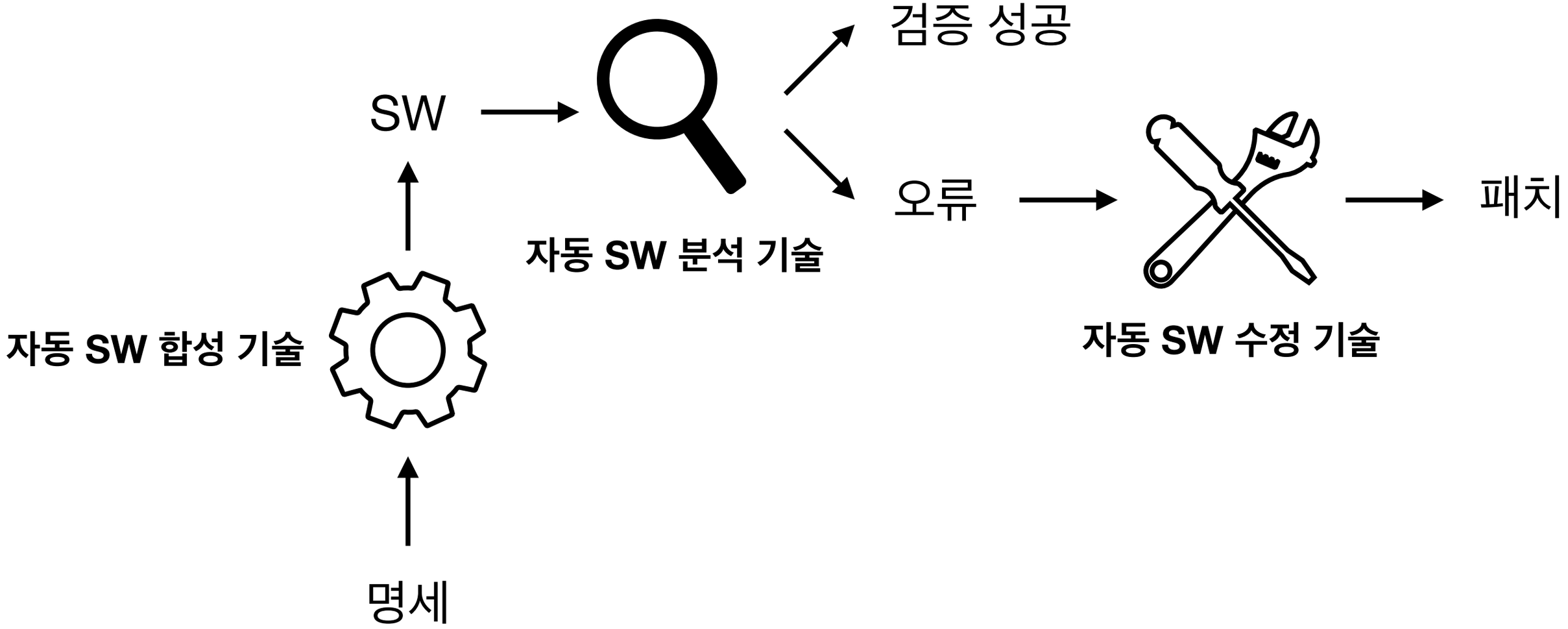
고려대 양자 프로그래밍 연구 소개

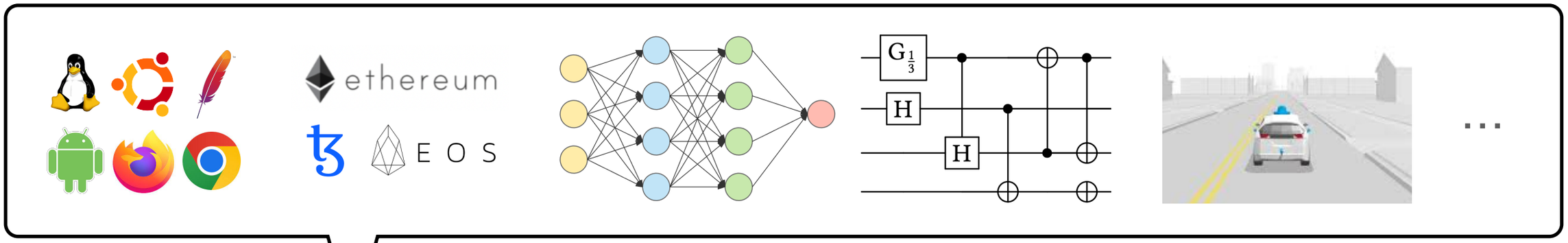
오학주
고려대학교 컴퓨터학과

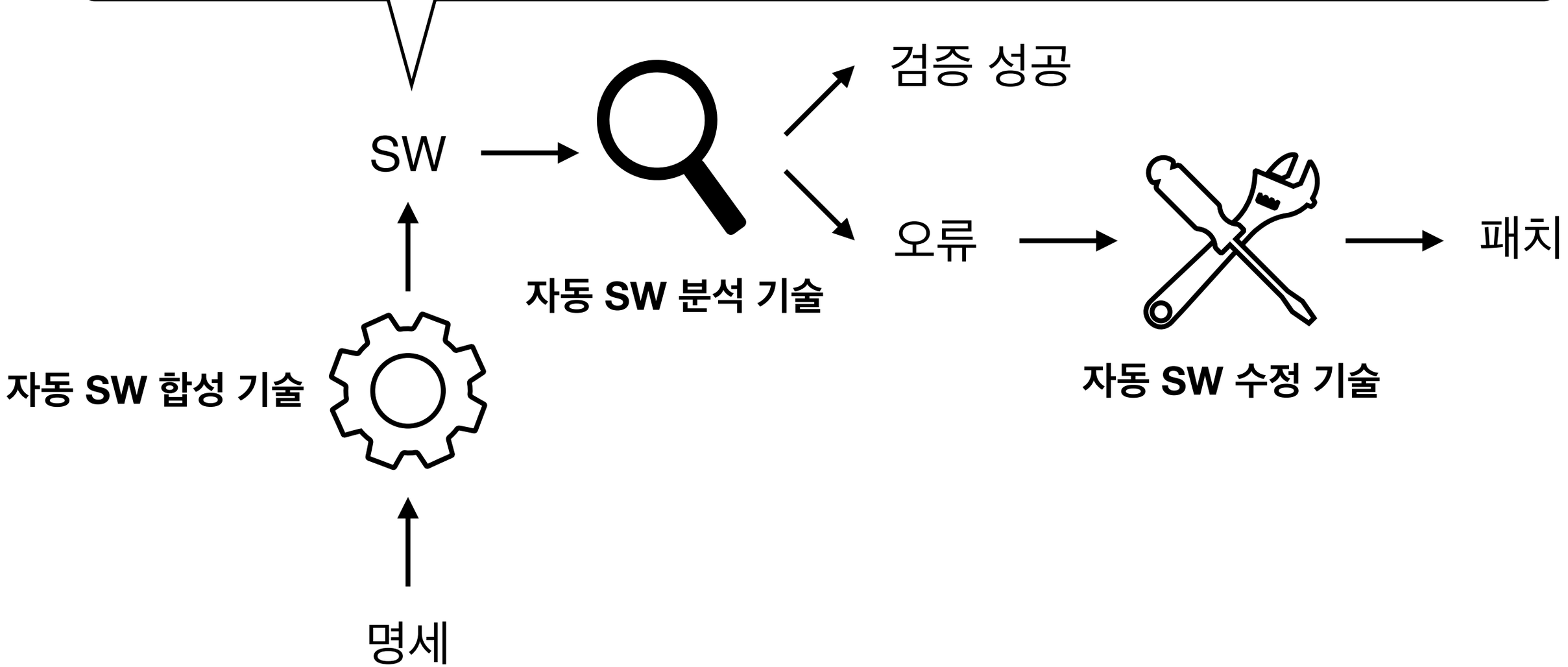
02/03/2023@ERC Workshop

Our Research

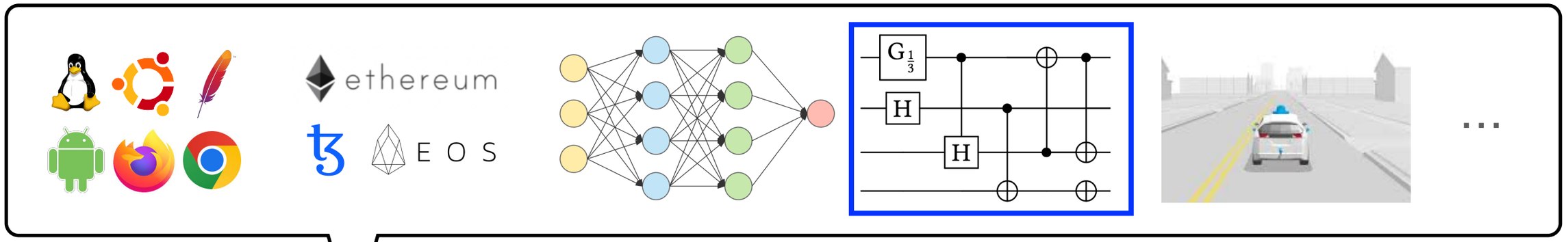


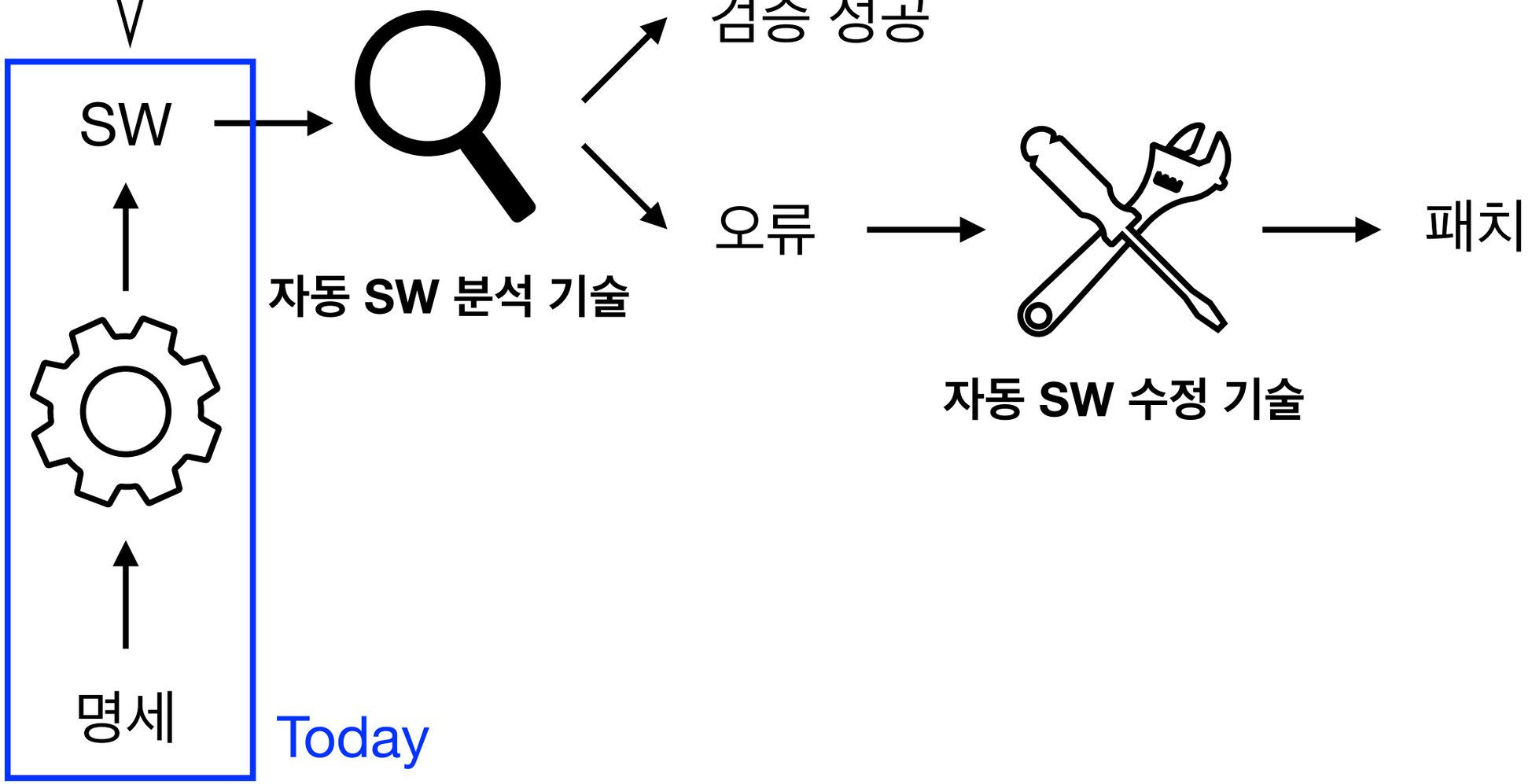
Our Research



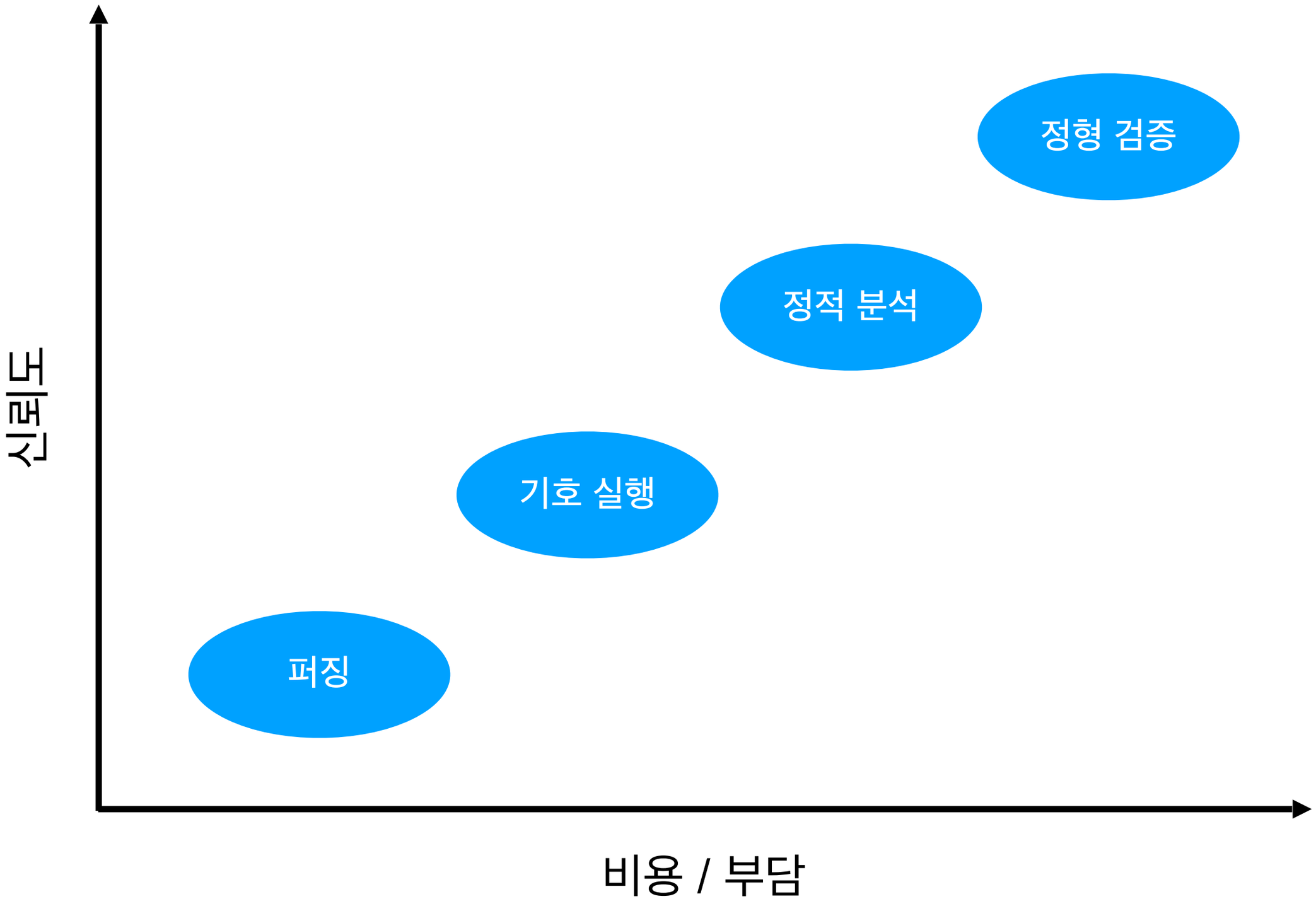


Our Research



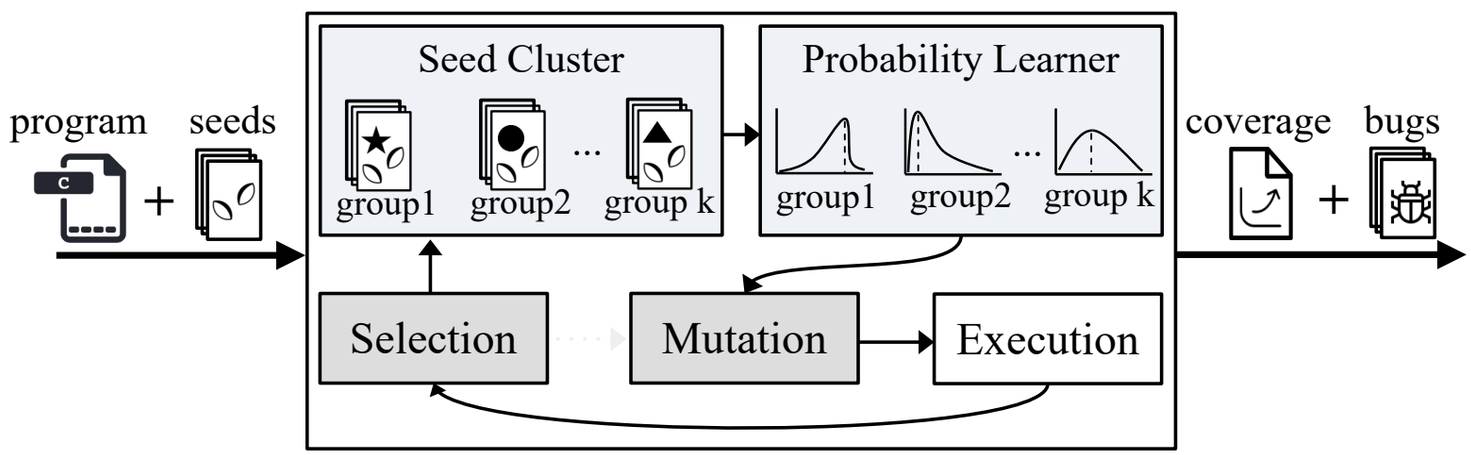


프로그램 분석



퍼징

- Seed-adaptive grey-box fuzzing [ICSE 2023]

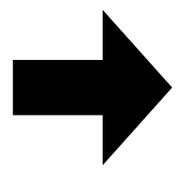


- Finding soundness bugs in SMT solvers [ICSE 2023]

기호 실행

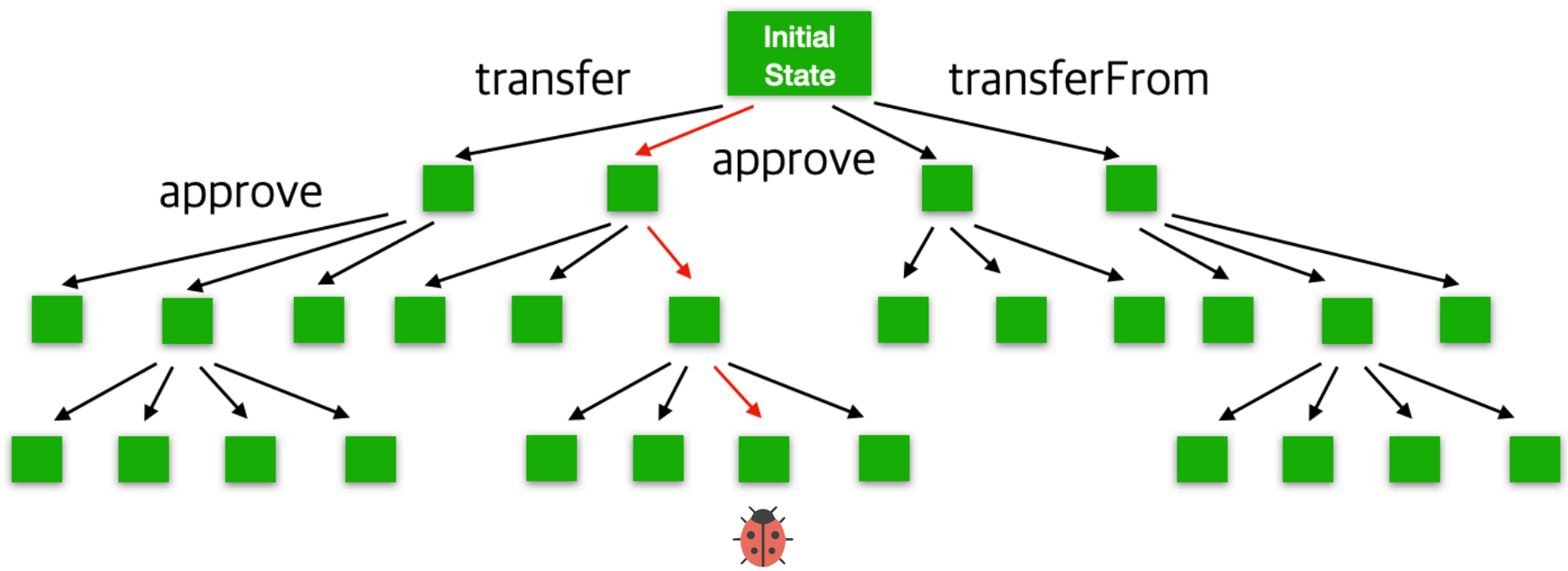
- Automated parameter configuration [ICSE 2022]

```
$ klee --simplify-sym-indices --max-memory=1000 --optimize
      --use-forked-solver --use-cex-cache --external-calls=all
      --max-sym-array-size=4096 --max-instruction-time=30s
      --max-time=60min --max-memory-inhibit=false
      --max-static-fork-pct=1 --max-static-solve-pct=1
      --max-static-cpfork-pct=1 --switch-type=internal
      --search=random-path --search=nurs:covnew
      --batch-instructions=10000 ./pgm.bc --sym-args 0 1 10
      --sym-args 0 2 2 --sym-files 1 8 --sym-stdin 8 --sym-stdout ...
```



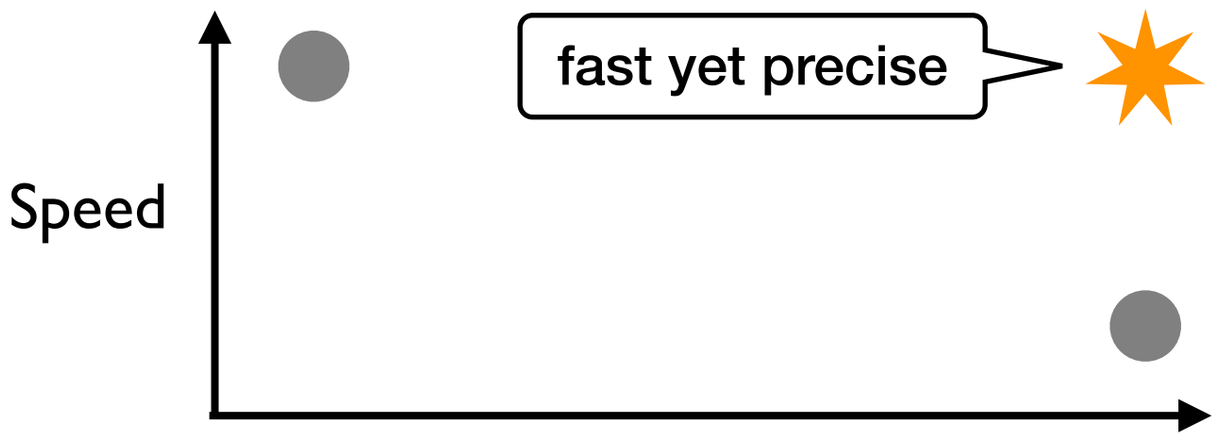
```
$ klee pgm.bc
```

- Language model-guided symbolic execution [Security 2021]

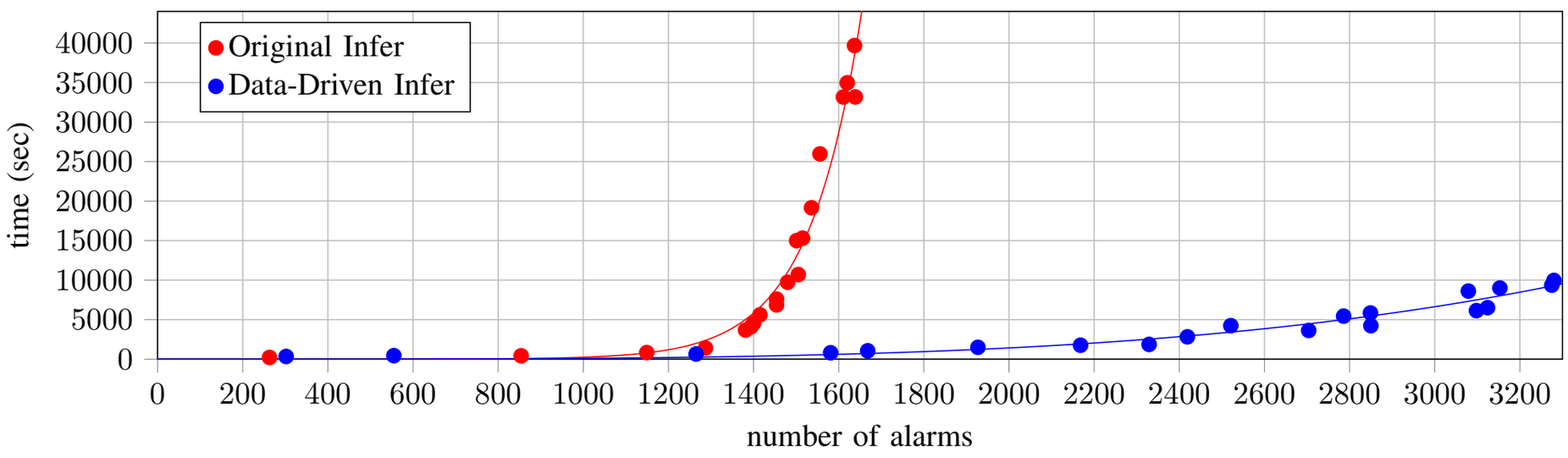


정적 분석

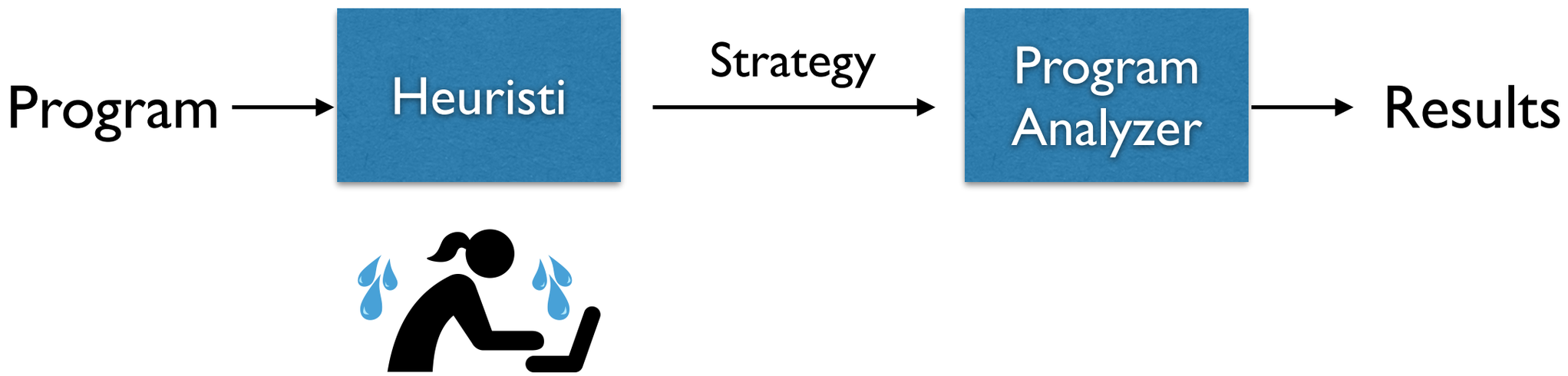
- Balancing precision & scalability of sound analysis [POPL 2022]



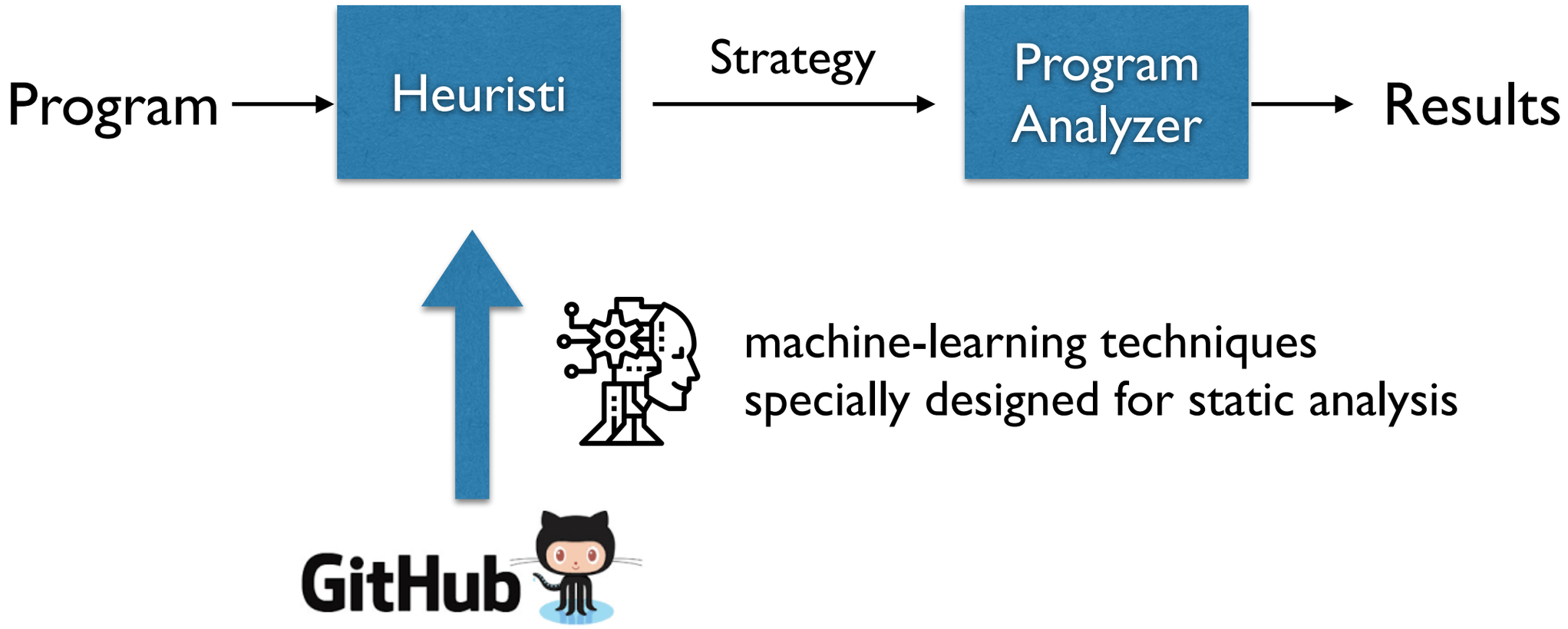
- Boosting unsound, path-sensitive bug-finders [ICSE 2023]



데이터 기반 프로그램 분석

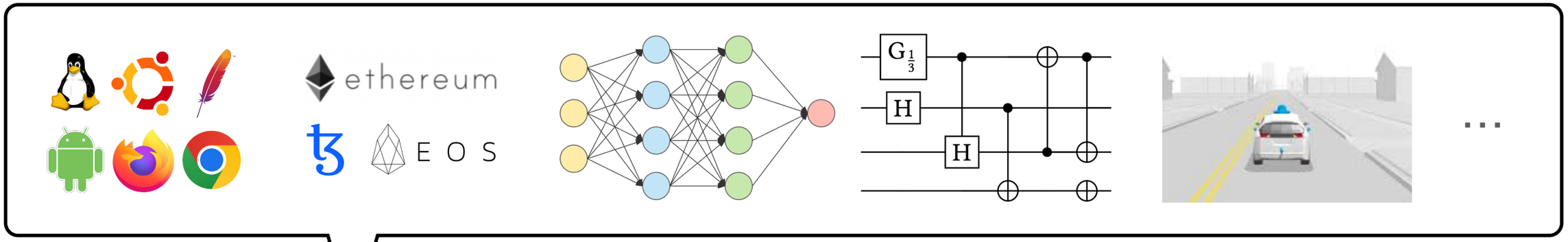


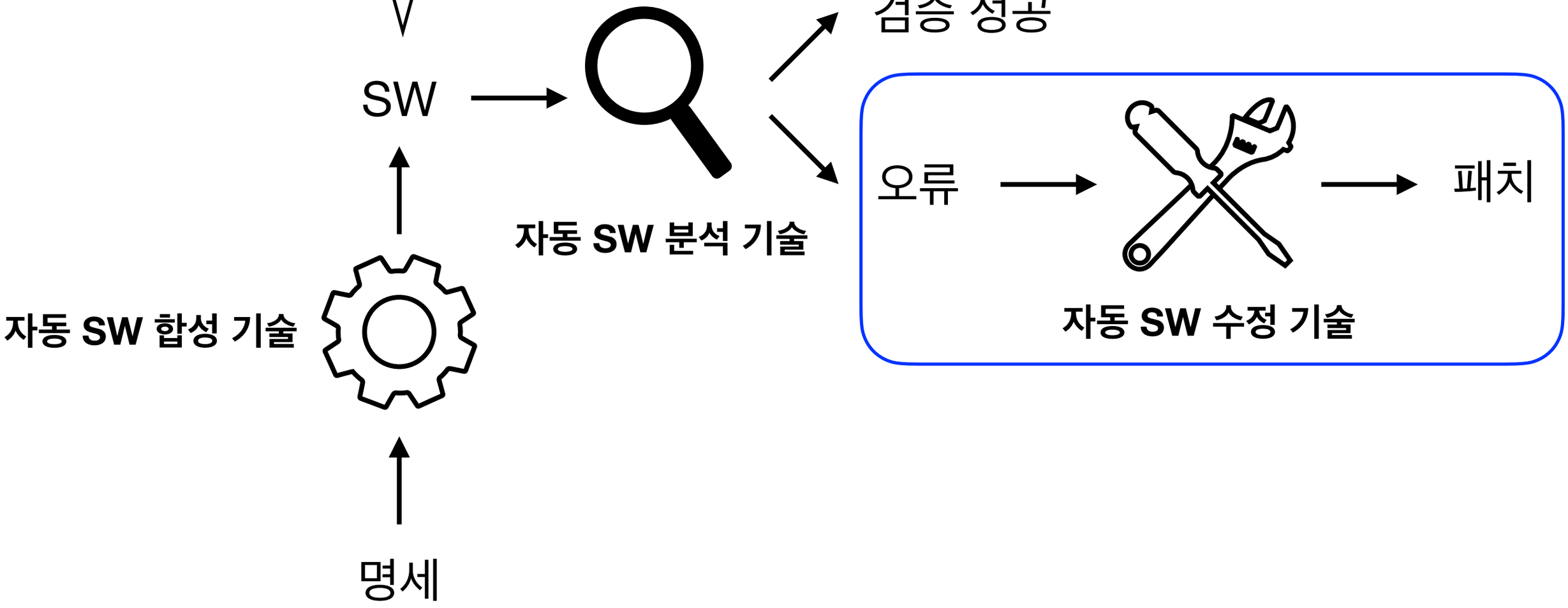
데이터 기반 프로그램 분석



- Automatic: little reliance on analysis designers
- Powerful: machine-tuning outperforms hand-tuning

Our Research



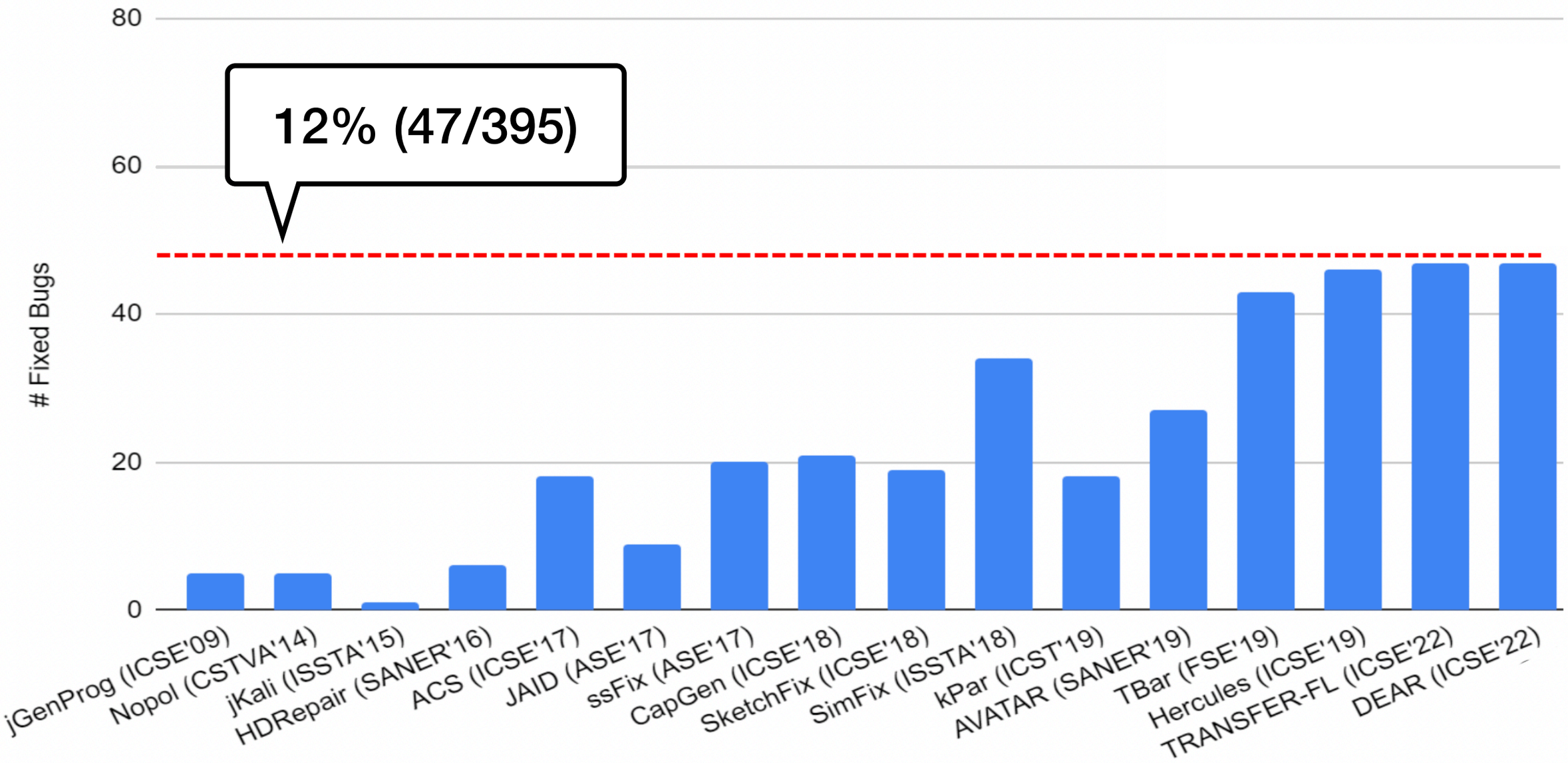


프로그램 자동 수정 연구

- 도메인 특화 기술
  - C/C++ 메모리 오류 자동 수정 [FSE 2018, ICSE 2020]
  - Java 널 포인터 오류 자동 수정 [ICSE 2022]
  - Python 타입 오류 자동 수정 [FSE 2022]
  - Solidity 보안 취약점 자동 수정 (in progress)
  - OCaml 프로그래밍 과제 자동 수정 [OOPSLA 2018/2019, FSE 2021]
- 일반 기술 (테스트 케이스 기반)

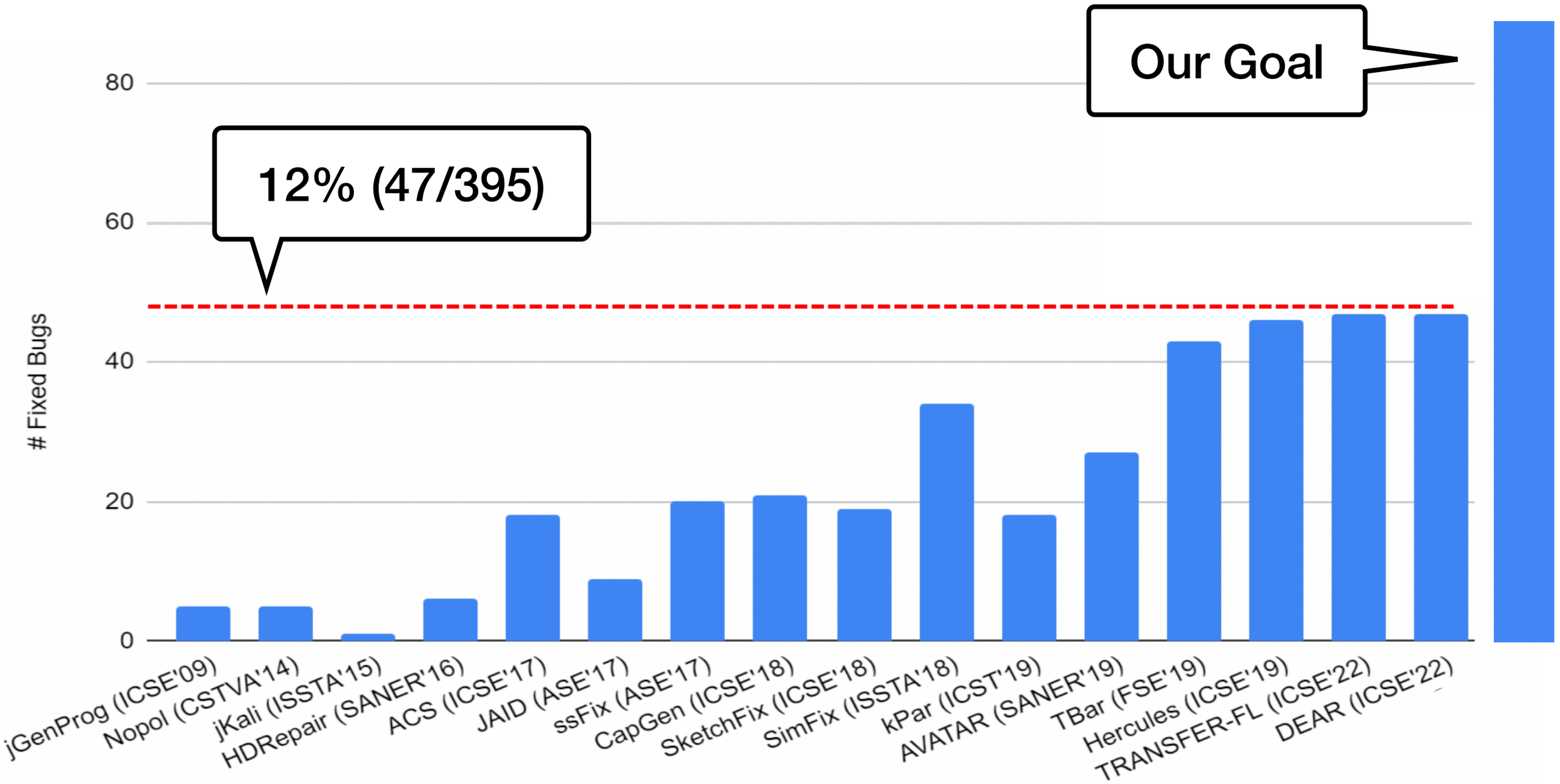
일반적 자동 수정 기술 현황

일반적 자동 수정 기술 현황



Defects4J 395개 오류 대상 자동 수정 성공률

일반적 자동 수정 기술 현황



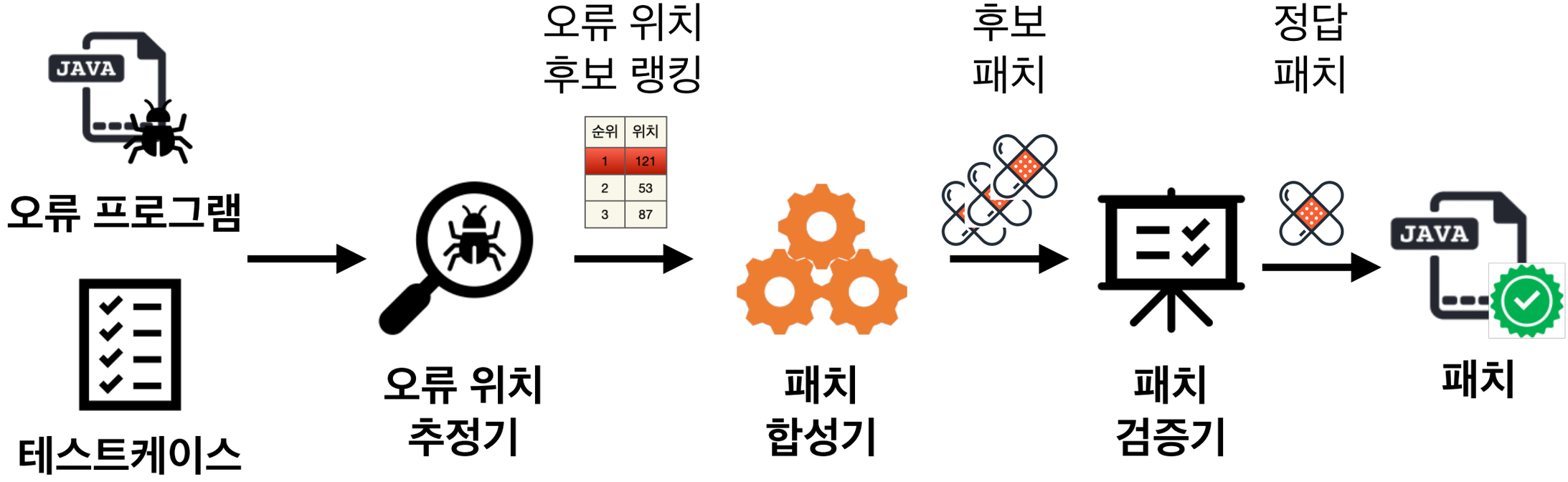
Defects4J 395개 오류 대상 자동 수정 성공률

일반적 자동 수정 기술 현황

Closure-71 (Defects4J) 사례

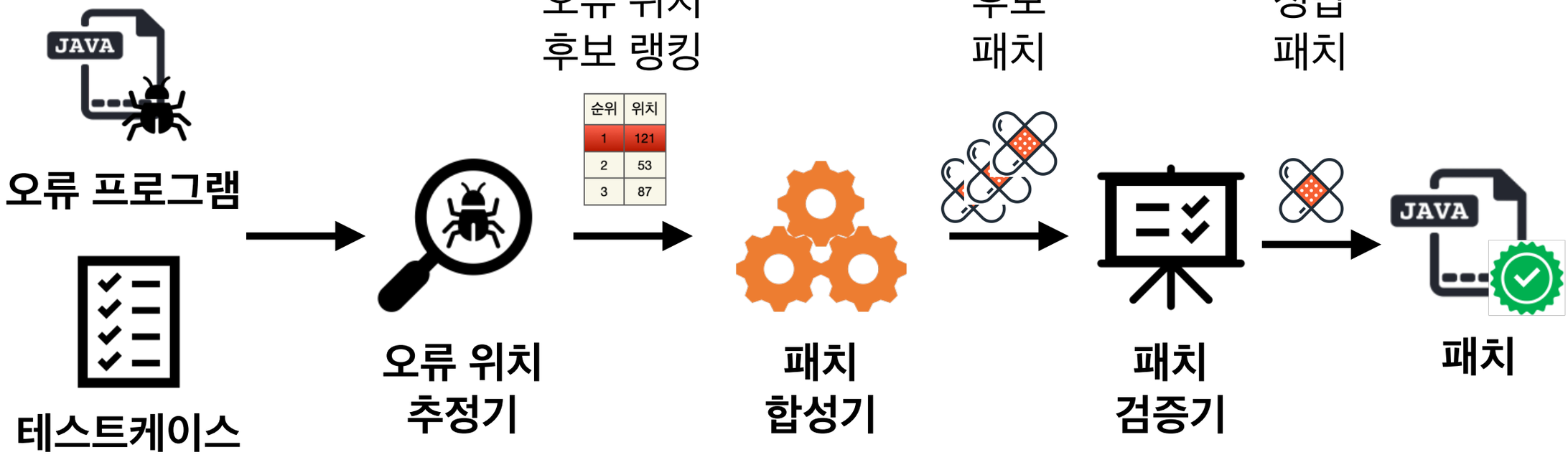
```
    void checkVisibility(Node t, Node parent) {
(-)   boolean isOverride = t.inGlobalScope() && ...
(+)   boolean isOverride = parent.getJSDocInfo() == null && ...

      if (isOverride) {
        compiler.report(PRIVATE_OVERRIDE);
      }
    }
```

일반적 자동 수정 기술 현황



일반적 자동 수정 기술 현황

68위 X 1700만개 X 1초 = 36년



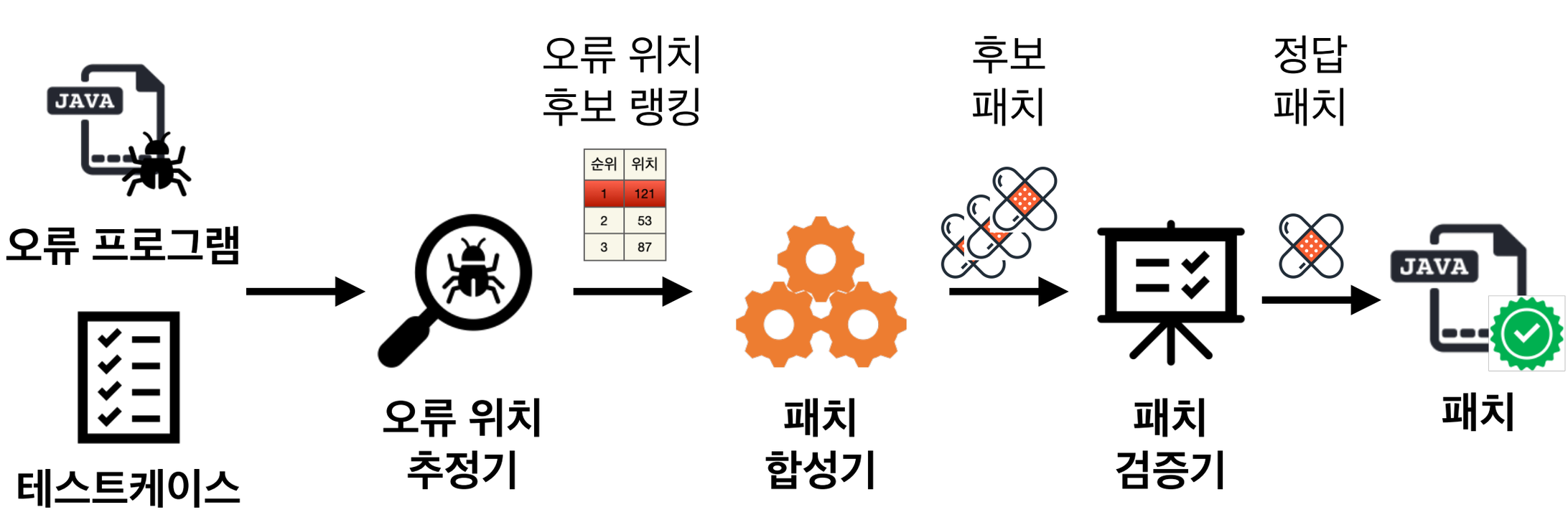
일반적 자동 수정 기술 현황

Our Goal

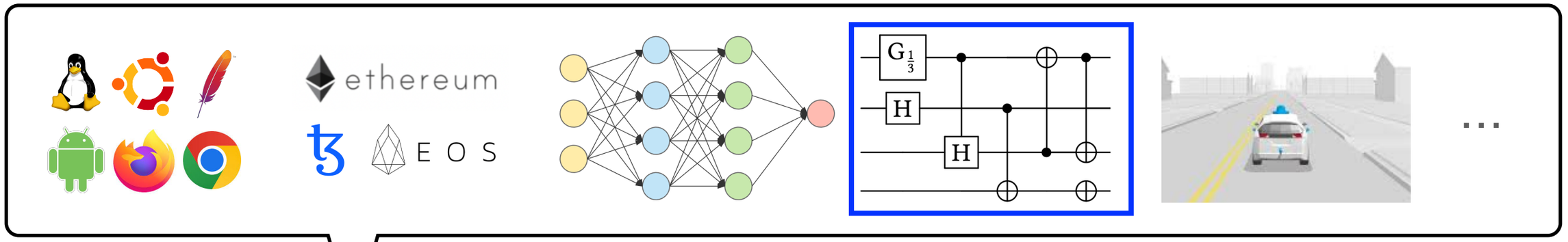
10위 X 10000개 X 0.1초 = 3시간

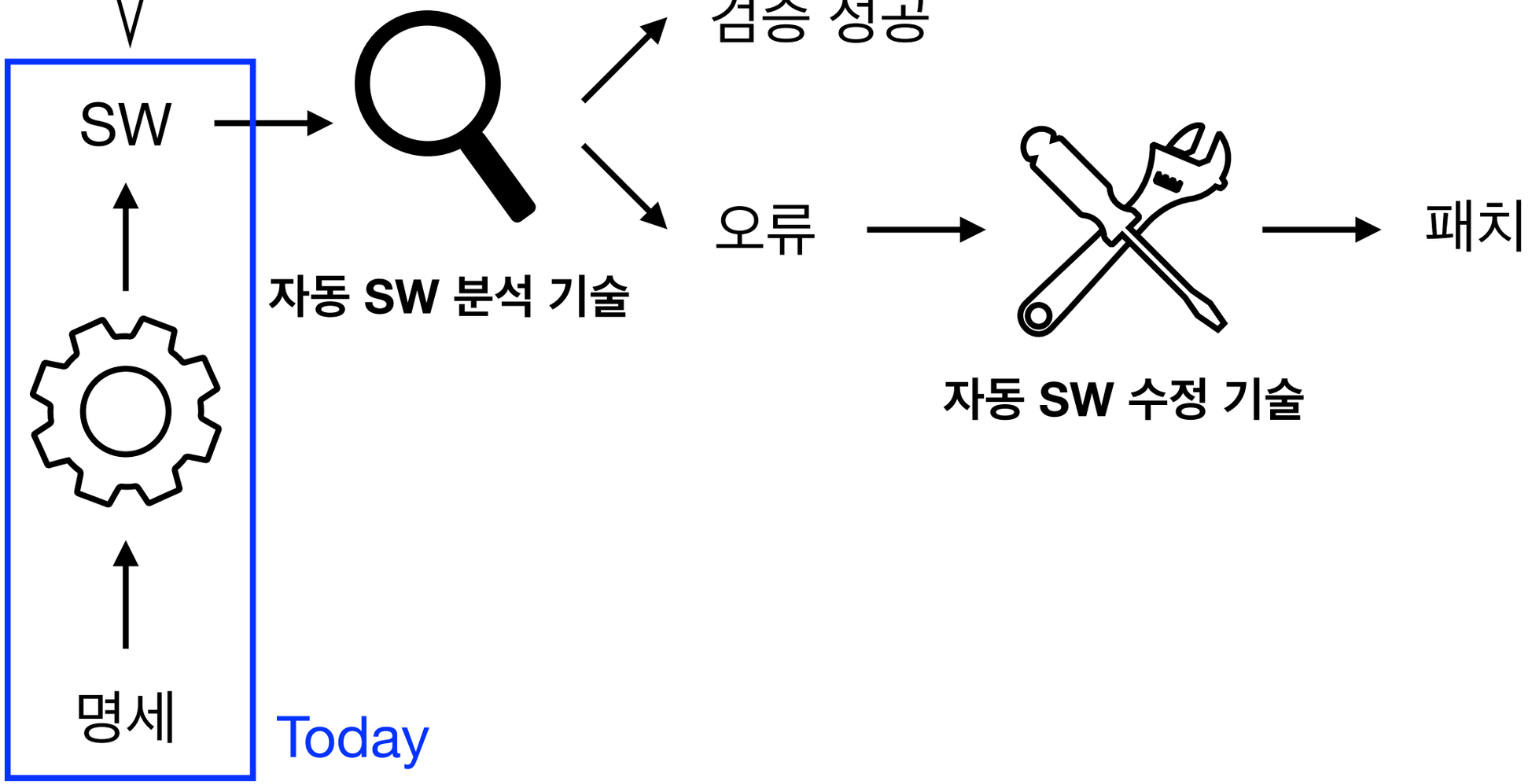
위치 추정 정밀화 | 테스트 통과 가능성 분석 | 의미 기반 패치 검증

68위 X 1700만개 X 1초 = 36년



Our Research





양자 회로 자동 합성 (OOPSLA 2023)

입출력 명세

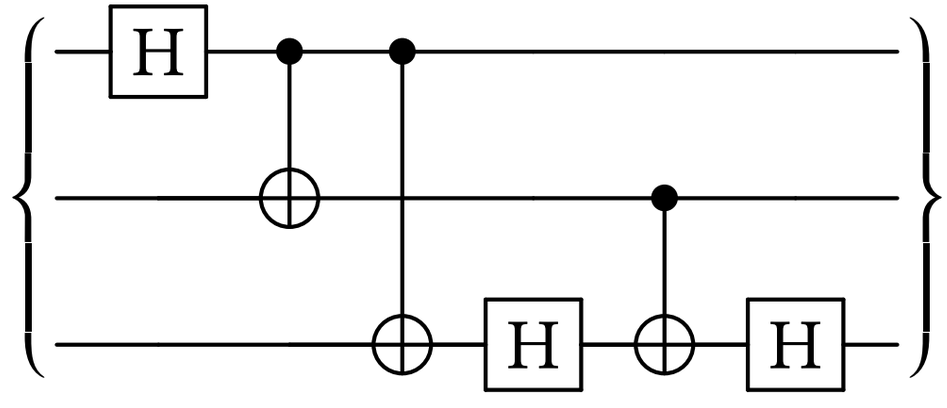
$$|100\rangle \mapsto \frac{1}{\sqrt{2}}(|000\rangle + |111\rangle)$$

컴포넌트 게이트

$$H, CNOT$$

양자 회로 합성기

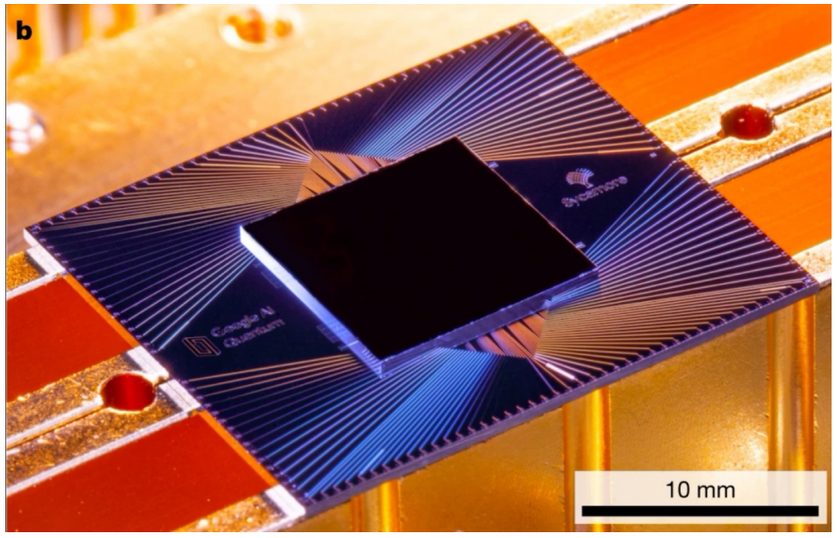
$|100\rangle$



$$|GHZ\rangle = \frac{1}{\sqrt{2}}(|000\rangle + |111\rangle)$$

양자 컴퓨터

- 하드웨어



Google Sycamore (2019)

IBM Hummingbird, Eagle, Osprey (2020-2022)

- 소프트웨어

Qiskit Cirq Q# SILQ

양자 컴퓨터

- 고전 컴퓨터의 일반화

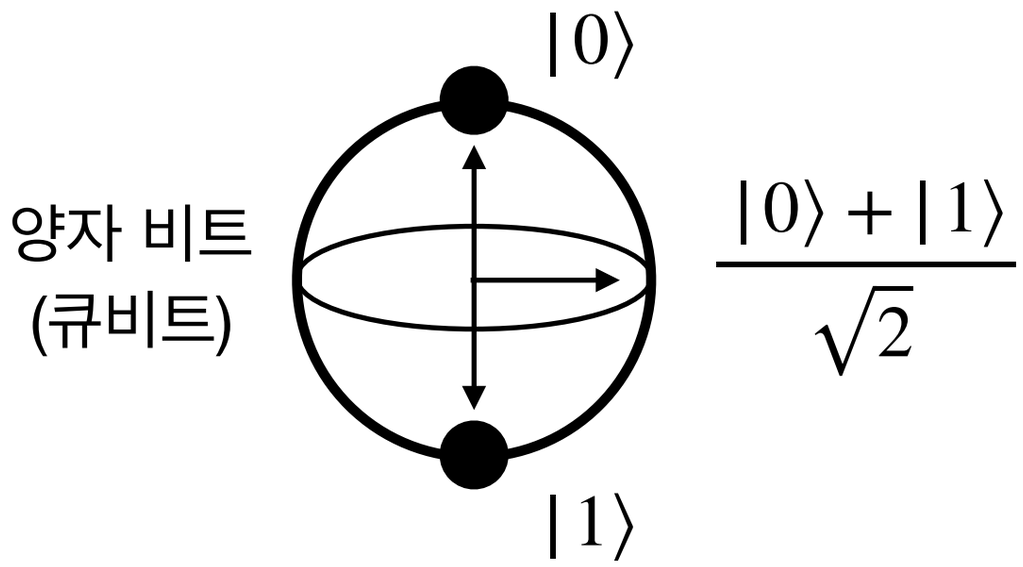
양자 컴퓨터

- 고전 컴퓨터의 일반화

고전 비트

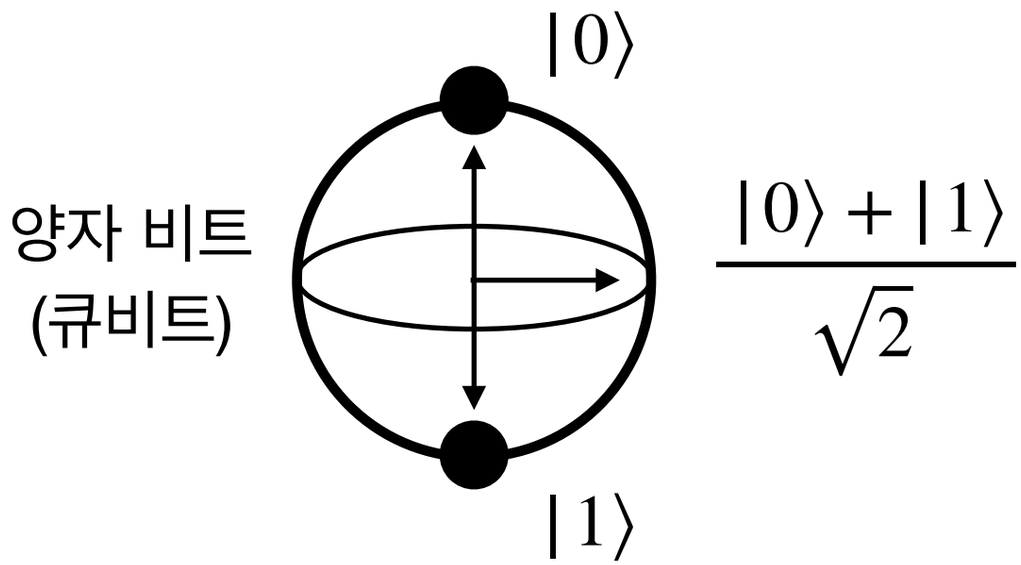
양자 컴퓨터

- 고전 컴퓨터의 일반화

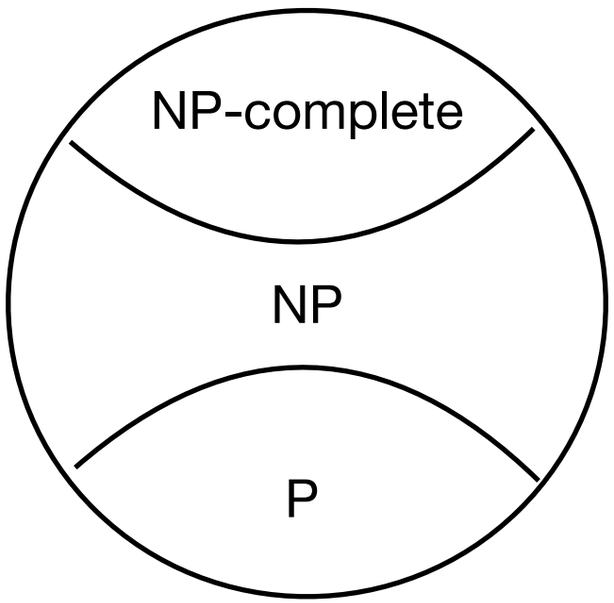


양자 컴퓨터

- 고전 컴퓨터의 일반화

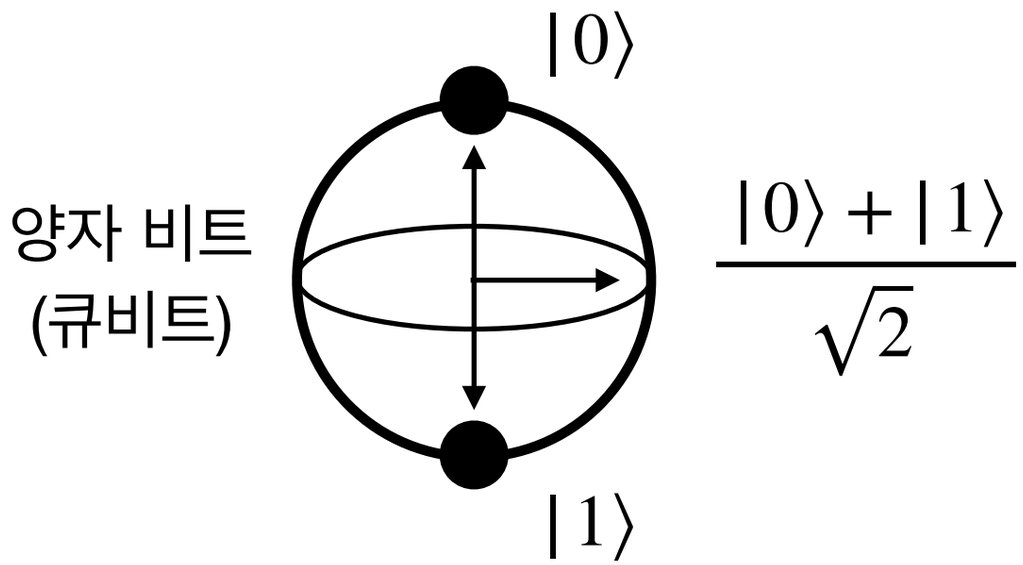


- 가능성 & 한계

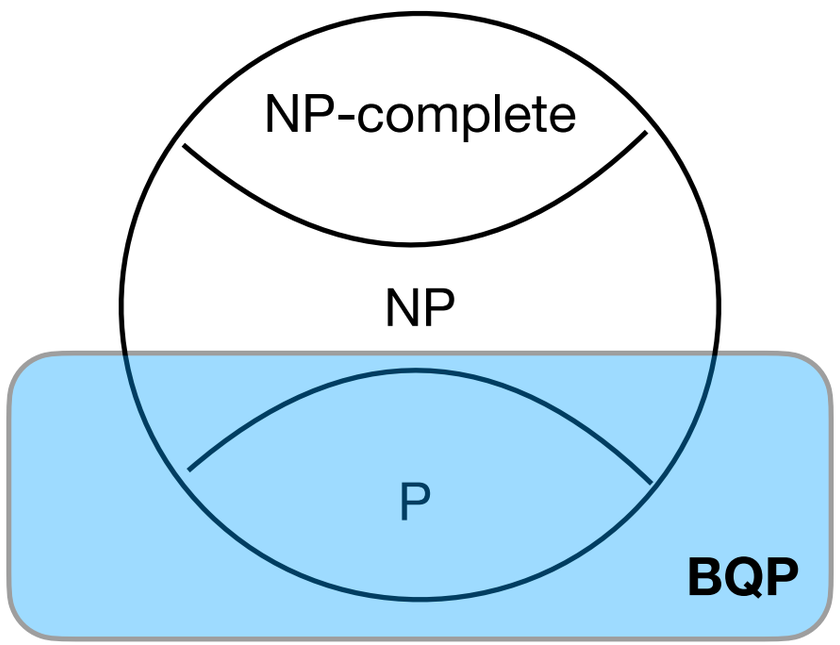


양자 컴퓨터

- 고전 컴퓨터의 일반화

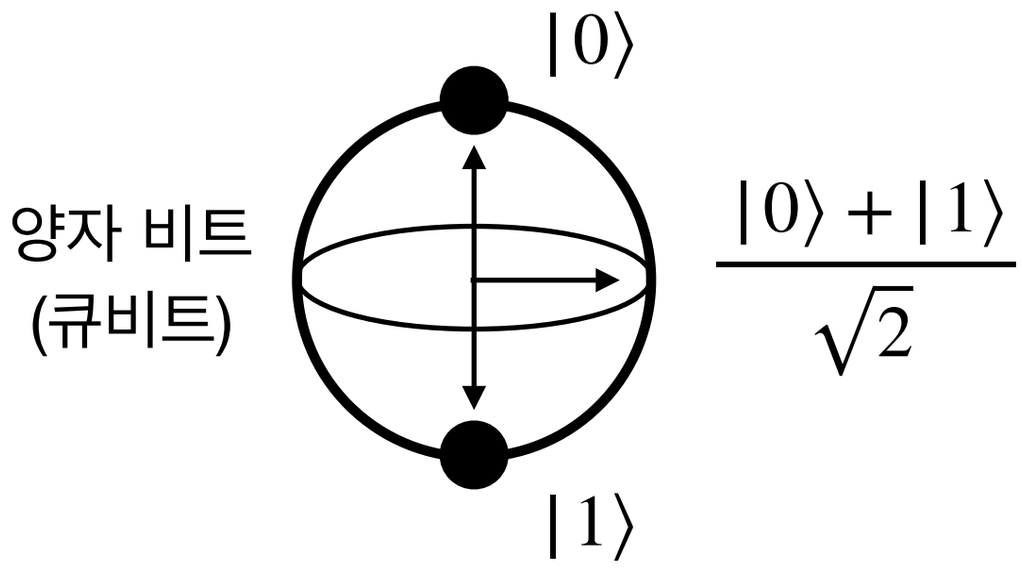


- 가능성 & 한계

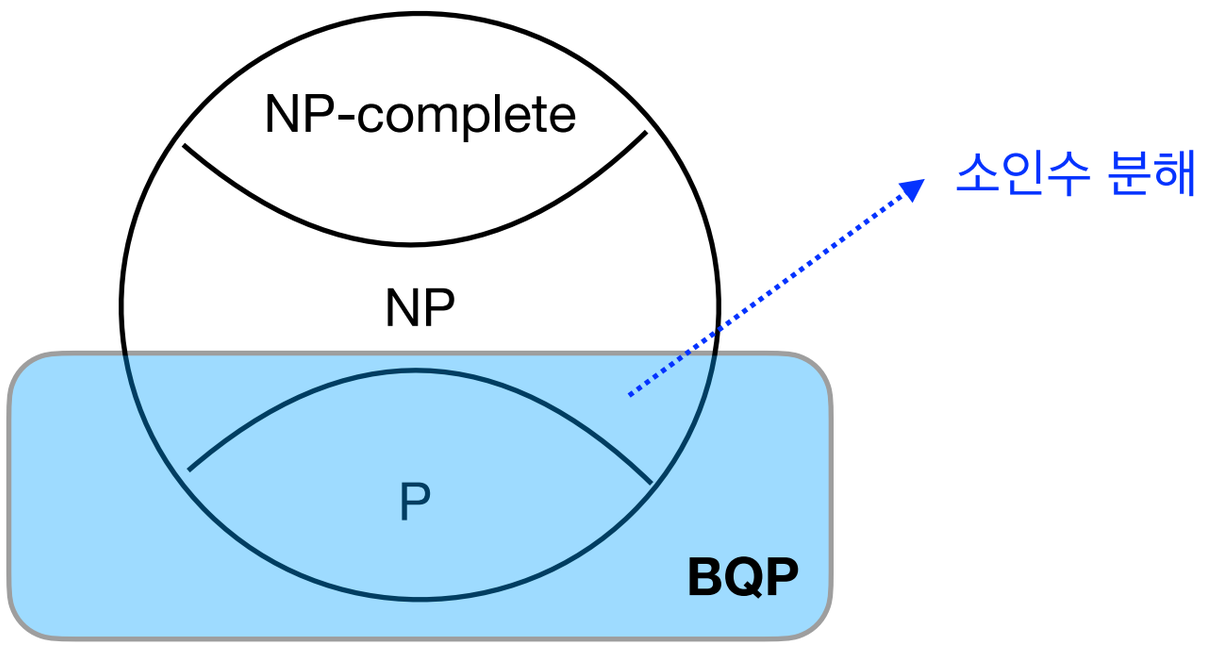


양자 컴퓨터

- 고전 컴퓨터의 일반화

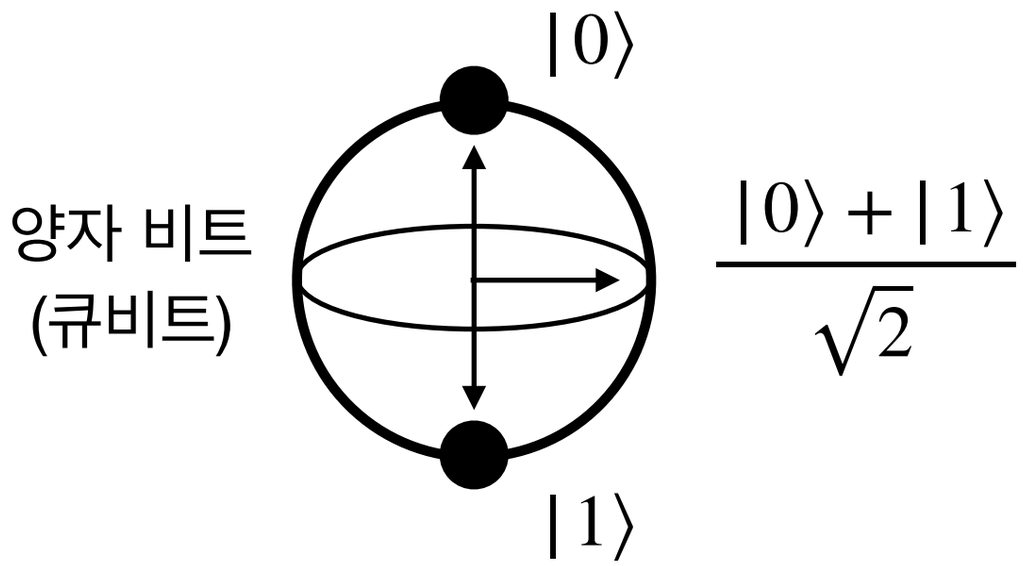


- 가능성 & 한계

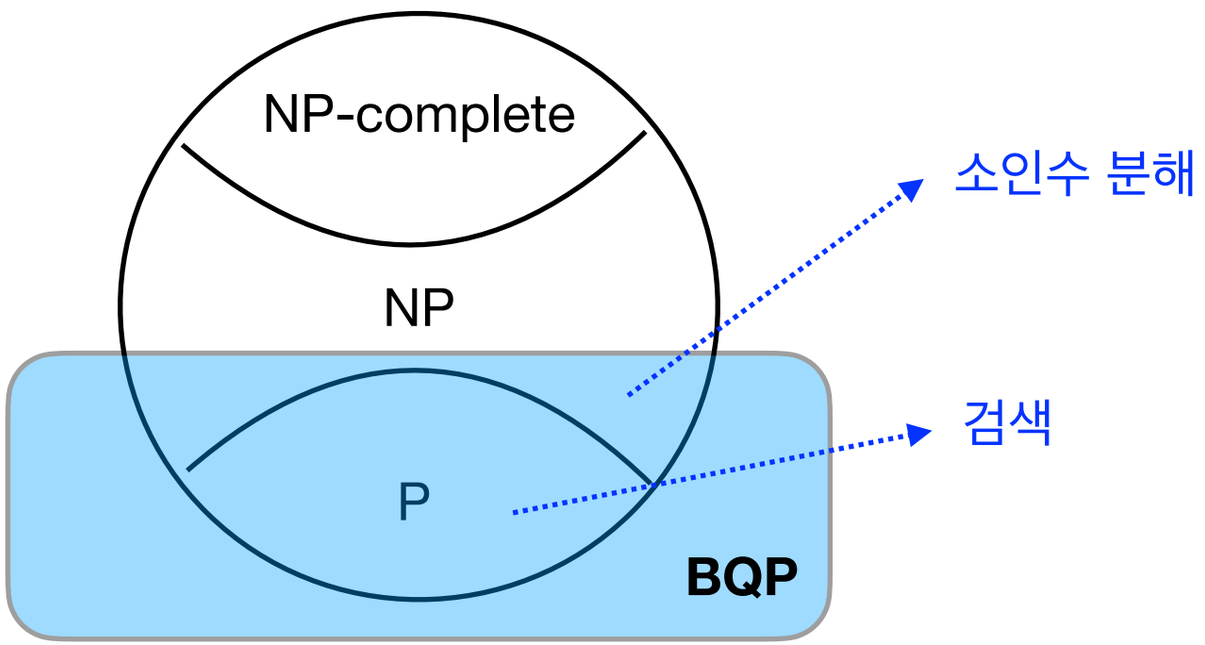


양자 컴퓨터

- 고전 컴퓨터의 일반화

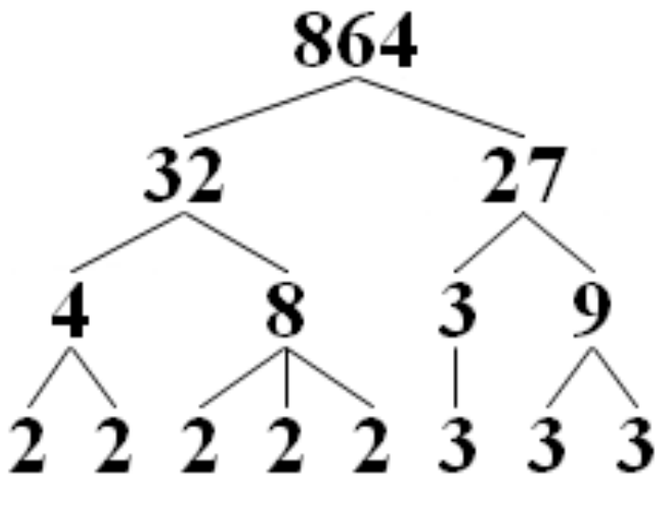


- 가능성 & 한계



적용 사례

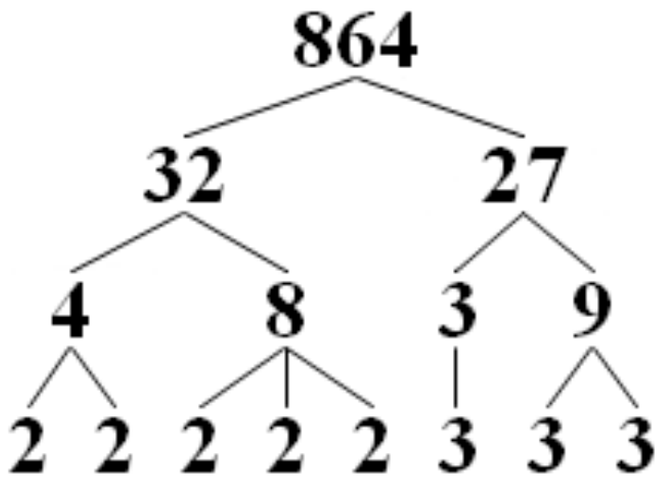
(1) 소인수분해 [Shor 1995]



$O(2^N)$ vs $O(N^3)$

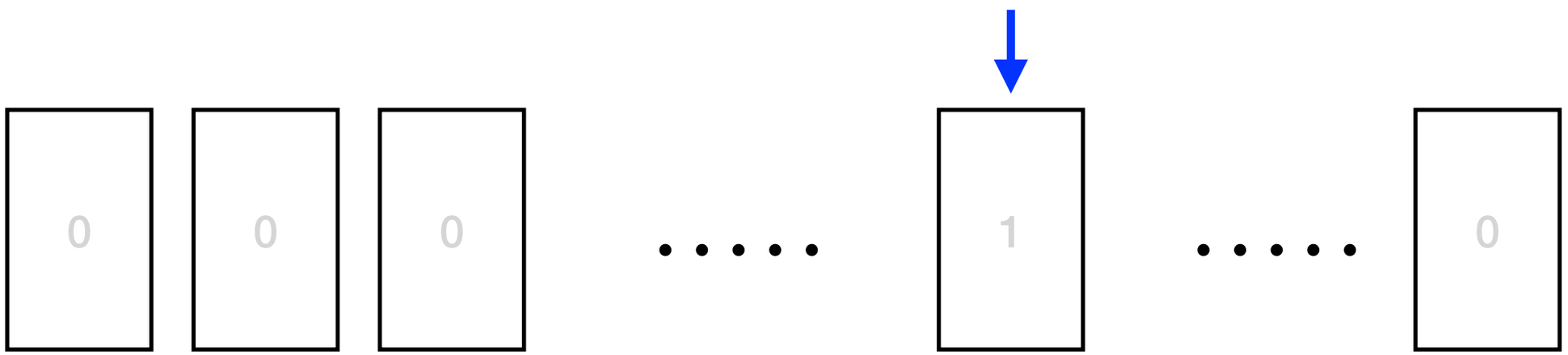
적용 사례

(1) 소인수분해 [Shor 1995]



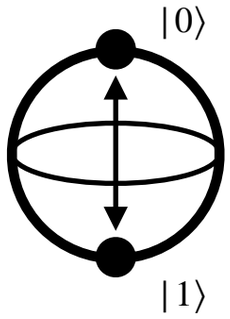
$O(2^N)$ vs $O(N^3)$

(2) 검색 [Grover 1996]



$O(N)$ vs $O(\sqrt{N})$

큐비트



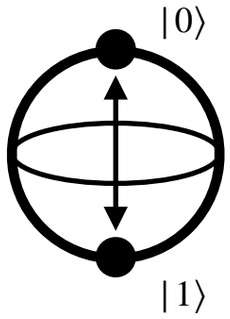
- 비트의 일반화 (0과 1의 중첩—선형 결합)

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

  - $\alpha_1, \alpha_2 \in \mathbb{C}$: 확률 진폭 (probability amplitude)

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

큐비트



- 비트의 일반화 (0과 1의 중첩—선형 결합)

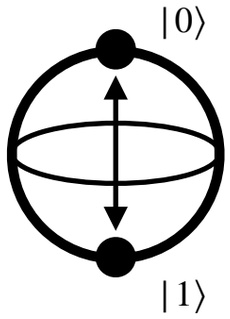
$$|\psi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle$$

  - $\alpha_1, \alpha_2 \in \mathbb{C}$: 확률 진폭 (probability amplitude)

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

  - $|0\rangle$
  - $|1\rangle$
  - $\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$

큐비트



- 비트의 일반화 (0과 1의 중첩—선형 결합)

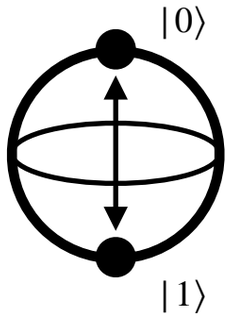
$$|\psi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle$$

  - $\alpha_1, \alpha_2 \in \mathbb{C}$: 확률 진폭 (probability amplitude)

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

  - $|0\rangle$
  - $|1\rangle$
  - $\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$
  - $\frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle$

큐비트



- 비트의 일반화 (0과 1의 중첩—선형 결합)

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

  - $\alpha_1, \alpha_2 \in \mathbb{C}$: 확률 진폭 (probability amplitude)

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

- $|0\rangle$
- $|1\rangle$
- $\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$
- $\frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle$
- $\sqrt{\frac{3}{4}}|0\rangle + \frac{i}{\sqrt{4}}|1\rangle$

N개 큐비트

- 고전 비트 2개로 만들어지는 상태

$$00, \quad 01, \quad 10, \quad 11$$

- 큐비트 2개 = 고전 2-bit 상태들의 중첩

$$|\psi\rangle = \alpha_0|00\rangle + \alpha_1|01\rangle + \alpha_2|10\rangle + \alpha_3|11\rangle \quad \left(\sum_i |\alpha_i|^2 = 1\right)$$

- 큐비트 N개 = 고전 N-bit 상태들의 중첩

$$|\psi\rangle = \sum_{x\in\{0,1\}^N} \alpha_x |x\rangle$$

2^N개 확률 진폭을 “저장”

병렬처리: 고전 컴퓨터 vs. 양자 컴퓨터

- $f: \{0,1\} \rightarrow \{0,1\}$

$$f(0) \qquad \text{vs.} \qquad f(\frac{|0\rangle + |1\rangle}{\sqrt{2}})$$

$$f(1)$$

병렬처리: 고전 컴퓨터 vs. 양자 컴퓨터

- $f: \{0,1\} \rightarrow \{0,1\}$

$$f(0) \qquad\qquad f(\frac{|0\rangle + |1\rangle}{\sqrt{2}})$$
$$f(1) \qquad \text{vs.}$$

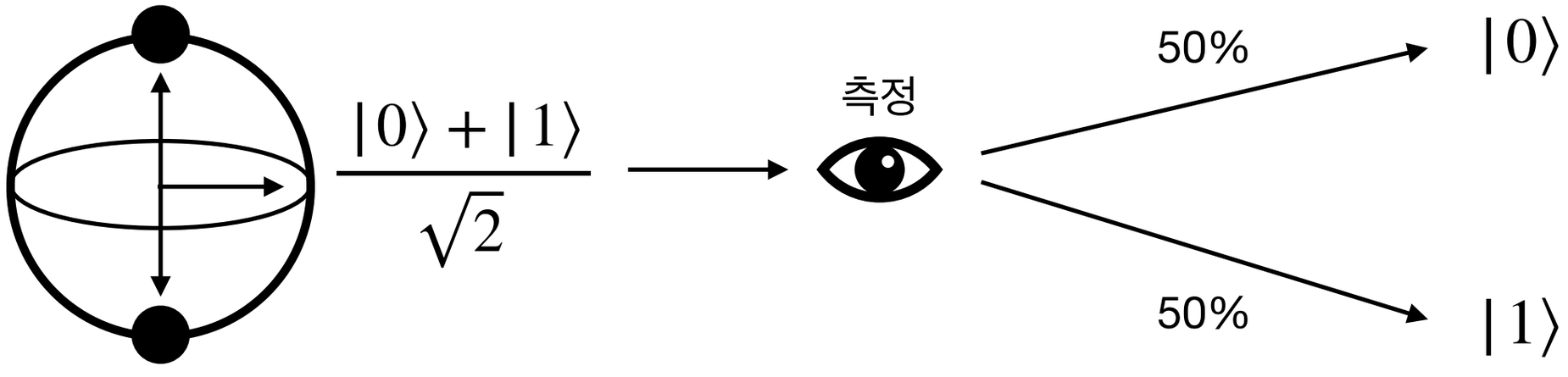
- $f: \{0,1\}^N \rightarrow \{0,1\}$

$$f(0)$$
$$f(1) \qquad \text{vs.} \qquad f(\frac{|0\rangle + |1\rangle + \ldots + |2^N - 1\rangle}{\sqrt{2^N}})$$
$$\vdots$$
$$f(2^N - 1)$$

측정과 붕괴

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

- 큐비트의 내부 상태(α_1, α_2)는 알 수 없고, 고전 정보만 관측 가능
- 큐비트 $|\psi\rangle$를 측정하면,
  - $|\alpha_0|^2$의 확률로 0이 관측되고 $|\psi\rangle = |0\rangle$ 로 붕괴
  - $|\alpha_1|^2$의 확률로 1이 관측되고 $|\psi\rangle = |1\rangle$ 로 붕괴



N개 큐비트 측정

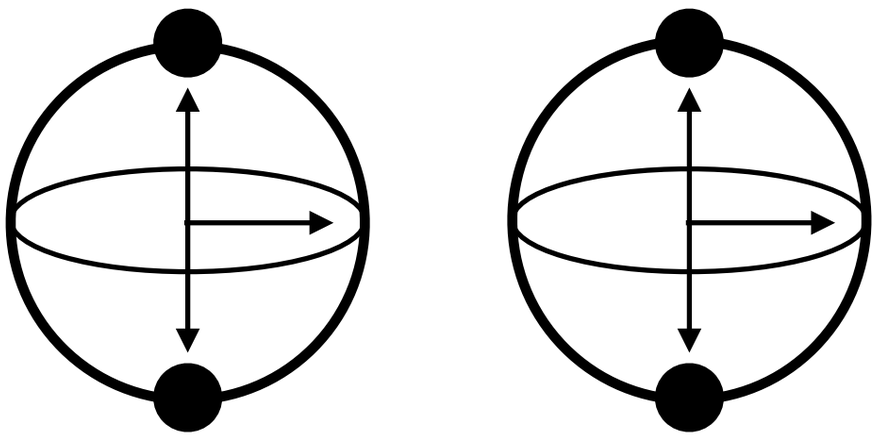
$$|\psi\rangle = \alpha_0|00\rangle + \alpha_1|01\rangle + \alpha_2|10\rangle + \alpha_3|11\rangle$$

- 상태 $|\psi\rangle$를 측정하면,
  - $|\alpha_0|^2$의 확률로 00이 관측되고 $|\psi\rangle = |00\rangle$ 로 붕괴
  - $|\alpha_1|^2$의 확률로 01이 관측되고 $|\psi\rangle = |01\rangle$ 로 붕괴
  - $|\alpha_2|^2$의 확률로 10이 관측되고 $|\psi\rangle = |10\rangle$ 로 붕괴
  - $|\alpha_2|^2$의 확률로 11이 관측되고 $|\psi\rangle = |11\rangle$ 로 붕괴
- 예:

$$|\psi\rangle = \frac{|00\rangle + |01\rangle + |10\rangle + |11\rangle}{2}$$

얽힘 (Entanglement)

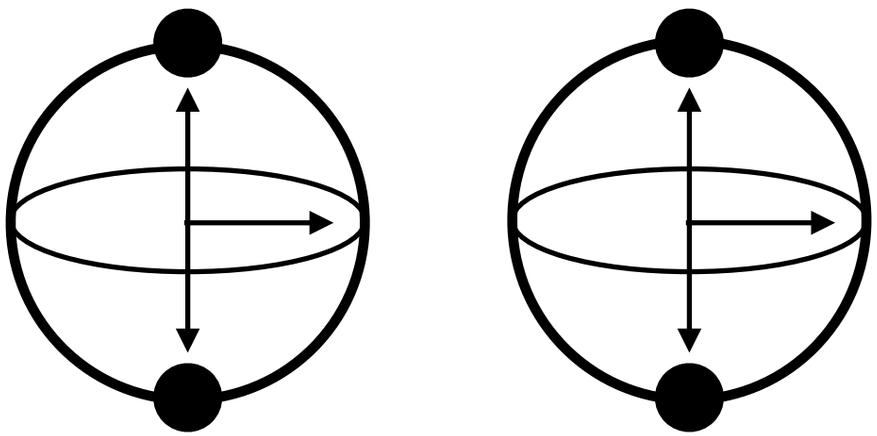
- 얽히지 않은 상태 = 개별 큐비트들이 단순 결합된 상태



$$\frac{|0\rangle + |1\rangle}{\sqrt{2}} \qquad \frac{|0\rangle + |1\rangle}{\sqrt{2}}$$

얽힘 (Entanglement)

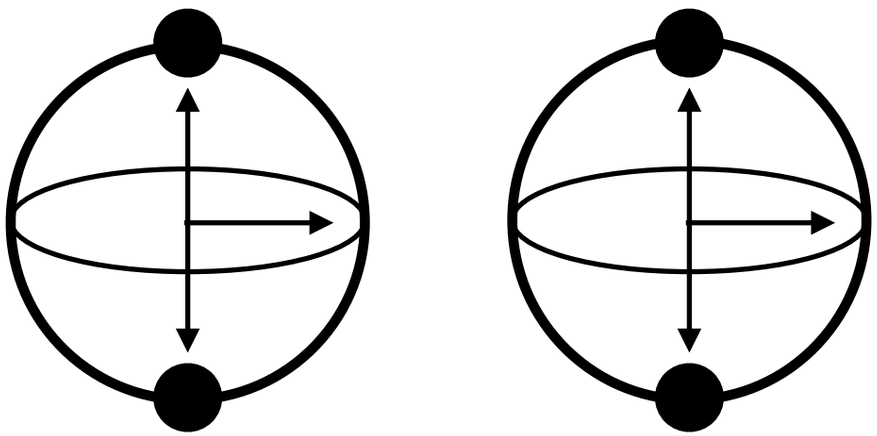
- 얽히지 않은 상태 = 개별 큐비트들이 단순 결합된 상태



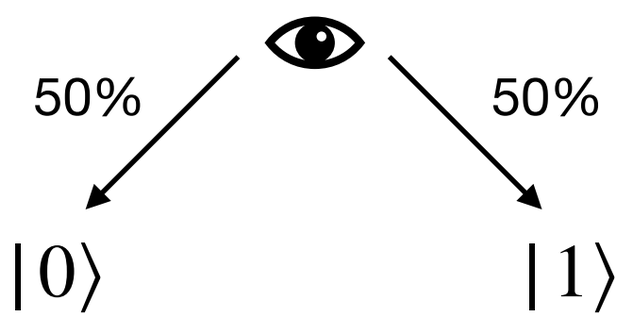
$$\frac{|0\rangle + |1\rangle}{\sqrt{2}} \quad \text{x} \quad \frac{|0\rangle + |1\rangle}{\sqrt{2}} \quad = \quad \frac{|00\rangle + |01\rangle + |10\rangle + |11\rangle}{2}$$

얽힘 (Entanglement)

- 얽히지 않은 상태 = 개별 큐비트들이 단순 결합된 상태



$$\frac{|0\rangle + |1\rangle}{\sqrt{2}} \quad x \quad \frac{|0\rangle + |1\rangle}{\sqrt{2}} \quad = \quad \frac{|00\rangle + |01\rangle + |10\rangle + |11\rangle}{2}$$

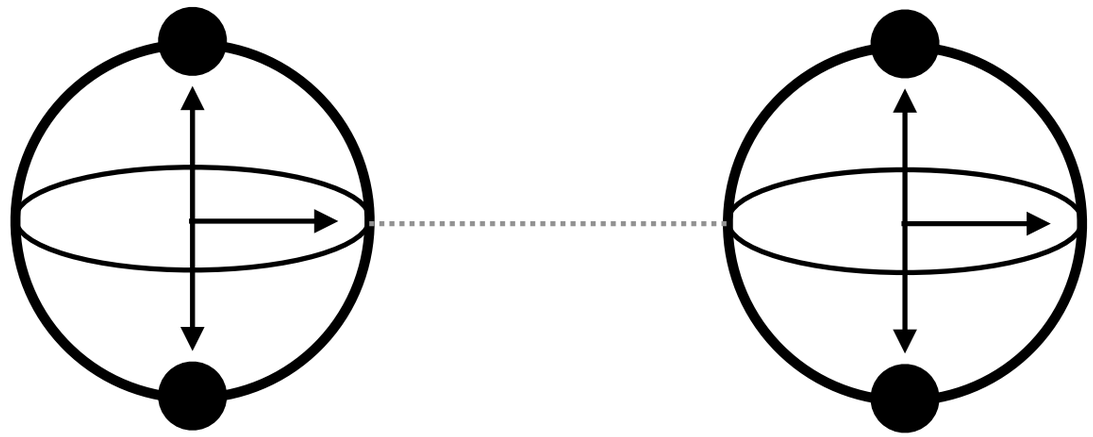


- 하나의 큐비트 관찰 결과가 다른 큐비트 상태에 영향을 주지 않음

얽힘 (Entanglement)

- 얽힌 상태 = 개별 큐비트 상태들로 분리 불가능

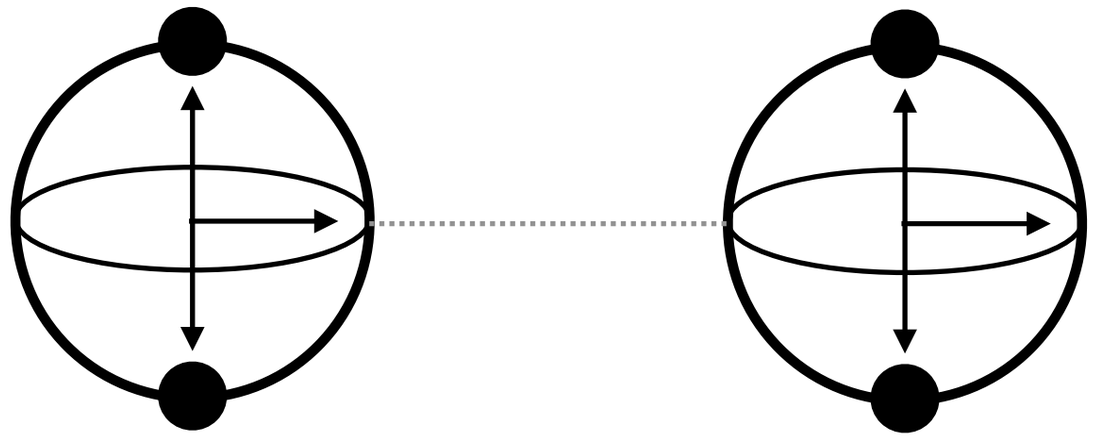
$$\frac{|00\rangle + |11\rangle}{\sqrt{2}}$$



얽힘 (Entanglement)

- 얽힌 상태 = 개별 큐비트 상태들로 분리 불가능

$$\frac{|00\rangle + |11\rangle}{\sqrt{2}}$$

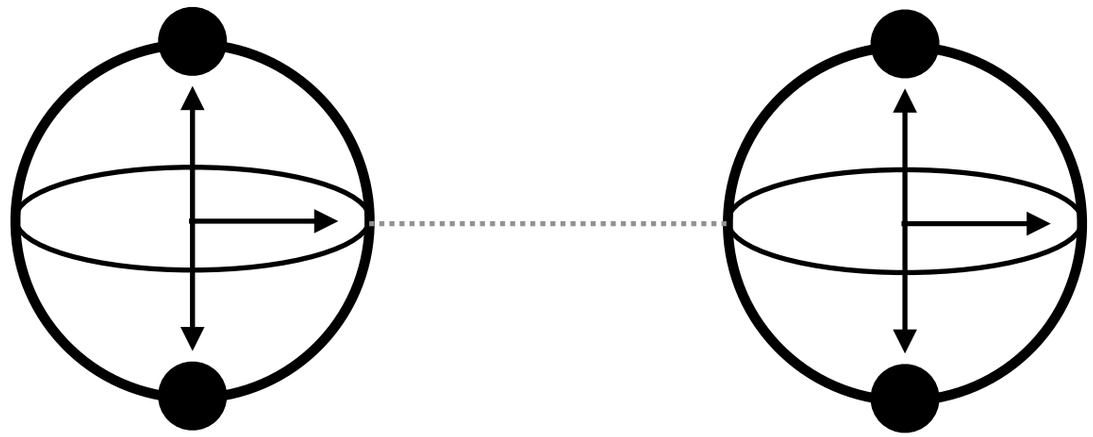


$$\neq (\alpha_0 |0\rangle + \alpha_1 |1\rangle) \times (\beta_0 |0\rangle + \beta_1 |1\rangle)$$

얽힘 (Entanglement)

- 얽힌 상태 = 개별 큐비트 상태들로 분리 불가능

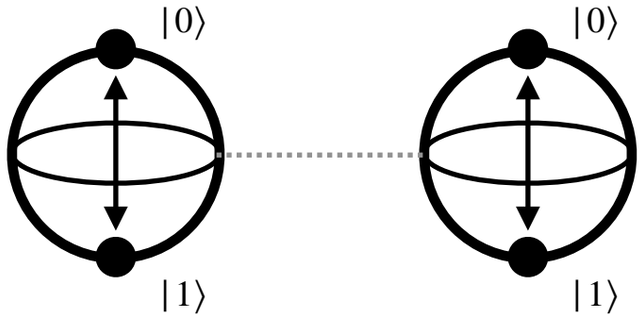
$$\frac{|00\rangle + |11\rangle}{\sqrt{2}}$$



$$\neq (\alpha_0|0\rangle + \alpha_1|1\rangle) \times (\beta_0|0\rangle + \beta_1|1\rangle)$$

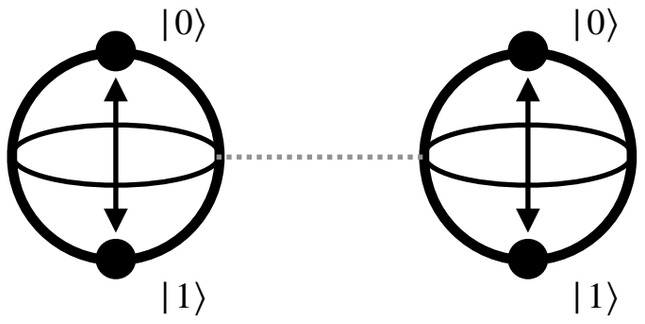
- 하나의 큐비트 관찰이 다른 큐비트 상태에 영향을 줌
  - 하나의 큐비트에서 0을 관측하면 다른 큐비트 상태는 $|0\rangle$로 결정
  - 하나의 큐비트에서 1을 관측하면 다른 큐비트 상태는 $|1\rangle$로 결정

EPR 패러독스

1. 두 큐비트 

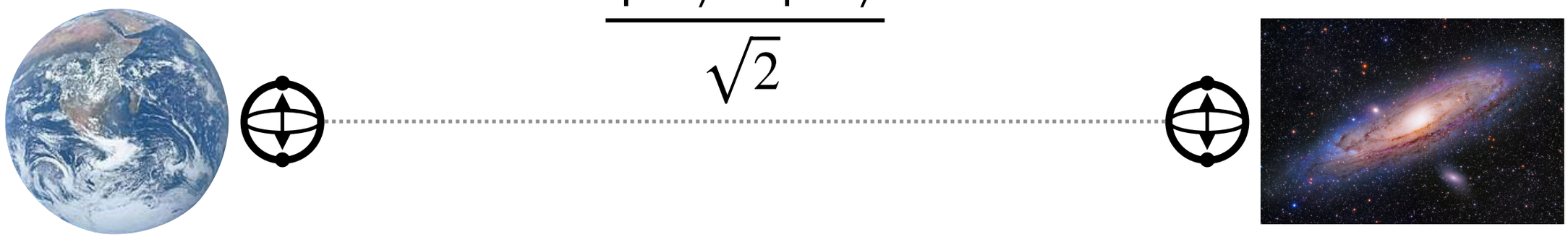
를 $\frac{|00\rangle + |11\rangle}{\sqrt{2}}$ 상태로 준비

EPR 패러독스

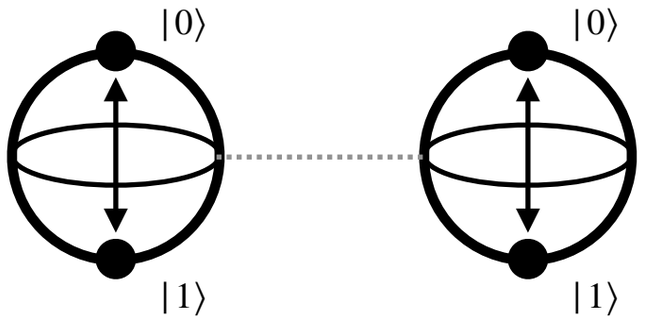
1. 두 큐비트 

를 $\frac{|00\rangle + |11\rangle}{\sqrt{2}}$ 상태로 준비

2. 두 큐비트를 서로 멀리 떨어지도록 이동

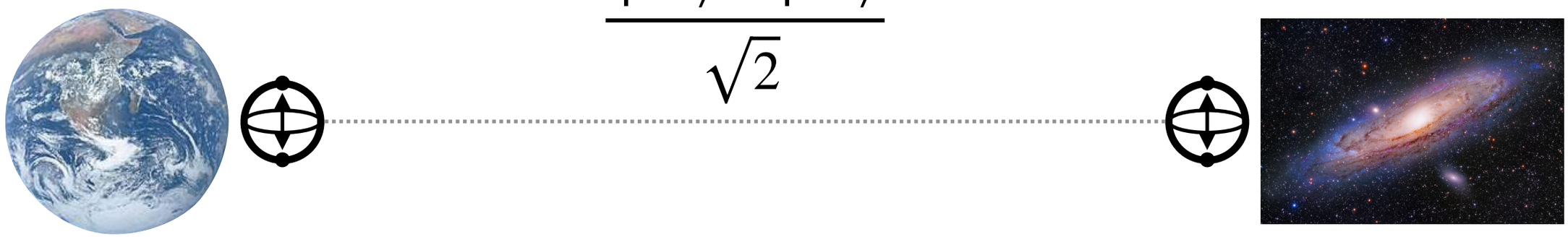


EPR 패러독스

1. 두 큐비트 

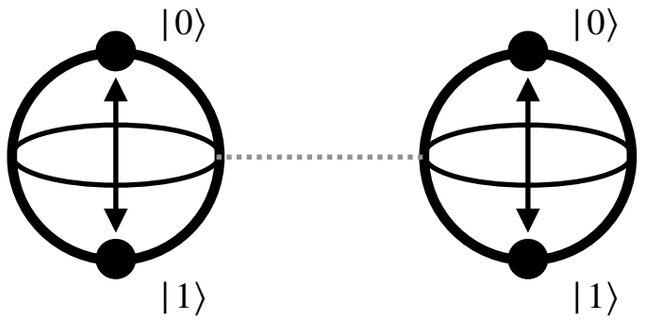
를 $\frac{|00\rangle + |11\rangle}{\sqrt{2}}$ 상태로 준비

2. 두 큐비트를 서로 멀리 떨어지도록 이동



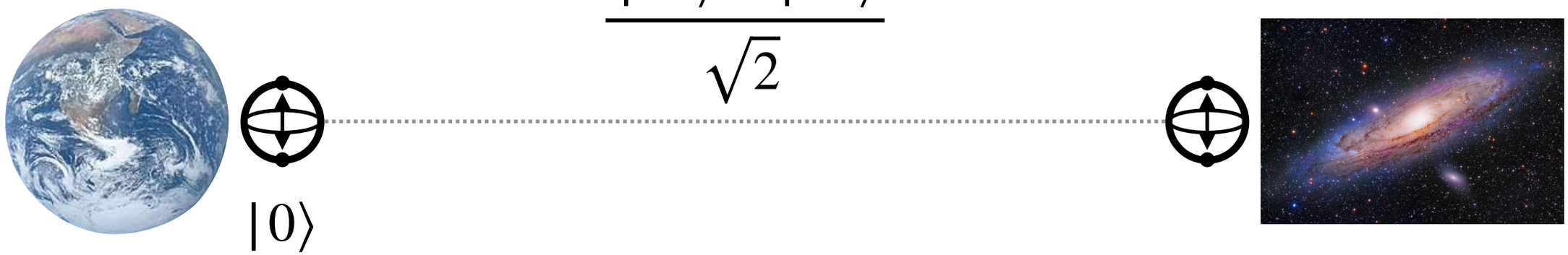
3. 측정

EPR 패러독스

1. 두 큐비트 

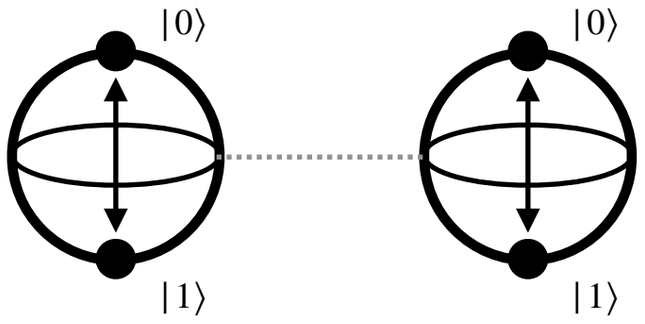
를 $\frac{|00\rangle + |11\rangle}{\sqrt{2}}$ 상태로 준비

2. 두 큐비트를 서로 멀리 떨어지도록 이동



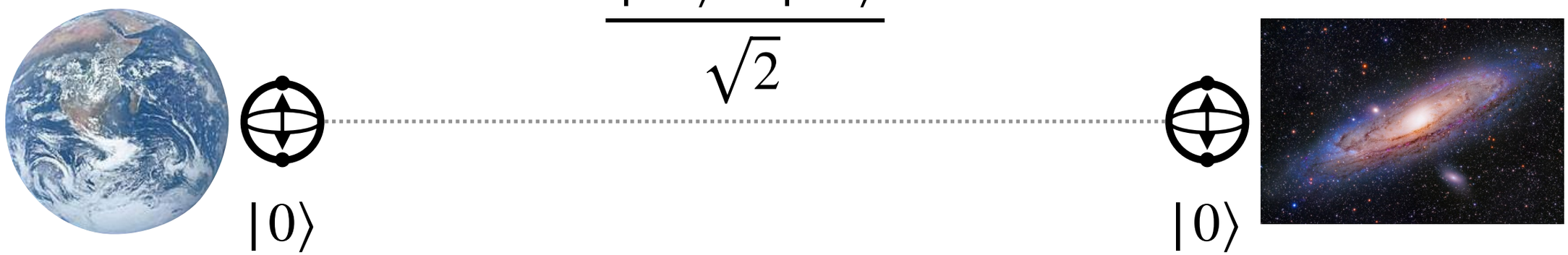
3. 측정

EPR 패러독스

1. 두 큐비트 

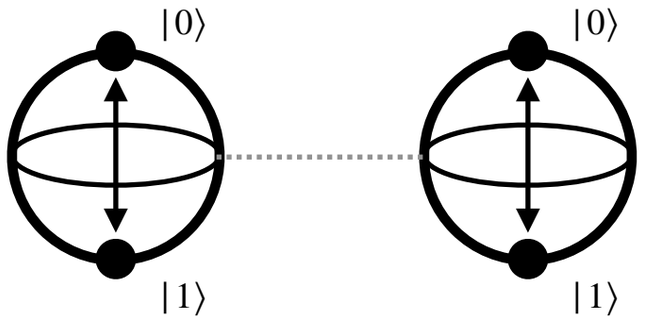
를 $\frac{|00\rangle + |11\rangle}{\sqrt{2}}$ 상태로 준비

2. 두 큐비트를 서로 멀리 떨어지도록 이동



3. 측정

EPR 패러독스

1. 두 큐비트 

를 $\frac{|00\rangle + |11\rangle}{\sqrt{2}}$ 상태로 준비

2. 두 큐비트를 서로 멀리 떨어지도록 이동

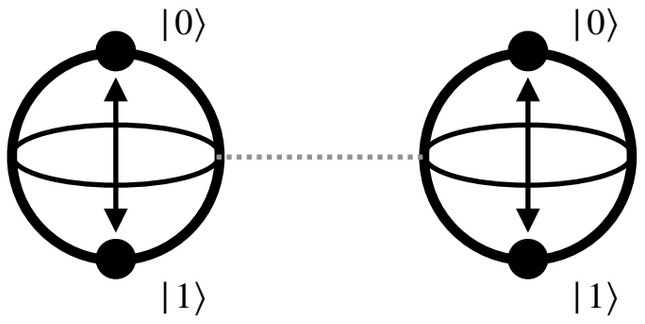
$|0\rangle$
$|1\rangle$

$\frac{|00\rangle + |11\rangle}{\sqrt{2}}$

$|0\rangle$

3. 측정

EPR 패러독스

1. 두 큐비트 

를 $\frac{|00\rangle + |11\rangle}{\sqrt{2}}$ 상태로 준비

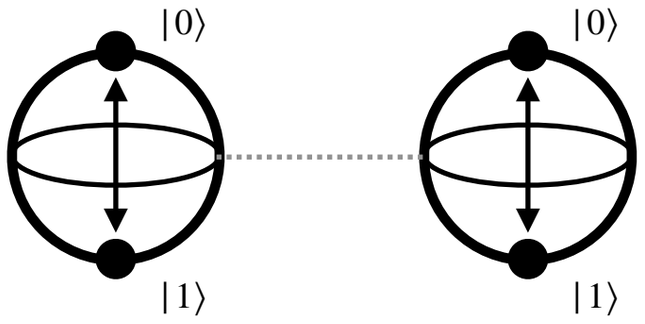
2. 두 큐비트를 서로 멀리 떨어지도록 이동

|0⟩
|1⟩

|0⟩
|1⟩

3. 측정

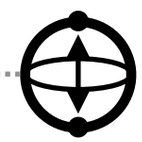
EPR 패러독스

1. 두 큐비트 

를 $\frac{|00\rangle + |11\rangle}{\sqrt{2}}$ 상태로 준비

2. 두 큐비트를 서로 멀리 떨어지도록 이동

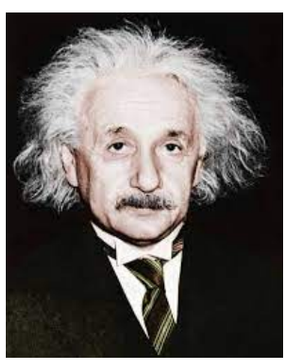
$|0\rangle$
$|1\rangle$



$|0\rangle$
$|1\rangle$

3. 측정

“Spooky action at a distance”



양자 게이트

$$\alpha_0|0\rangle + \alpha_1|1\rangle \quad \boxed{X} \quad \alpha_1|0\rangle + \alpha_0|1\rangle$$

- 큐비트 상태 = 벡터

$$|\psi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle = \alpha_0 \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \alpha_1 \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha_0 \\ \alpha_1 \end{bmatrix}$$

- 게이트 연산 = 행렬 곱

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \times \begin{bmatrix} \alpha_0 \\ \alpha_1 \end{bmatrix} = \begin{bmatrix} \alpha_1 \\ \alpha_0 \end{bmatrix}$$

양자 게이트

$$\alpha_0|0\rangle + \alpha_1|1\rangle \quad \boxed{Z} \quad \alpha_0|0\rangle - \alpha_1|1\rangle$$

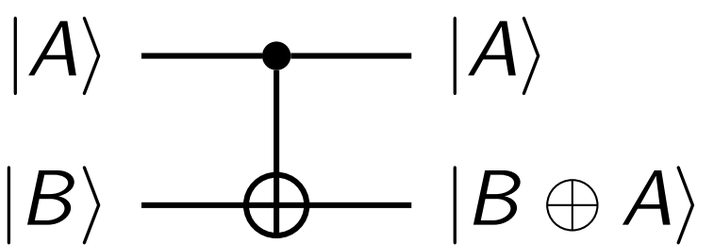
$$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

$$\alpha_0|0\rangle + \alpha_1|1\rangle \quad \boxed{H} \quad \alpha_0\left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right) + \alpha_1\left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right)$$

$$\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

양자 게이트

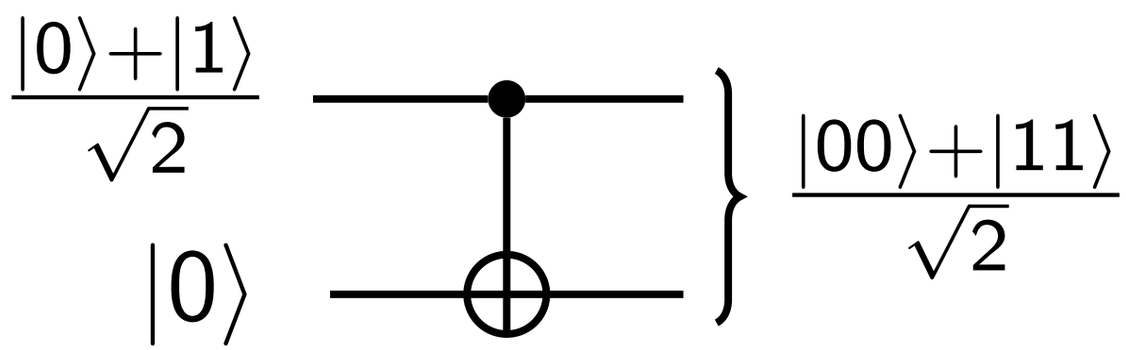
- CNOT (Controlled-Not) 게이트



$$\text{CNOT} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

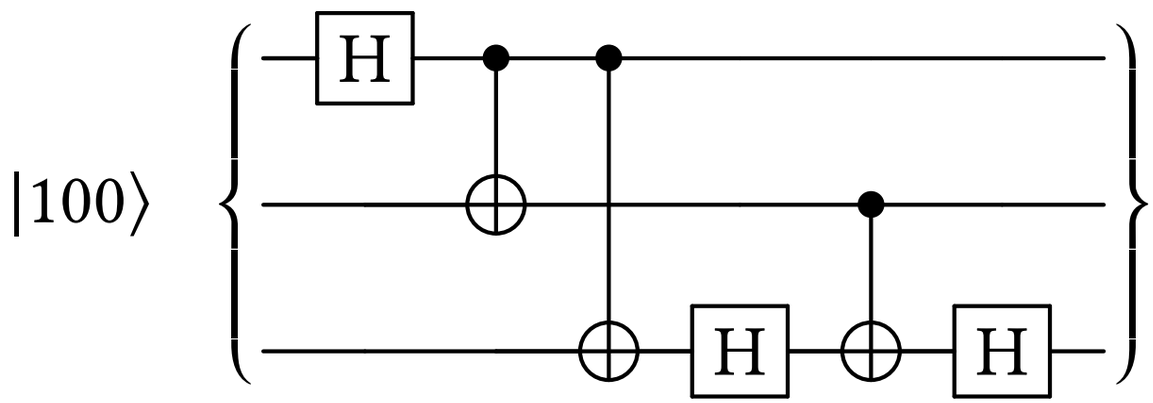
$$\begin{aligned} CNOT(\alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle) \\ = \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|11\rangle + \alpha_{11}|10\rangle \end{aligned}$$

- CNOT: 얽힌 상태를 만드는데 사용



양자 회로

- 양자 게이트들의 조합



$$|GHZ\rangle = \frac{1}{\sqrt{2}}(|000\rangle + |111\rangle)$$

```
circ = QuantumCircuit(3)
circ.h(0)
circ.cx(0, 1)
circ.cx(0, 2)
circ.h(2)
circ.cx(1, 2)
circ.h(2)
```

Qiskit

양자 프로그래밍의 어려움

- 전혀 다른 계산 모델

양자 회로 자동 합성 (OOPSLA 2023)

입출력 명세

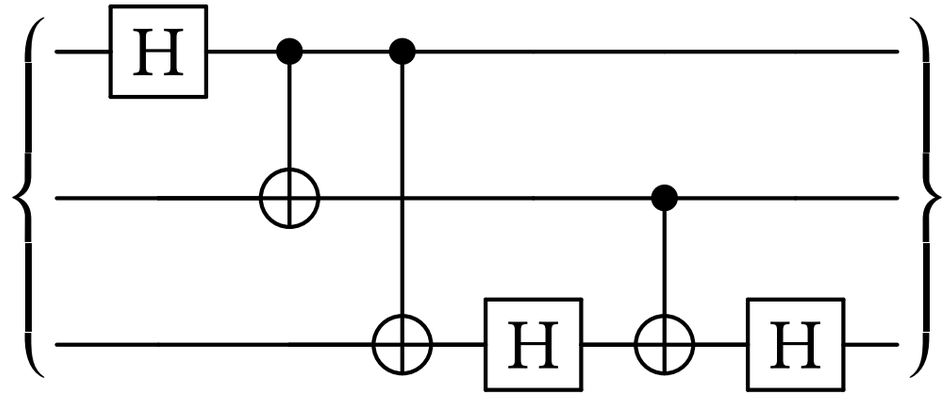
$$|100\rangle \mapsto \frac{1}{\sqrt{2}}(|000\rangle + |111\rangle)$$

컴포넌트 게이트

$$H, CNOT$$

양자 회로 합성기

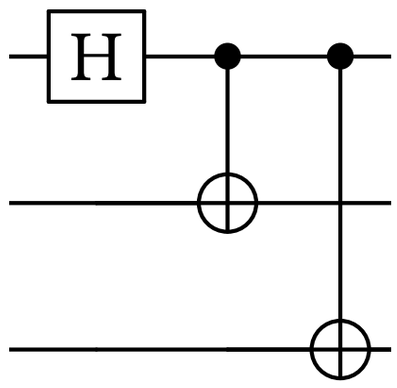
$|100\rangle$

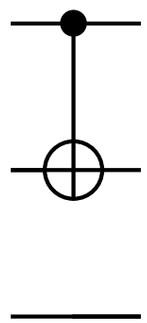


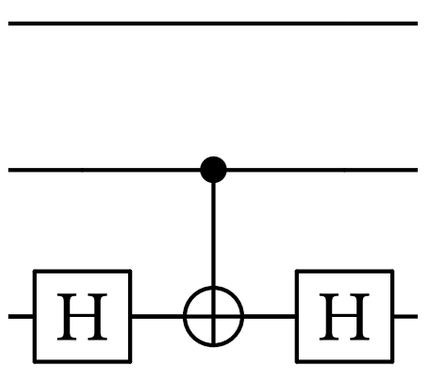
$$|GHZ\rangle = \frac{1}{\sqrt{2}}(|000\rangle + |111\rangle)$$

아이디어: 모듈 레벨 양자 회로 합성

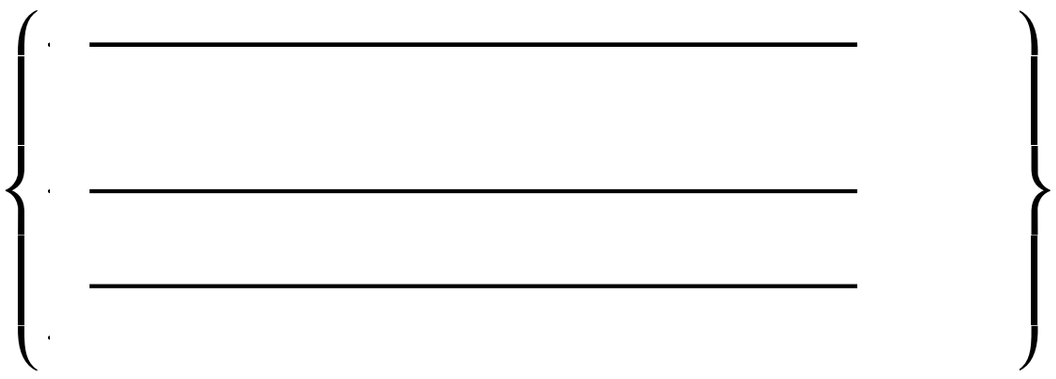
1. 모듈 = 게이트 조합 (up to some fixed length)

$M_1 =$ 

$M_2 =$ 

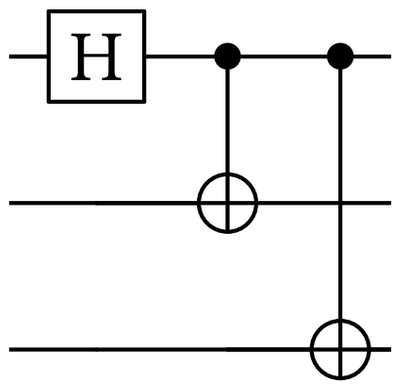
$M_3 =$ 

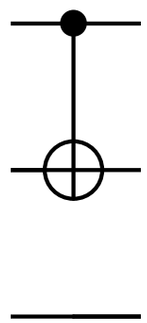
2. 명세와의 특성 차이를 줄이는 방향으로 모듈 쌓기

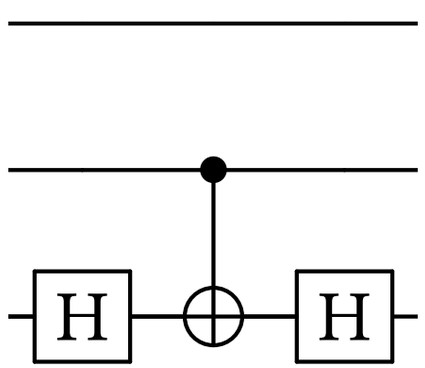
$|100\rangle$  $|GHZ\rangle = \frac{1}{\sqrt{2}}(|000\rangle + |111\rangle)$

아이디어: 모듈 레벨 양자 회로 합성

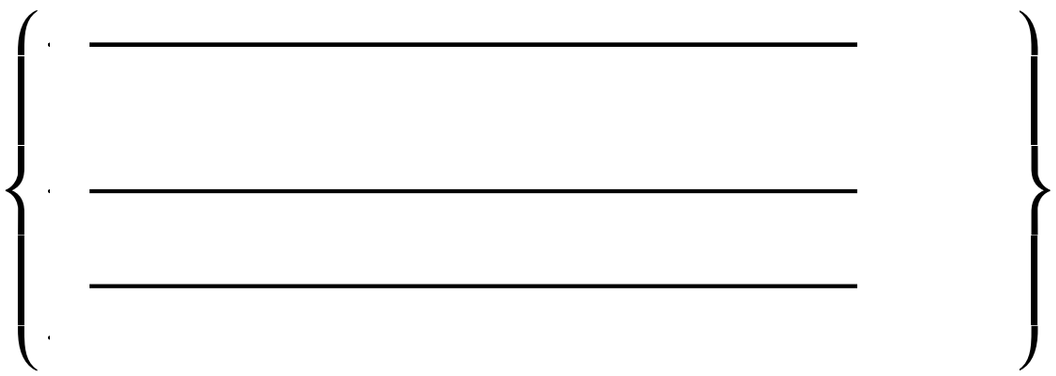
1. 모듈 = 게이트 조합 (up to some fixed length)

$M_1 =$ 

$M_2 =$ 

$M_3 =$ 

2. 명세와의 특성 차이를 줄이는 방향으로 모듈 쌓기

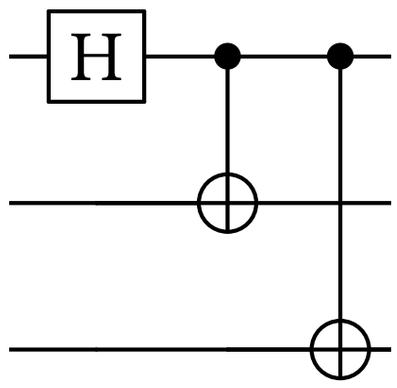
$|100\rangle$  $|GHZ\rangle = \frac{1}{\sqrt{2}}(|000\rangle + |111\rangle)$

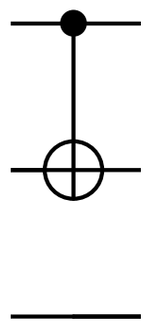
$|100\rangle \longleftrightarrow \frac{1}{\sqrt{2}}(|000\rangle + |111\rangle)$

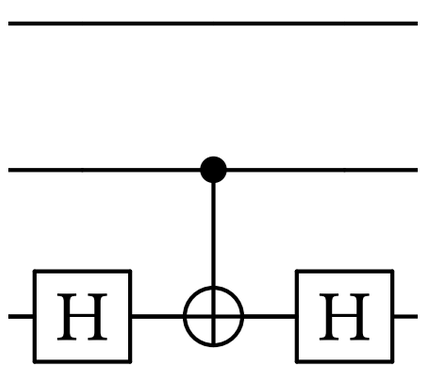
'얽힘' 차이

아이디어: 모듈 레벨 양자 회로 합성

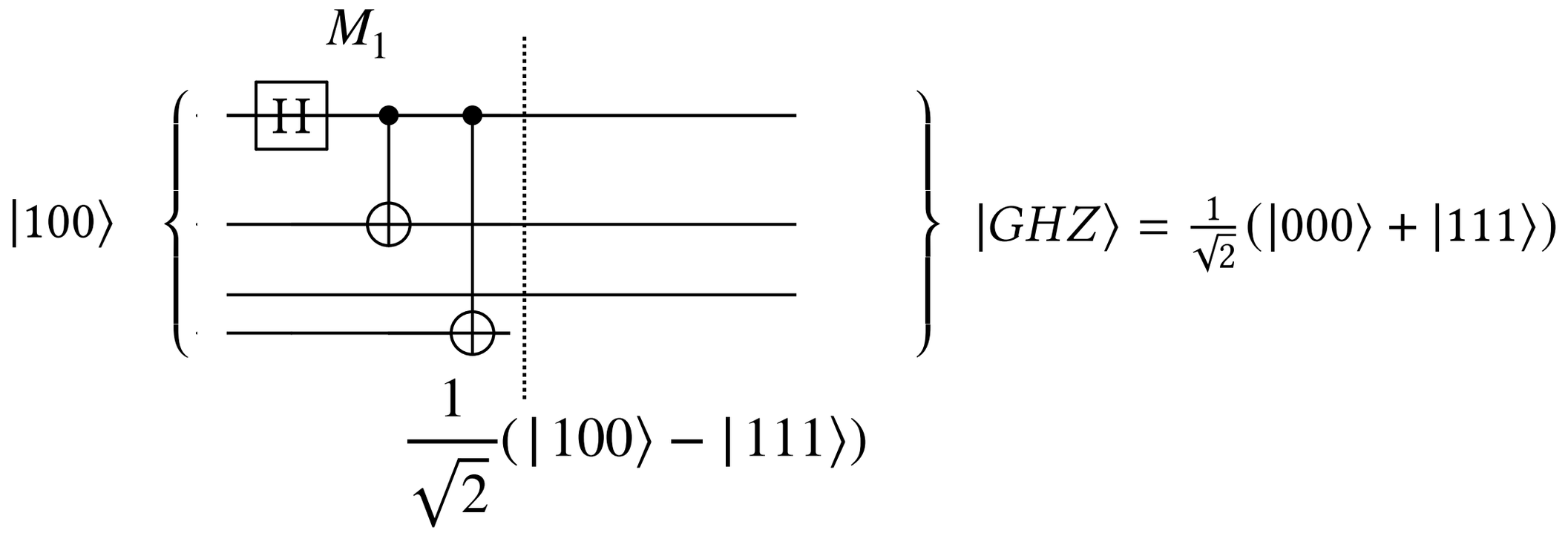
1. 모듈 = 게이트 조합 (up to some fixed length)

$M_1 =$ 

$M_2 =$ 

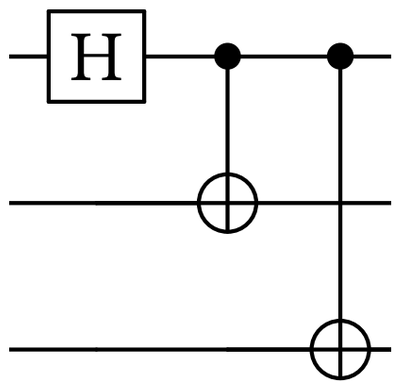
$M_3 =$ 

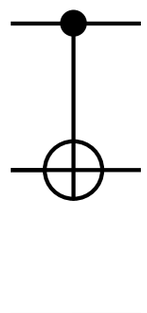
2. 명세와의 특성 차이를 줄이는 방향으로 모듈 쌓기

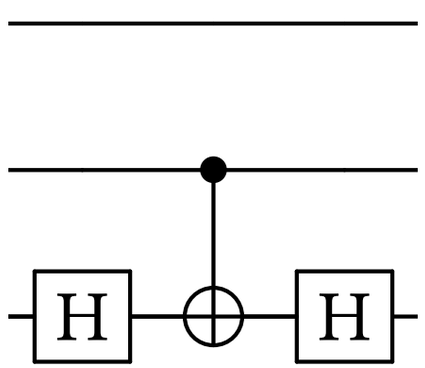


아이디어: 모듈 레벨 양자 회로 합성

1. 모듈 = 게이트 조합 (up to some fixed length)

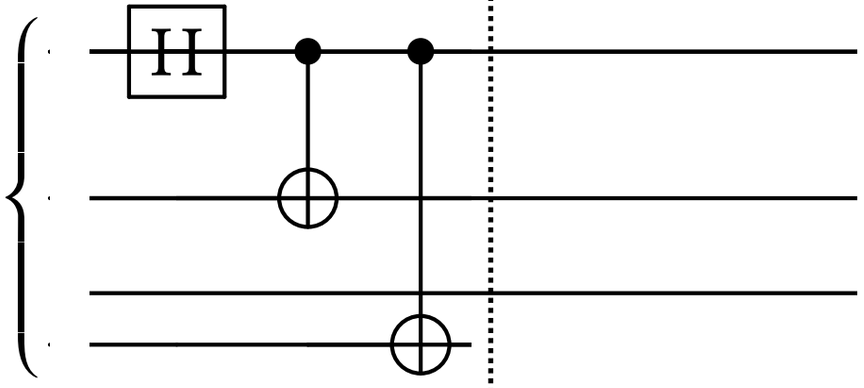
$M_1 =$ 

$M_2 =$ 

$M_3 =$ 

2. 명세와의 특성 차이를 줄이는 방향으로 모듈 쌓기

M_1

$|100\rangle$ 

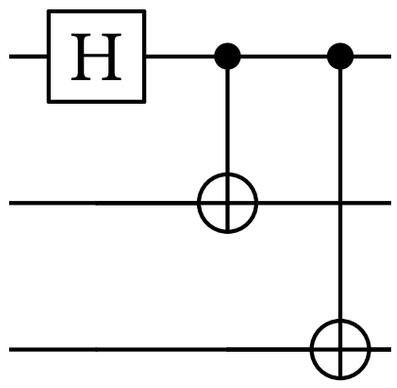
$|GHZ\rangle = \frac{1}{\sqrt{2}}(|000\rangle + |111\rangle)$

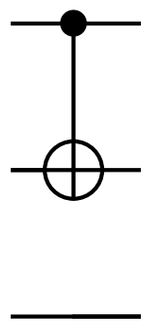
$\frac{1}{\sqrt{2}}(|100\rangle - |111\rangle)$ ⟷ $\frac{1}{\sqrt{2}}(|000\rangle + |111\rangle)$

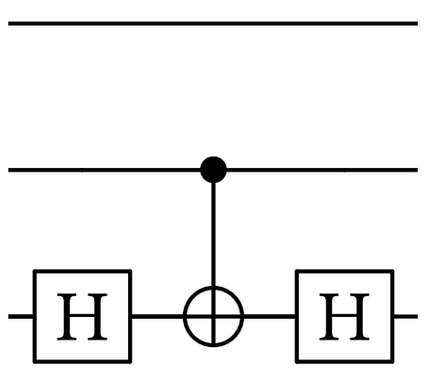
‘위상’ 차이

아이디어: 모듈 레벨 양자 회로 합성

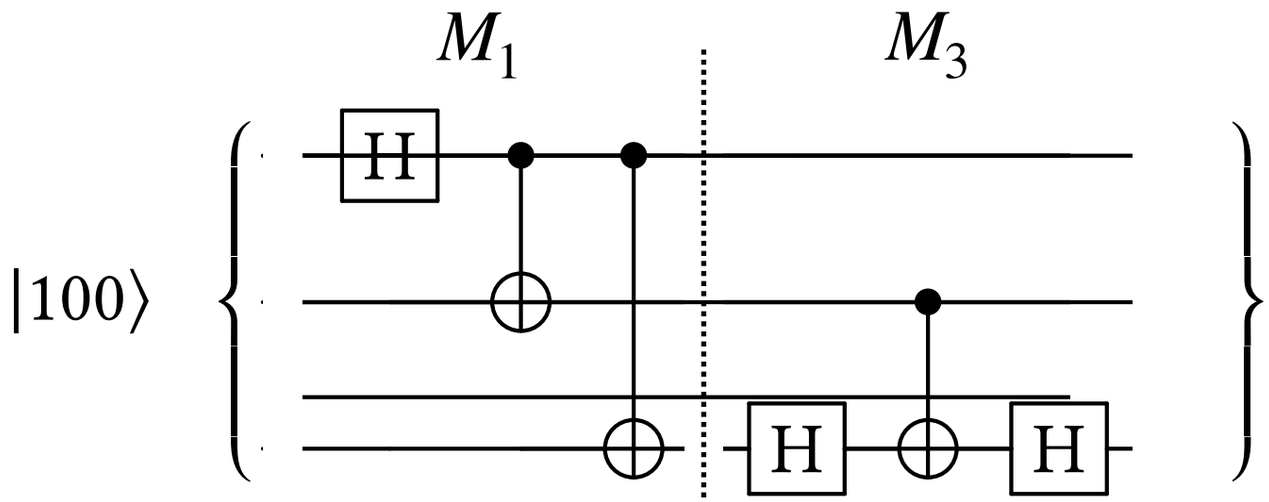
1. 모듈 = 게이트 조합 (up to some fixed length)

$M_1 =$ 

$M_2 =$ 

$M_3 =$ 

2. 명세와의 특성 차이를 줄이는 방향으로 모듈 쌓기



$|100\rangle$

$|GHZ\rangle = \frac{1}{\sqrt{2}}(|000\rangle + |111\rangle)$

$\frac{1}{\sqrt{2}}(|100\rangle - |111\rangle)$ ‘위상’ 차이 $\frac{1}{\sqrt{2}}(|000\rangle + |111\rangle)$

합성 알고리즘의 안전성 (Soundness)

"특정 가정하에서 모듈 기반 합성 알고리즘이 항상 정답을 찾아냄"

THEOREM 4.17. *Let* $E = \{(|in\rangle, |out\rangle)\}$ *be an example and* $C^* = M_1; \cdots; M_k$ *(*$M_i \in \mathcal{M}$ *and attribute of each* M_i *is not* IDENTITY*) be the solution circuit to be synthesized such that* $C^*(|in\rangle) = |out\rangle$*. Suppose* C^* *is monotonically decreasing (by input* $|in\rangle$*). Then, for any prefix* $C = M_1; \cdots; M_{l-1}$ $(l \leq k)$ *of* C^**,*

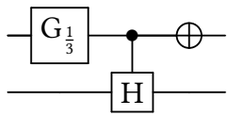
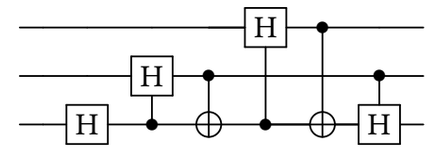
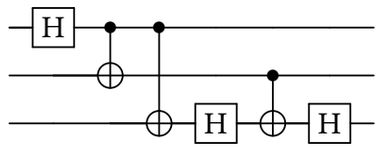
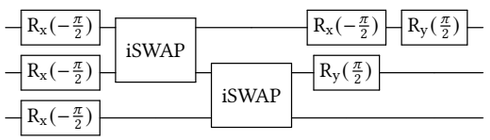
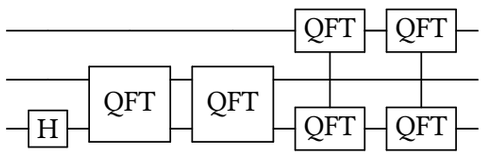
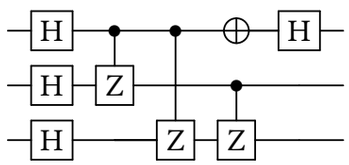
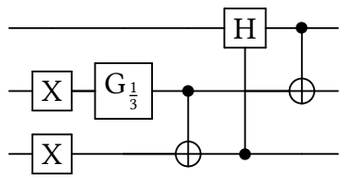
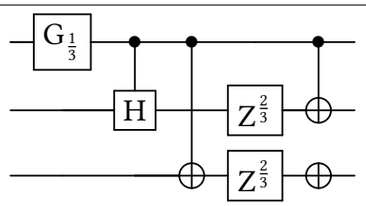
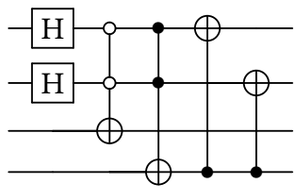
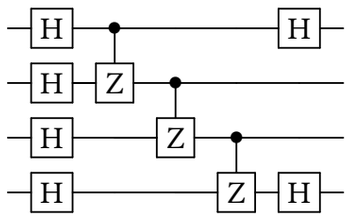
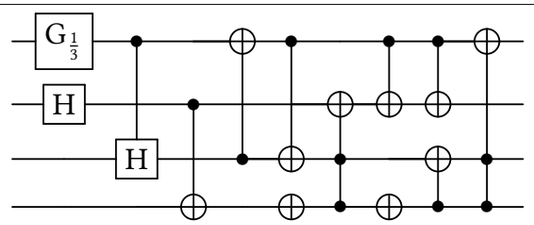
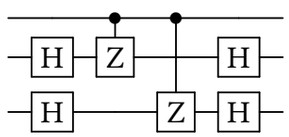
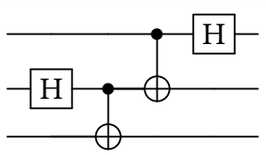
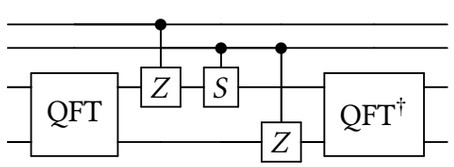
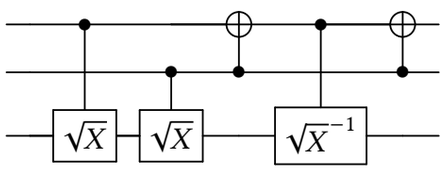
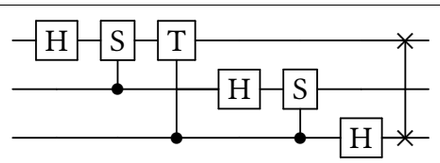
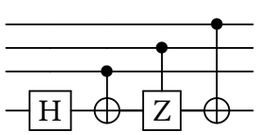
$$\text{is\_gap\_filled}(C, M_l, (|in\rangle, |out\rangle)) = \mathit{True}.$$

PROOF. Let $|\psi\rangle = C\,|in\rangle = M_{l-1} \cdots M_1\,|in\rangle$ be the output vector of C. By definition $M_k \cdots M_l\,|\psi\rangle = |out\rangle$ and thus

$$Att_{|\psi\rangle}(M_k M_{k-1} \cdots M_l) = |out\rangle \ominus |\psi\rangle = \text{gap\_att}_{|in\rangle,|out\rangle}(C).$$

Since C^* (and so C) is decreasing, by Lemma 4.16 $Att_{|\psi\rangle}(M_k M_{k-1} \cdots M_l) = Att_{|\psi\rangle}(M_l)$. Therefore, $\text{gap\_att}_{|in\rangle,|out\rangle}(C) = Att_{|\psi\rangle}(M_l)$, which is satisfying the criterion. □

벤치마크

Type	ID	Circuit	ID	Circuit
State Preparation	three_superpose		M_valued	
	GHZ_from_100		GHZ_by_iSWAP	
	GHZ_by_QFT		GHZ_Game	
	W_orthog		W_phased	
	W_four		cluster	
	bit_measure			
Multi IO	flip		teleportation	
	draper		toffoli_by_$\sqrt{X}$	
	QFT		indexed_bell	

게이트 레벨 합성 알고리즘

모듈 레벨 합성 알고리즘

ID	Base_{no_prune}	Base	Ours_{no_prune}	Ours	Spd-up
three_superpose	0.14	0.12	0.12	0.09	1x
M_valued	1764.75	1126.79	666.25	3.89	290x
GHZ_from_100	106.81	48.16	⊥	0.47	102x
GHZ_by_iSWAP	⊥	⊥	690.67	2.19	-
GHZ_by_QFT	116.90	101.65	101.17	39.26	3x
GHZ_Game	⊥	2305.71	4.51	0.57	4058x
W_orthog	2927.20	2075.06	248.23	2.43	854x
W_phased	⊥	⊥	258.56	5.43	-
W_four	⊥	⊥	2851.10	254.88	-
cluster	⊥	⊥	3560.38	8.91	-
bit_measure	⊥	⊥	⊥	⊥	-
flip	18.56	3.95	0.76	0.83	5x
teleportation	2.02	1.30	1.35	1.35	1x
indexed_bell	14.67	11.66	1.60	1.52	8x
toffoli_by_$\sqrt{X}$	956.29	716.28	306.10	264.66	3x
QFT	⊥	⊥	⊥	220.87	-
draper	⊥	⊥	933.47	737.99	-
Avg. (excluding ⊥/-)	656.37	639.07	687.45	96.58	20x

기회

- 양자 SW + X
  - 양자 SW 합성
  - 양자 SW 최적화
  - 양자 SW 정적 분석
  - 양자 SW 동적 분석
  - 양자 SW 정형 검증
  - 양자 SW 자동 수정
- 3無
  - 관심, 적용사례, 컴퓨터

감사합니다!