

계산과학2

노현민-김태원 조

2023년 10월 10일

요 약

Ewin Tang의 2023년 Parc City Mathematics Institute 대학원생 대상 강의 *Quantum and quantum-inspired linear algebra*(<https://ewintang.com/pcmi/>)를 중심으로 블록인코딩, QSVT, 양자 시뮬레이션, 양자 선형대수의 개념을 익힌다.

차 례

차 례	2
제 0 장 기초 양자역학	3
0.1 에르미트 연산자	3
0.2 스핀과 파울리 행렬	3
제 1 장 블록-인코딩	5
1.1 블록-인코딩	6
1.2 블록-인코딩의 확장가능성	7
제 2 장 양자 특잇값 변환	10
제 3 장 다항식에 의한 근사	11
제 4 장 양자 선형대수	12
제 5 장 양자적 영향상의 알고리즘	13

기초 양자역학

0.1 에르미트 연산자

양자역학에 따르면 우주는 복소 공간이다. 하지만 우리의 직관 혹은 실험 및 관측 결과는 실수여야 한다. 그래서 우리가 측정할 수 있는 대상인 관측량을 표현하는 수학적 도구는 이들 두 공간을 잘 이어줘야 한다. 그것이 바로 **에르미트 연산자**다. 선형 연산자 M 은 다음 성질을 만족하는 경우 에르미트 연산자다.

$$M = M^\dagger$$

관측량은 관측량을 나타내는 연산자의 고윳값과 대응하는데, 요컨대 에르미트 연산자의 고윳값은 모두 실수다. 임의의 에르미트 연산자 L 과 고윳값 λ 와 고유벡터 $|\lambda\rangle$ 에 대해 아래가 성립한다.

$$\begin{aligned} L|\lambda\rangle &= \lambda|\lambda\rangle \\ \Rightarrow \langle\lambda|L^\dagger &= \langle\lambda|\lambda^* \\ \Rightarrow \langle\lambda|L &= \langle\lambda|\lambda^* \\ \Rightarrow \langle\lambda|L|\lambda\rangle &= \langle\lambda|\lambda|\lambda\rangle \text{ \& } \langle\lambda|L|\lambda\rangle = \langle\lambda|\lambda^*|\lambda\rangle \\ \Rightarrow \langle\lambda|\lambda|\lambda\rangle - \langle\lambda|\lambda^*|\lambda\rangle &= 0 \\ \Rightarrow \lambda = \lambda^* &\implies \lambda \in \mathbb{R} \end{aligned}$$

또한 명확히 구분할 수 있는 결과를 표현하는 고유벡터는 서로 다른 고윳값을 가져야 한다. 다시 말해 상이한 대상에 대해 상이한 실험 결과가 보증되어야 한다. $\lambda_1 \neq \lambda_2$ 에 대해 아래가 성립한다.

$$\begin{aligned} L|\lambda_1\rangle &= \lambda_1|\lambda_1\rangle \text{ \& } L|\lambda_2\rangle = \lambda_2|\lambda_2\rangle \\ \Rightarrow \langle\lambda_1|L|\lambda_2\rangle &= \lambda_1\langle\lambda_1|\lambda_2\rangle \text{ \& } \langle\lambda_1|L|\lambda_2\rangle = \lambda_2\langle\lambda_1|\lambda_2\rangle \\ \Rightarrow 0 &= (\lambda_1 - \lambda_2)\langle\lambda_1|\lambda_2\rangle \implies \langle\lambda_1|\lambda_2\rangle = 0 \end{aligned}$$

즉 두 고유벡터는 직교한다.

0.2 스핀과 파울리 행렬

이를 비롯 에르미트 연산자는 양자역학적인 세계와 우리의 관측 및 실험을 적절하게 잇는 좋은 성질을 여럿 지닌다. 여기서 우리가 관측하고자 하는 주요 대상은 바로 전자^{electron}의 운동이다. 전자의 운동을 설명하는 데는 일련의 변수가 필요하다. 위치 좌표와 별개로 전자는 **스핀**이라는 부가적인 변수 혹은 자유도를 지닌다.

스핀이란 완전히 양자역학적인 개념으로, 고전 역학 및 우리의 직관에 대응되는 개념을 찾기 어

렵다. 거칠게 스핀은 x, y, z 축에 따라 아래처럼 구성된다고 할 수도 있다.

$$\sigma_x, \sigma_y, \sigma_z$$

위와 같은 스핀의 성분을 일종의 연산자로 보면, 고유벡터와 고유값이 존재할 것이다. 이를테면 σ_z 는 아래와 같은 연산자다.

$$\begin{aligned}\sigma_z \begin{pmatrix} 1 \\ 0 \end{pmatrix} &= +1 \begin{pmatrix} 1 \\ 0 \end{pmatrix} \\ \sigma_z \begin{pmatrix} 0 \\ 1 \end{pmatrix} &= -1 \begin{pmatrix} 0 \\ 1 \end{pmatrix}\end{aligned}$$

여기서 고유값 ± 1 은 $(1, 0), (0, 1)$ 에 대한 σ_z 의 관측 결과다. 그렇다면 σ_z 는 어떻게 생겼을까?

$$\begin{aligned}\begin{pmatrix} \sigma_{z11} & \sigma_{z12} \\ \sigma_{z21} & \sigma_{z22} \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} &= \begin{pmatrix} 1 \\ 0 \end{pmatrix} \\ \begin{pmatrix} \sigma_{z11} & \sigma_{z12} \\ \sigma_{z21} & \sigma_{z22} \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} &= -\begin{pmatrix} 0 \\ 1 \end{pmatrix}\end{aligned} \Rightarrow \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

여기서 $(1, 0), (0, 1)$ 을 $|0\rangle, |1\rangle$ 로 표기하여 상태 $|A\rangle$ 를 일반화하면 아래와 같다.

$$|A\rangle = \alpha_0 |0\rangle + \alpha_1 |1\rangle$$

여기서 α_1, α_2 는 이른바 진폭이며, 이를 제공한 것이 바로 확률이다. 따라서 임의의 $\alpha_i \in \mathbb{C}$ 는 아래와 같은 보른 규칙을 만족해야 한다.

$$|\alpha_1|^2 + |\alpha_2|^2 + \dots + |\alpha_n|^2 = 1$$

그리하여 $|0\rangle, |1\rangle$ 의 일차결합으로 아래와 같은 벡터를 나타낼 수 있다.

$$|r\rangle = \frac{1}{\sqrt{2}} |0\rangle + \frac{1}{\sqrt{2}} |1\rangle, \quad |l\rangle = \frac{1}{\sqrt{2}} |0\rangle - \frac{1}{\sqrt{2}} |1\rangle$$

σ_x 가 고유값 ± 1 에 대해 위와 같은 $|r\rangle, |l\rangle$ 을 고유벡터로 지닌다고 하면, σ_x 는 아래와 같을 것이다.

$$\begin{pmatrix} \sigma_{x11} & \sigma_{x12} \\ \sigma_{x21} & \sigma_{x22} \end{pmatrix} \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix} = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix}, \quad \begin{pmatrix} \sigma_{x11} & \sigma_{x12} \\ \sigma_{x21} & \sigma_{x22} \end{pmatrix} \begin{pmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{pmatrix} = -\begin{pmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{pmatrix} \Rightarrow \sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

α_2 를 $\pm \frac{1}{\sqrt{2}}$ 가 아니라 $\pm \frac{i}{\sqrt{2}}$ 로 일반화하면 σ_y 를 얻을 수 있다. 이들 스핀 연산자를 모두 모은 것이 바로 **파울리 행렬**이다.

$$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

또한 이때 스핀을 제 전자로부터 고립시켜 파울리 행렬로 구성한 것이 바로 **큐비트**다.

제 1 장

블록-인코딩

양자 계(quantum systems)의 시뮬레이션은 양자 컴퓨터 개발의 주요 동기 가운데 하나다. 이들 양자 계 시뮬레이션 가운데 가장 간결한 것이 이른바 **해밀토니언 시뮬레이션**이다.

해밀토니언은 자신의 환경과 상호작용하는 관측량 혹은 파울리 행렬들의 텐서곱 E_a 에 그러한 상호작용의 강도를 결정하는 λ_a 를 곱한 것의 총합 H 로 볼 수 있다.

$$H = \sum_{a=1}^m \lambda_a E_a$$

그리고 해밀토니언 시뮬레이션은 아래와 같이 정의되는 문제다.

문제 1.1. $\|U - e^{iHt}\| \leq \epsilon$ 이도록 e^{-iHt} 에 가까운 유니터리 U 를 구현하는 알고리즘을 찾아라.

우선 **유니터리**와 e^{-iHt} 라는 표기가 무슨 뜻인지 알 필요가 있다. 시간 t 에 상태 $|\psi\rangle$ 에 있는 닫힌 계가 있으며 $|\psi\rangle$ 가 특정한 시간 t 에 $|\psi\rangle$ 였다는 사실을 $|\psi(t)\rangle$ 로 표기하자. 초기 상태 $|\psi(0)\rangle$ 에서 계의 시간이 전개되어 $|\psi(t)\rangle$ 에 이르렀다는 사실은 아래처럼 나타낸다.

$$|\psi(t)\rangle = U(t) |\psi(0)\rangle$$

앞서 언급했듯 직교성으로 인해 2개의 다른 기저 벡터는 2개의 구분 가능한 상태를 나타낸다. 이처럼 구분 가능한 두 개의 상태를 $|\psi(0)\rangle, |\phi(0)\rangle$ 이라고 하자.

$$\langle\psi(0)|\phi(0)\rangle = 0$$

임의의 시간이 흐른 이후에도 이 둘은 구분될 것이다. 이런 성질을 만족하는 것이 유니터리 U 다.

$$\langle\psi(t)|\phi(t)\rangle = 0 \Rightarrow \langle\psi(0)|U^\dagger(t)U(t)|\phi(0)\rangle = 0 \Rightarrow U^\dagger U = I$$

유니터리 변환은 $t = \epsilon = 0$ 인 경우 그냥 I 다. 그런데 ϵ 이 극미량이면 아래처럼 쓸 수 있다.

$$U(\epsilon) = I - i\epsilon H, \quad U^\dagger(\epsilon) = I + i\epsilon H^\dagger$$

극소 시간 $t = \epsilon$ 으로 특정하여 시간 전개식을 다시 쓰면 아래와 같다.

$$\begin{aligned} |\psi(\epsilon)\rangle &= |\psi(0)\rangle - i\epsilon H |\psi(0)\rangle \Rightarrow \frac{|\psi(\epsilon)\rangle - |\psi(0)\rangle}{\epsilon} = -iH |\psi(0)\rangle \\ &\Rightarrow i \frac{d}{dt} |\psi\rangle = H |\psi\rangle \\ &\Rightarrow |\psi(t)\rangle = e^{-iHt} |\psi(0)\rangle \end{aligned}$$

정리하면, 해밀토니언 시뮬레이션은 문제는 말 그대로 해밀토니언을 시뮬레이션하는 문제다. 대표적인 해법은 **트로터 근사**다. 충분히 큰 $r \rightarrow \infty$ 에 대해 아래가 성립한다.

$$e^{-iHt} \approx \left(e^{-iE_1 t/r} e^{-iE_2 t/r} \dots e^{-iE_m t/r} \right)^r$$

이 근사를 구현하기 위해서는 $\frac{1}{\epsilon}$ 에 비례하는 게이트가 필요하다. 즉 게이트 복잡도가 $\text{poly}(1/\epsilon)$ 이다. 결코 최적해가 아니다. 이후 개선된 알고리즘들이 나타났으며 이들은 아래 프레임워크로 이어진다.

- (i) “블록-인코딩”이라는 유형의 양자 회로를 정의한다.
- (ii) λ_a, E_a 가 주어질 때 H 의 효율적인 블록-인코딩을 구성할 수 있다는 사실을 보인다.
- (iii) H 의 블록-인코딩을 적게 사용하여 e^{-iHt} 의 (근사의) 블록-인코딩을 취할 수 있다는 사실을 보인다.
- (iv) 이 블록-인코딩을 사용해 상태에 대한 근사에 적용한다.

1.1 블록-인코딩

정의 1.1. $A \in \mathbb{C}^{r \times c}$ 가 주어질 때, $U \in \mathbb{C}^{d \times d}$ 는 U 가 $\mathcal{O}(Q)$ 게이트로 구현가능하고 항등행렬의 첫 r 열과 c 열인 $B_{L,1} \in \mathbb{C}^{d \times r}, B_{R,1} \in \mathbb{C}^{d \times c}$ 에 대해

$$B_{L,1}^\dagger U B_{R,1} = A \quad (1.1)$$

를 만족하면 A 의 Q -블록 인코딩이라고 부른다. 다시 말해 아래와 같다.

$$U = \begin{pmatrix} A & \cdot \\ \cdot & \cdot \end{pmatrix} \quad (1.2)$$

여기서 \cdot 은 U 의 임의의 원소를 나타낸다.

방정식 1.1과 방정식 1.2가 같은 말이라는 사실을 이해하는 것이 중요하다. $c = 2, d = 8, r = 4$ 라고 가정하자. 다시 말해 $A \in \mathbb{C}^{4 \times 2}, U \in \mathbb{C}^{8 \times 8}, B_{L,1} \in \mathbb{C}^{8 \times 4}, B_{R,1} \in \mathbb{C}^{8 \times 2}$ 라고 가정하자.

$$\begin{aligned} B_{L,1}^\dagger &= \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}, U = \begin{pmatrix} A & \cdot \\ \cdot & \cdot \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{21} & a_{22} & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{31} & a_{32} & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{41} & a_{42} & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{pmatrix}, B_{R,1} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix} \\ \Rightarrow B_{L,1}^\dagger U &= \begin{pmatrix} a_{11} & a_{12} & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{21} & a_{22} & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{31} & a_{32} & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{41} & a_{42} & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{pmatrix} \Rightarrow B_{L,1}^\dagger U B_{R,1} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \\ a_{41} & a_{42} \end{pmatrix} = A \end{aligned}$$

여기서 d, r, c 를 2^n 으로 고려하면 위 항들을 큐비트로 받아들일 수 있다.

$$(\langle 0|^{\otimes a_L} \otimes I)U(|0\rangle^{\otimes a_R} \otimes I) = A \quad (1.3)$$

여기서 $|0\rangle^{\otimes a_R}$ 은 두 가지 의미로 해석할 수 있는데, 첫째는 말 그대로 $|0\rangle$ 을 a_R 번 텐서곱하는 것이다. 즉

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \Rightarrow |0\rangle \otimes |0\rangle = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \Rightarrow |0\rangle^{\otimes a_R} = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \quad |0\rangle^{\otimes a_R} \otimes I = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

그리하여 d, r, c 가 2^n 일 때 $B_{L,1}^\dagger = \langle 0|^{\otimes a_L} \otimes I, B_{R,1} = |0\rangle^{\otimes a_R} \otimes I$ 다.

다른 한편 양자회로의 각 와이어는 텐서곱으로 인터랙션한다. 다시 말해 $|0\rangle$ 으로 초기화된 와이어가 a 개 있다고 볼 수 있다. 그런데 이런 블록-인코딩이 필요한 이유는 무엇일까? 우선 아래 보조정리를 확인하자.

보조정리 1.1. 유니타리 U 를 Q 게이트로 구현하는 양자 회로는 U 의 Q -블록 인코딩이다.

당연한 사실이다. 한편 $|\psi\rangle \mapsto U|\psi\rangle$ 와 같은 사상을 위해 유니타리 U 를 구현하는 회로를 사용하는 것과 마찬가지로, A 의 블록-인코딩 또한 $|\psi\rangle \mapsto A|\psi\rangle$ 와 같은 사상을 수행하기 위해 쓰일 수 있다. 물론 실패할 확률이 존재하지만, 이는 유니타리 회로가 제공하는 것보다 더 일반적인 유형의 선형대수학적 연산을 수행할 수 있도록 한다. 즉 블록-인코딩은 유니타리 양자 회로의 일반화다.

보조정리 1.2. $A \in \mathbb{C}^{r \times c}$ 의 Q -블록 인코딩 $U \in \mathbb{C}^{d \times d}$ 와 상태 $|\psi\rangle \in \mathbb{C}^c$ 가 주어질 때, 상태 $A|\psi\rangle$ 를 $\mathcal{O}(Q)$ 게이트와 $\|A|\psi\rangle\|^2$ 의 확률로 생산하는 양자 회로가 존재한다.

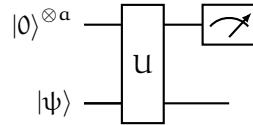


그림 1.1: 기본적인 블록-인코딩 회로. U 가 행렬 $A \in \mathbb{C}^{r \times c}$ 의 블록-인코딩이라면 첫 번째 와이어의 측정 결과가 $|0\rangle^{\otimes a}$ 인 경우 회로의 출력은 $A|\psi\rangle$ 다.

Proof. 그림 1.1상의 회로를 보라. 우리는 상태 $|\psi\rangle$ 를 취해 $|0\rangle$ 으로 초기화된 a_R 큐비트를 더한다. 그런 다음, 블록-인코딩 U 를 적용하고 첫 a_L 큐비트를 측정한다. 이들 모두 0이라는 결과를 지닌다면, 방정식 1.3에 의해 최종 상태는 $A|\psi\rangle$ 다. 이는 확률 $\|A|\psi\rangle\|^2$ 로 일어난다. \square

여기서 측정이 무엇인지 이해해야 한다. 위 증명에서 측정 직전 양자 계의 상태는 $U(|0\rangle^{\otimes a_R} \otimes I)|\psi\rangle$ 였고 이에 $|0\rangle^{\otimes a_L}$ 를 측정한 결과가 모두 0인지, 즉 모든 a_L 큐비트가 $\langle 0|$ 으로 결정되느냐는 것이 관건이다.

$$P(0) = \|(\langle 0|^{\otimes a_L} \otimes I)U(|0\rangle^{\otimes a_R} \otimes I)|\psi\rangle\|^2 = \|A|\psi\rangle\|^2 = P(A|\psi\rangle)$$

1.2 블록-인코딩의 확장가능성

행렬의 효율적인 블록-인코딩을 만들 수 있는 조건은 무엇인가? 이에 해밀토니언 시뮬레이션 문제를 재고한다.

문제 1.2. $\{E_a\}_{a \in [m]}$ 의 1-블록 인코딩이 주어져 해밀토니언 $H = \sum_{a=1}^m \lambda_a E_a$ 를 정의할 때, e^{-iHt} 의 (근사의) 블록-인코딩을 취할 수 있는가?

블록-인코딩은 여러 **확장 성질** extensibility properties을 지닌다. 즉, A와 B의 블록-인코딩이 주어질 때, AB 와 $c_0A + c_1B$ 의 블록-인코딩을 취할 수 있다. 마찬가지로 크기 조정 상수 α 에 대해 H/α 의 블록-인코딩을 취할 수 있다.

보조정리 1.3. U 와 V 가 $A \in \mathbb{C}^{r \times s}$ 와 $B \in \mathbb{C}^{s \times t}$ 의 Q_U 및 Q_V 블록-인코딩이라고 하자. 이에 AB 의 $(Q_U + Q_V)$ -블록 인코딩을 구성할 수 있다.

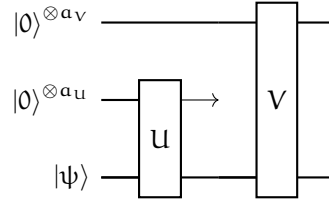


그림 1.2: U 가 A 의 블록-인코딩이고 V 가 B 의 블록-인코딩이라면 이 회로는 AB 의 블록-인코딩이다. 여기서 a_u 와 a_v 는 각각의 블록-인코딩에 필요한 패딩이다.

Proof. AB 를 구현하는 회로가 그림 1.2에 있다. 이는 그림 1.1의 회로 두 개의 합성 composition이므로 AB 의 블록-인코딩이다. \square

유니타리의 선형 결합 Linear Combination of Unitaries (LCU) 알고리즘으로 블록-인코딩의 선형결합의 블록-인코딩을 구성할 수 있다.

보조정리 1.4. 모든 $i = 0, \dots, k-1$ 에 대해 $U^{(i)}$ 가 $A^{(i)}$ 의 $Q^{(i)}$ -블록-인코딩이라고 하자. 이에 모든 $\alpha_i \in \mathbb{C}$ 에 대해 $\sum |\alpha_i| \leq 1$ 인 $\sum \alpha_i U^{(i)}$ 의 $(k + \sum_{i=0}^{k-1} Q^{(i)})$ -블록인코딩을 구성할 수 있다.

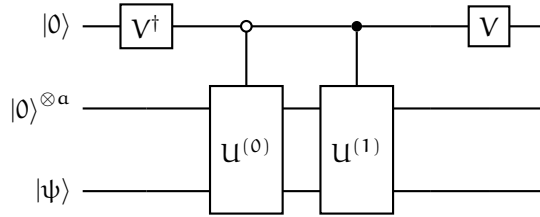


그림 1.3: $U^{(0)}$ 과 $U^{(1)}$ 이 $A^{(0)}$ 과 $A^{(1)}$ 의 블록-인코딩이라고 할 때, 이 회로는 $|V_{0,0}|^2 A^{(0)} + |V_{0,1}|^2 A^{(1)}$ 의 블록-인코딩으로, 입력 $|\psi\rangle$ 에 대해 쓰인다. 여기서 $U^{(0)}$ 과 $U^{(1)}$ 은 $|0\rangle$ 과 $|1\rangle$ 에 의해 통제된다.

Proof. 우선 $k = 2$ 블록-인코딩의 선형결합을 고려하자. 그림 1.3은 선형결합을 구현하는 회로를 나타낸다. 통제- $U^{(0)}$ 과 통제- $U^{(1)}$ 은 아래 유니타리를 적용한다.

$$\begin{pmatrix} U^{(0)} & \\ & I \end{pmatrix} \begin{pmatrix} I & \\ & U^{(1)} \end{pmatrix} = \begin{pmatrix} U^{(0)} & \\ & U^{(1)} \end{pmatrix} = (|0\rangle\langle 0| \otimes U^{(0)} + |1\rangle\langle 1| \otimes U^{(1)}) \quad (1.4)$$

따라서 전체 회로는 아래를 수행한다.

$$\begin{pmatrix} V_{0,0}I & V_{0,1}I \\ V_{1,0}I & V_{1,1}I \end{pmatrix}^\dagger \begin{pmatrix} U^{(0)} & \\ & U^{(1)} \end{pmatrix} \begin{pmatrix} V_{0,0}I & V_{0,1}I \\ V_{1,0}I & V_{1,1}I \end{pmatrix} = (V^\dagger |0\rangle \langle 0| V) \otimes U^{(0)} + (V^\dagger |1\rangle \langle 1| V) \otimes U^{(1)} \quad (1.5)$$

이 행렬에서 블록-인코딩에 해당하는 것은 아래와 같다.

$$|V_{00}|^2 U^{(0)} + |V_{1,0}|^2 U^{(1)} \quad (1.6)$$

따라서 임의의 음이 아니며 합이 1인 두 실수 α_0, α_1 에 대해 단일 큐비트 유니타리 V 가 존재하여 첫 번째 열이 $\sqrt{\alpha_0}, \sqrt{\alpha_1}$ 이다. 이에 위 블록-인코딩이 가능하다. α_0 가 음이면 $|\alpha_0|$ 에 대해 회로를 설계 하여 $U^{(0)}$ 대신 $-U^{(0)}$ 을 통제 유니타리로 사용하면 된다.

일반화하면, U 가 A 의 블록-인코딩일 때, $I \otimes U$ 또한 A 의 블록-인코딩이기에 추가 손실 없이 $U^{(i)}$ 의 크기에 따라 차원을 요컨대 $d \times d$ 로 늘릴 수 있다. 또한, 추가 손실 없이 $U^{(i)} = I$ 와 $\alpha_i = 0$ 를 선형결합에 더해 k 를 2의 제곱수로 만들 수 있다. $V \in \mathbb{C}^{k \times k}$ 가 아래를 만족하며

$$V|0\rangle = \sum_{i=0}^k \sqrt{|\alpha_i|} |i\rangle$$

$U \in \mathbb{C}^{kd \times kd}$ 가 아래와 같은 유니타리라고 하자.

$$\sum_{i=0}^{k-1} (|k\rangle \langle k|) \otimes \left(\frac{\alpha_i}{|\alpha_i|} U^{(i)} \right)$$

그렇다면 $(V^\dagger \otimes I)U(V \otimes I)$ 는 $\sum \alpha_i U^{(i)}$ 의 블록-인코딩이다. V 를 적용하는 비용은 $\mathcal{O}(k)$ 이며 통제- $U^{(i)}$ 사용 비용이 $U^{(i)}$ 자체 비용의 상수배라고 할 때, U 의 비용은 $\mathcal{O}(k + \sum_i Q^{(i)})$ 다. \square

제 2 장

양자 특잇값 변환

제 3 장

다항식에 의한 근사

제 4 장

양자 선형대수

제 5 장

양자적 영향상의 알고리즘