다.

언어를 평가하기 위해, 팀은 잘 알려진 양자 알고리즘 세트에 대한 프로그램을 Twist로 작성하고 양자 시뮬레이터에서 실행했습니다. "우리는 이러한 런타임 검사를 실행하는 오버헤드가 기본 프로그램을 실행하는 것보다 3.5% 이상 적다는 것을 보여주는 실험을 수행했는데, 이는 상당히 낮고 언어가 제공하는 안전 보장에 대한 좋은 절충안이라고 생각합니다."라고 Yuan은 말합니다.

팀은 또한 일부 프로그램에 작은 버그를 도입했으며 Twist가 이러한 버그를 감지하고 잘못된 프로그램을 거부할 수 있음을 발견했습니다. "우리는 사람들이 우리 언어를 사용하거나 특정 사용 사례에 대해 새로운 양자 언어를 설계할 때, 우리 작업을 보고 순수성과 얽힘을 기능으로 갖는 아이디어가 원하는 것이라고 말할 수 있기를 바랍니다. 왜냐하면 그것은 값비싼 시뮬레이션과 테스트를 많이 하지 않고도 그들의 프로그램이 정확하다는 더 많은 확신을 줄 것입니다."라고 Yuan은 말합니다.

많은 연구자들이 효율적이고 최적화된 양자 하드웨어를 구축하는 데 집중하고 있지만, Twist는 양자 소프트웨어의 격차를 메우는 것을 목표로 합니다. "모든 새로운 단계에서 새로운 시스템과 잠재적으로 새로운 기능을 얻을 수 있는 머신 러닝 및 기타 고성능 컴퓨팅 응용 프로그램에서 몇 가지 유사점을 도출하면, 하드웨어를 활용함으로써 엄청난 기회를 얻을 수 있을 것입니다. 그러나 사람들이 해당 하드웨어에 액세스하고 이를 다른 소프트웨어 시스템에 배포하고 널리 사용하는 것을 방해하는 것은 거의 항상 소프트웨어입니다."라고 MIT의 부교수이자 Twist에 대한 논문의 공동 저자인 Michael Carbin 은 말합니다. "우리가 하고 있는 많은 작업은 토대를 마련하고 이러한 유형의 장치를 더 프로그래밍 가능하게 만들 수 있는 핵심 추상화 중 일부가 무엇인지 알아내기 위해 노력하

는 것입니다."

그러나 Twist를 구축할 때 팀이 직면한 과제 중 하나는 양자 프로그램이 어떻게 생겼는지에 대한 표준이 없다는 것입니다. "수년에 걸쳐 사람들은 정수 인수 분해와 같은 복잡한 개별 작업을 해결하기 위한 핵심 알고리즘을 개발했지만, 이를 위한 전체 소프트웨어 생태계를 구축하는 방법은 명확하지 않습니다."라고 Yuan은 말합니다. "Twist를 사용하여, 양자 컴퓨터에서 수행하려는 작업에 대한 최상의 합의를 기반으로 언어를 구축할 수 있었고, 해당 작업에 대해 가능한 한 표현력을 높일 수 있었습니다."

제한 사항의 측면에서, Twist는 한 데이터 조각이 다른 데이터 조각과 얽혀 있는지 여부만 알려줄 수 있을 뿐 어떻게 얽혀 있는지는 알 수 없습니다. Yuan은 "그들이 얽히는 정확한 방법은 양자 알고리즘이 올바른지 여부를 결정하는 것이지만 데이터가 얽힐 수 있는 방법은 무한합니다."라고 말했습니다. "그렇게 세세한 세부 사항을 제공할 수 있다는 것은 정말 어려운 일이며, 앞으로 우리가 해야 할 일이기도 합니다."

팀은 현재 위상 및 중첩과 같은 다른 양자 현상을 처리하기 위해 Twist를 기반으로 하는 다른 언어를 연구하고 있지만, 그들은 Twist가 더 나은 양자 프로그램을 만들기 위한 길을 열어주기를 바라고 있습니다.

"양자 알고리즘을 구현하려는 개발자의 경우 프로그램에서 얽힘으로 인해 발생하는 무언가를 알려주는 도구가 언어에 내장되어 있어야 합니다."라고 Yuan은 말합니다. "개발자가 얽힘에 대해 추론할 수 있는 핵심 언어 원리와 기능을 구축할 수 있다면 얽힘이 인지적 부담을 덜고 개발자가 좀 더 직관적인 프로그램을 작성할 수 있도록 만들 수 있습니다."

[출처]

<https://spectrum.ieee.org/quantum-programming-language-twist>