A Short Summary

and a lot of questions…so far

Chenchao Ding, 09/11/2023

Single Qubit Operations

$\mathbf{p} : \mathbb{Q} \to \mathbb{Q}$

- Pauli gates X, Y, Z + Hadamard gate H
- usually represented as unitary matrix
- rotating $\boldsymbol{\pi}$ degree anti-clockwise along one certain axis
- reversible

Single Qubit Operations

$\mathbf{p} : \mathbb{Q} \rightarrow \mathbb{Q}$

Example : Hadamard gate

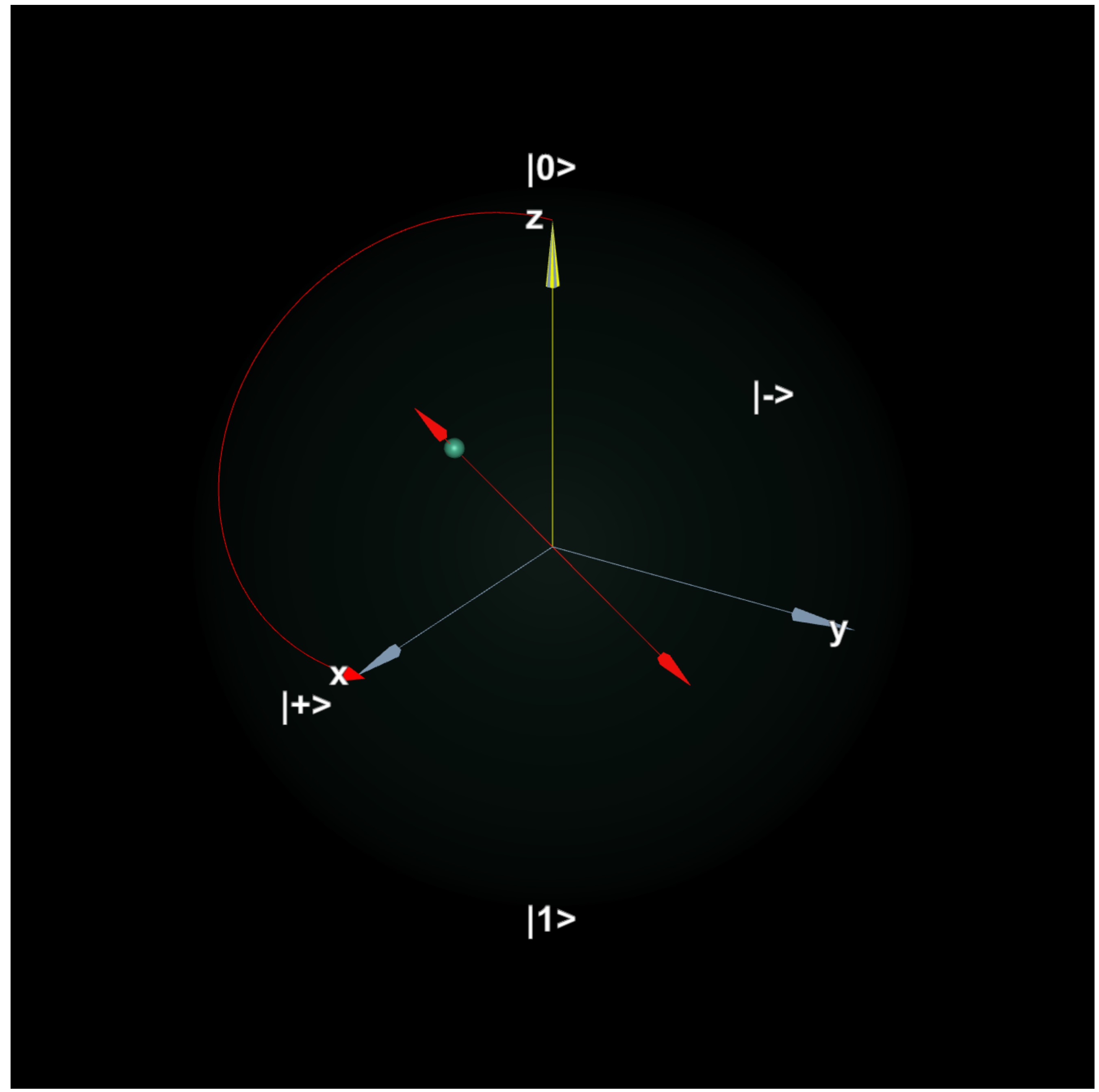
It’s “total effect” is - for every pure state (“point” on the surface), it rotates **π** degree around the red axis in this diagram.

$|0\rangle$ goes to $|+\rangle$

$|+\rangle$ goes to $|0\rangle$

$|1\rangle$ goes to $|-\rangle$

$|-\rangle$ goes to $|1\rangle$



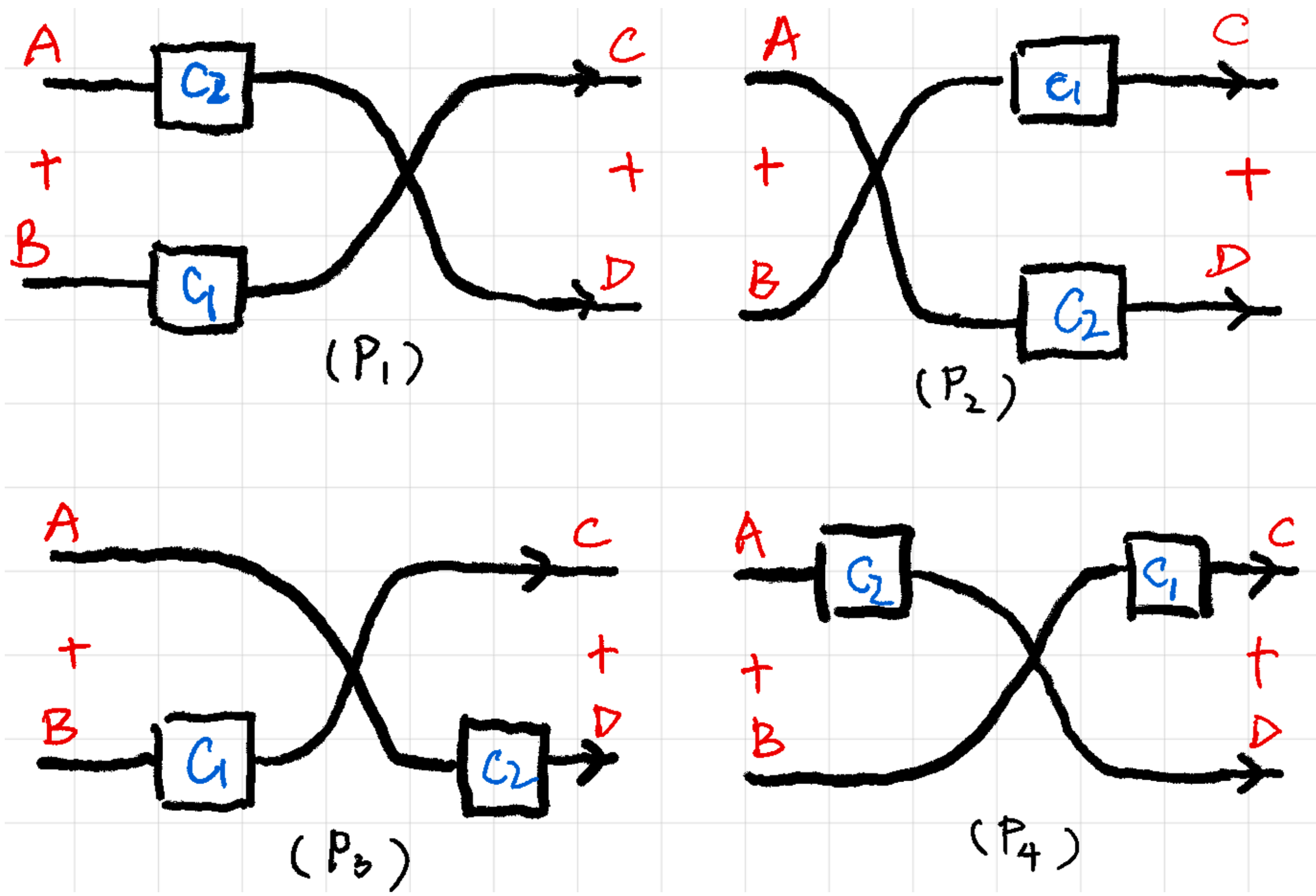
Sequential Composition

`_∘_ : (p : A → B) → (q : B → C) → (A → C)`

- `I ∘ I ~ I`
- `X ∘ I ~ X`
- `I ∘ X ~ X`
- `X ∘ X ~ I`
- `H ∘ H ~ I`
- `f ∘ (g ∘ h) ~ (f ∘ g) ∘ h`
- …

Homotopic Operations

p ~ q




```
p₁ p₂ p₃ p₄ : (A + B) → (C + D)
p₁ = (c₂ ⊕ c₁) ⊙ swap₊
p₂ = swap₊ ⊙ (c₁ ⊕ c₂)
p₃ = (id ⊕ c₁) ⊙ swap₊ ⊙ (id ⊕ c₂)
p₄ = (c₂ ⊕ id) ⊙ swap₊ ⊙ (c₁ ⊕ id)
```

2-Level Combinators

p ⇔ q

```
data _⇔_ : {A B : Π₂} (p q : A ↔ B) → Type where

  `id₂   : {A B : Π₂} {c : A ↔ B} → c ⇔ c
  !₂_    : {A B : Π₂} {p q : A ↔ B} → (p ⇔ q) → (q ⇔ p)
  _⊙₂_   : {A B : Π₂} {p q r : A ↔ B} → (p ⇔ q) → (q ⇔ r) → (p ⇔ r)

  !id₁   : {A : Π₂} → !₁ `id₁{A} ⇔ `id₁{A}
  !not   : !₁ `not ⇔ `not

  idl⊙l : {A B : Π₂} {c : A ↔ B} → (`id₁ ⊙ c) ⇔ c
  idr⊙l : {A B : Π₂} {c : A ↔ B} → (c ⊙ `id₁) ⇔ c


  assoc⊙l : {A B C D : Π₂} {p : A ↔ B} {q : B ↔ C} {r : C ↔ D}
          → (p ⊙ q) ⊙ r ⇔ p ⊙ (q ⊙ r)
  assoc⊙r : {A B C D : Π₂} {p : A ↔ B} {q : B ↔ C} {r : C ↔ D}
          → p ⊙ (q ⊙ r) ⇔ (p ⊙ q) ⊙ r
```

Back to Hadamard…

H ~ ?

Example : Hadamard gate

It's "total effect" is - for every pure state ("point" on the surface), it rotates π degree around …

What does this suspicious "total effect" mean???

What kind of operations are "homotopic to" Hadamard?

A "representative" of a bunch of equivalent operations?

If we have (one of) Hadamard and CNOT…

Square Roots Introduction

sqrt : (p : ℚ → ℚ) → (ℚ → ℚ)

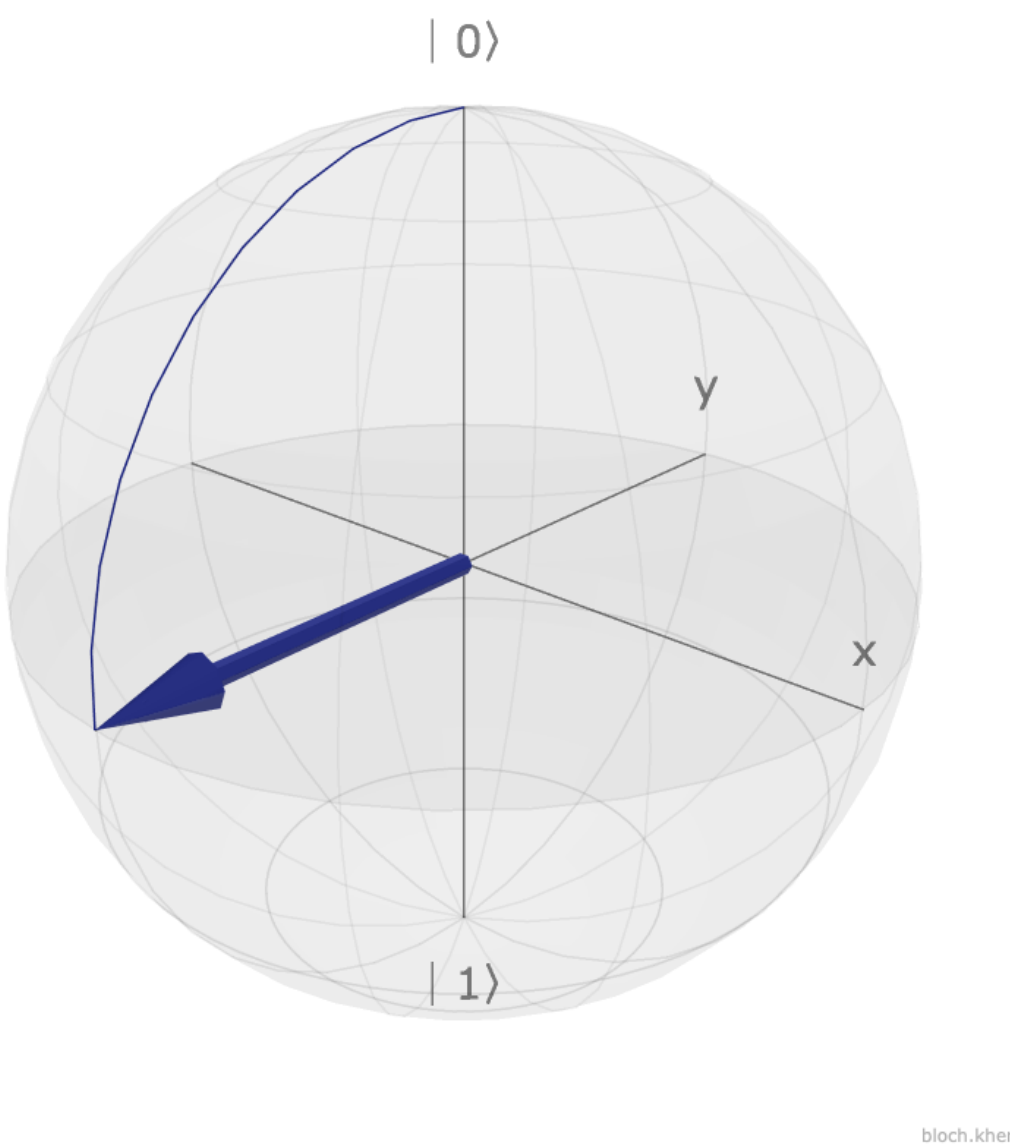
```
data _↔_ : (A B : Π₂) → Type where

  sqrt   : {A : Π₂} → (c : A ↔ A) → (A ↔ A)
```

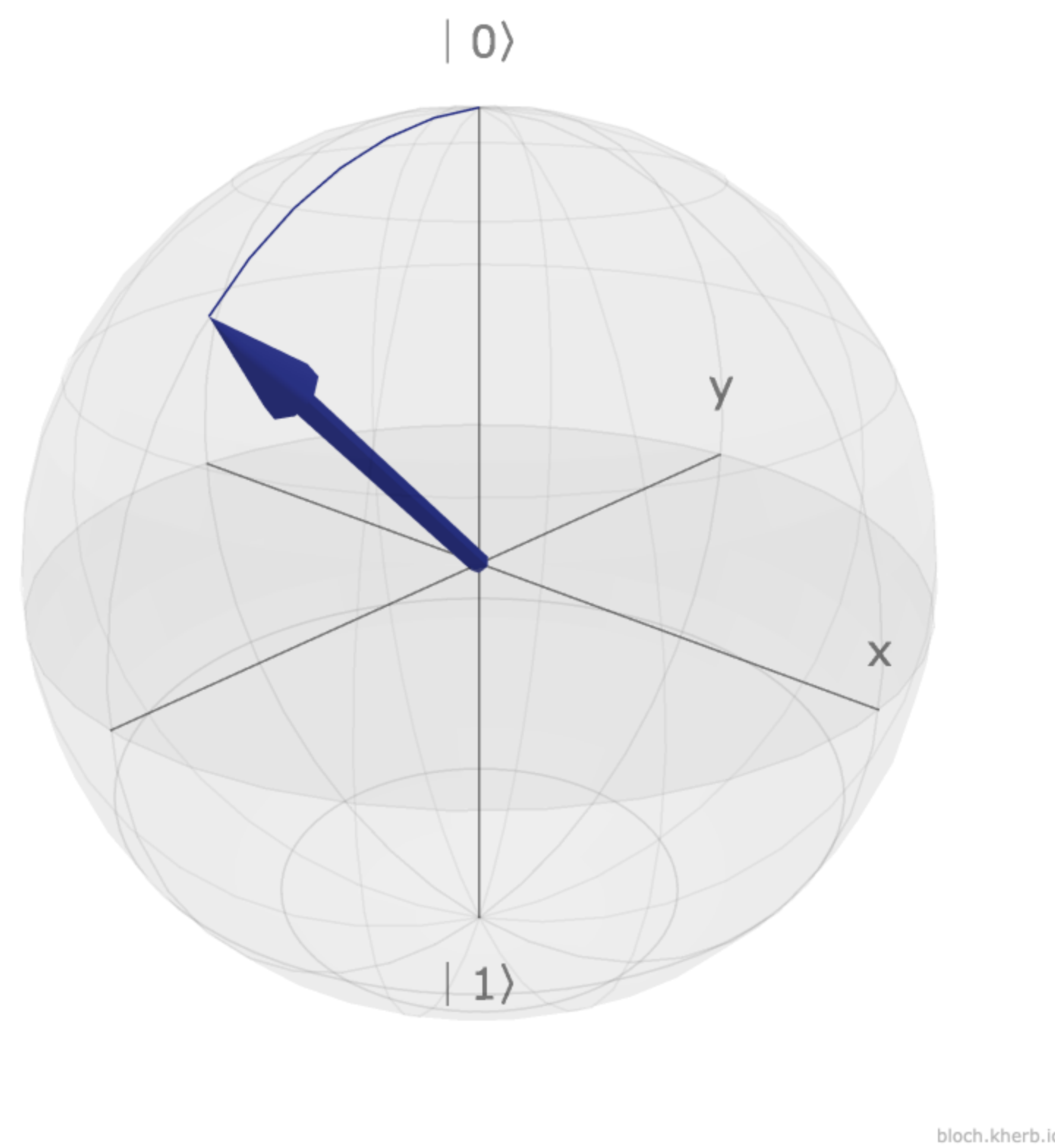
```
data _⇔_ : {A B : Π₂} (p q : A ↔ B) → Type where

sqd    : {A : Π₂} {c : A ↔ A} → sqrt c ⊙ sqrt c ⇔ c
sqf    : {A : Π₂} {c : A ↔ A} → sqrt (c ⊙ c) ⇔ sqrt c ⊙ sqrt c
sqi    : {A : Π₂} {p q : A ↔ A} → (p ⇔ q) → sqrt p ⇔ sqrt q
sqc    : {A : Π₂} {c : A ↔ A} → sqrt c ⊙ c ⇔ c ⊙ sqrt c -- derivable from assoc and sqd
sq!    : {A : Π₂} {c : A ↔ A} → sqrt (!₁ c) ⇔ !₁ (sqrt c)
```

Square Roots Introduction

`sqrt : (p : ℚ → ℚ) → (ℚ → ℚ)`

Example : `sqrt(X), sqrt(sqrt(X))`

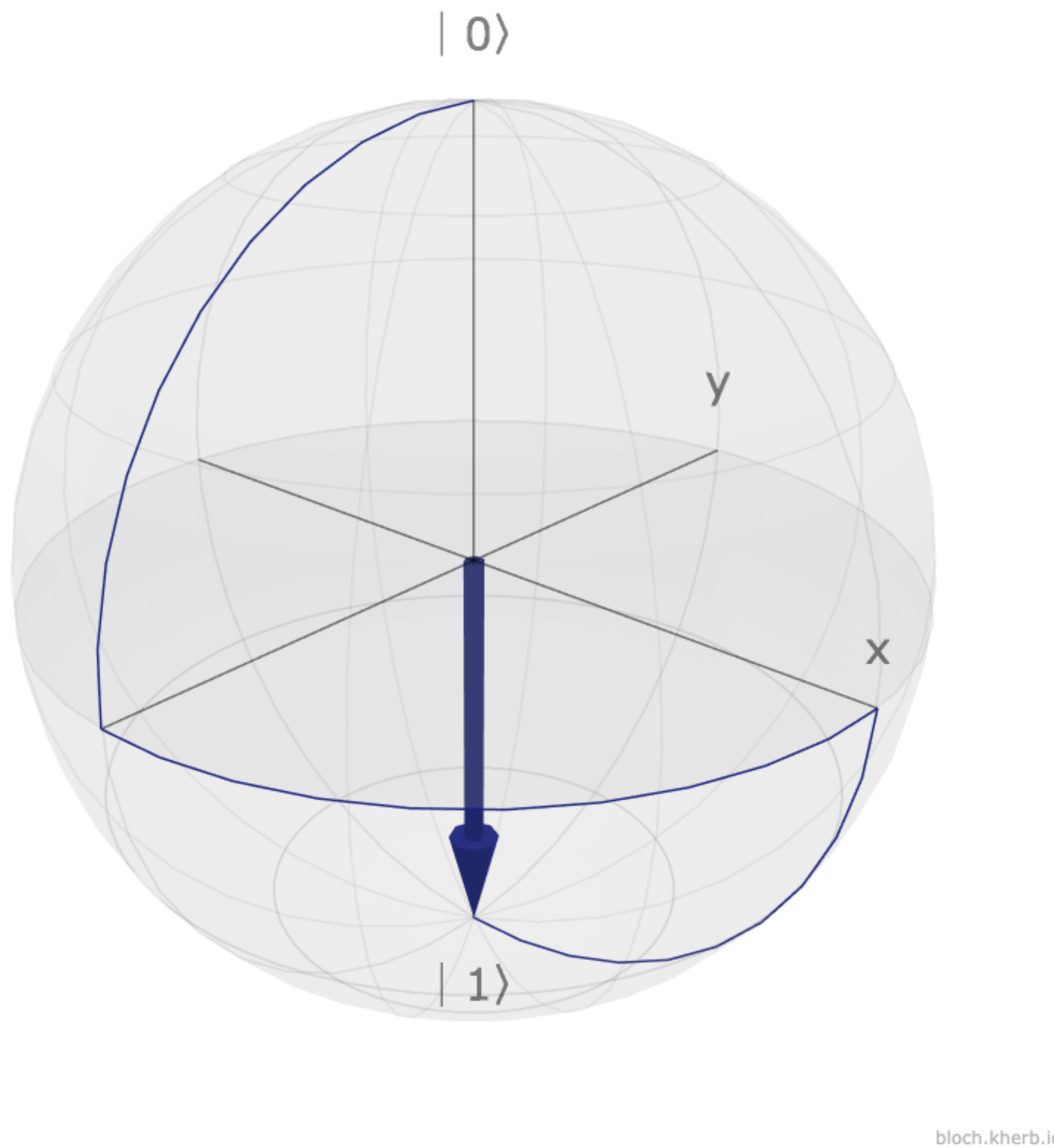




Square Roots Introduction

`sqrt : (p : ℚ → ℚ) → (ℚ → ℚ)`

Example : `sqrt(X) ∘ S ∘ sqrt(Y)`



Back to Hadamard…Again!

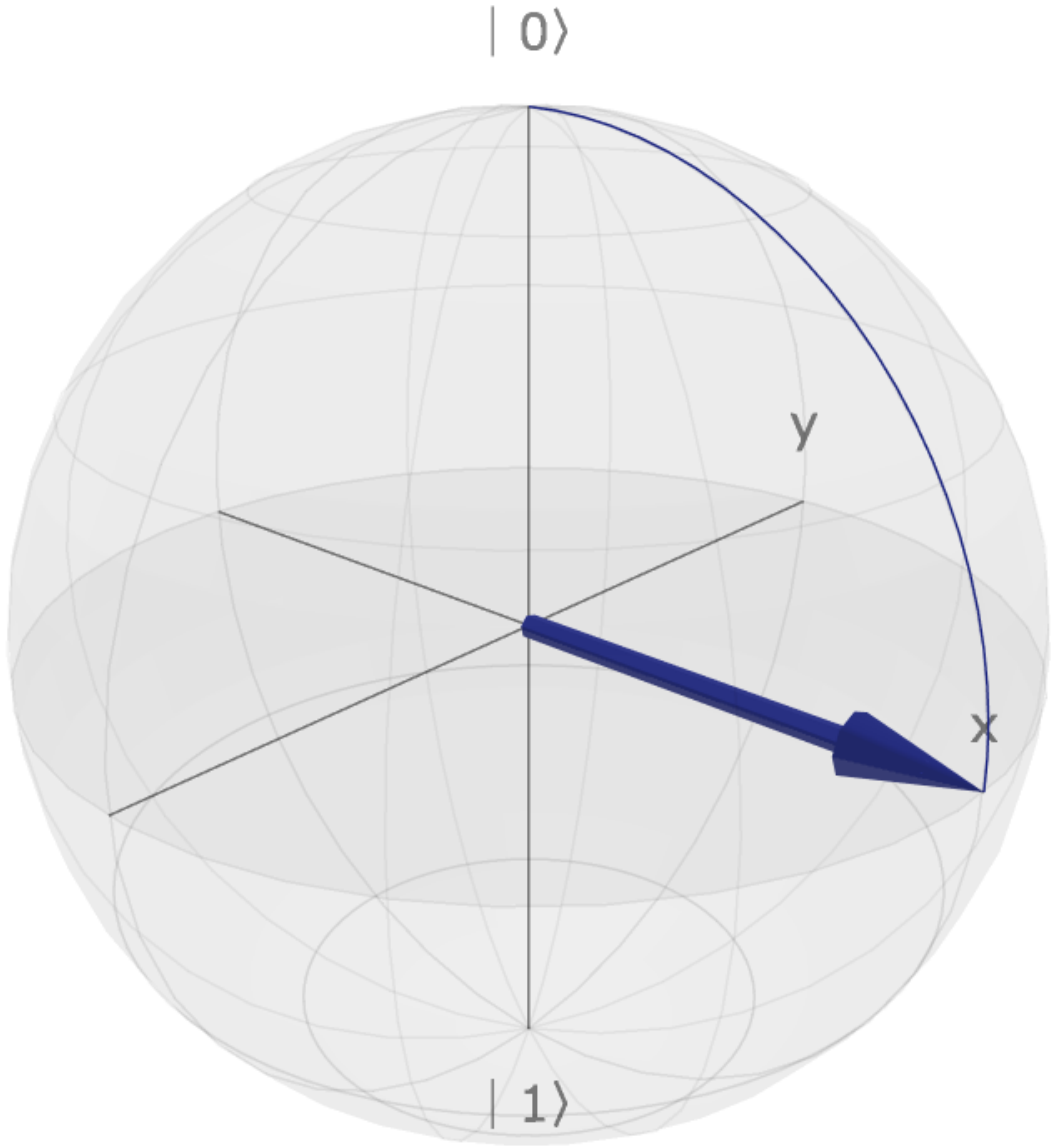
$H \sim \omega^2 \bullet X \circ S \circ \mathrm{sqrt}(X) \circ S \circ X$

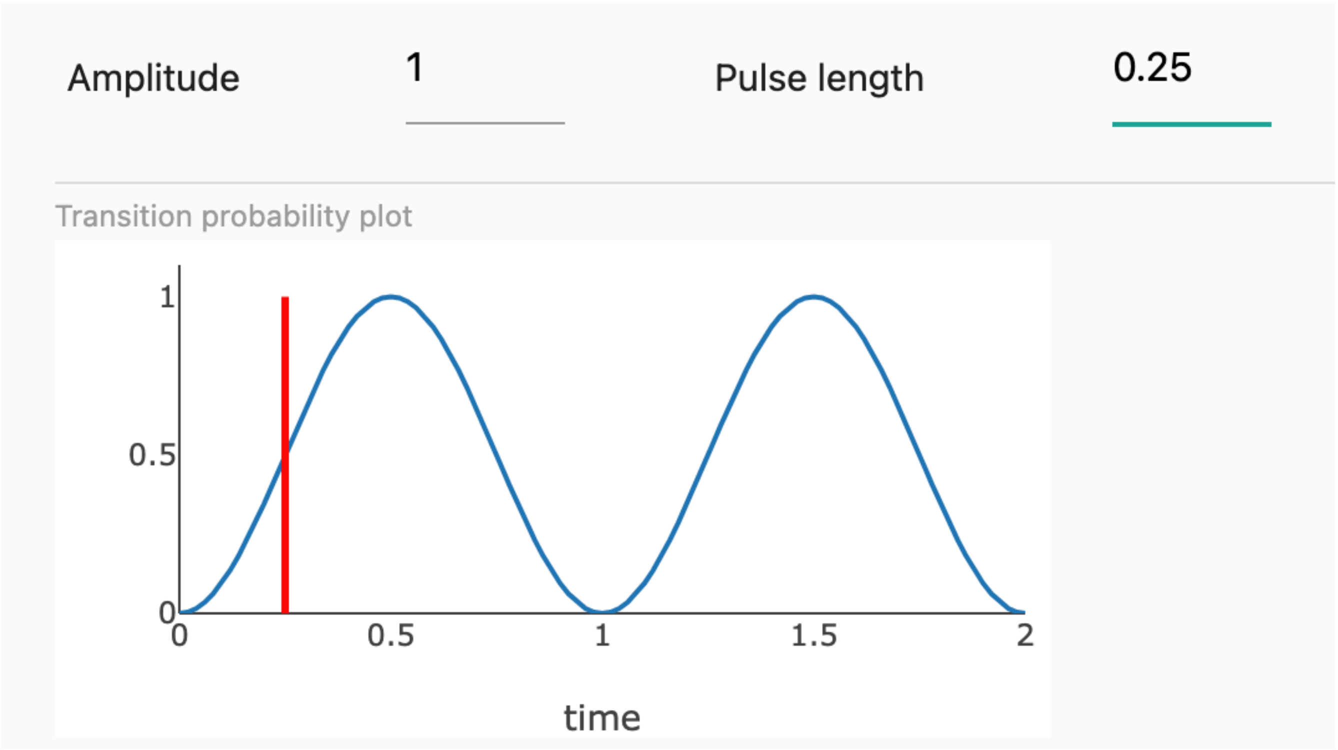
Completeness : In general, how few square roots (of what operations/gates?) do we need as minimum requirements to get quantum computing?

More precise approximation of quantum states/gates requires more square roots (?)

Pulses & Full-Analog Signals

timely computation

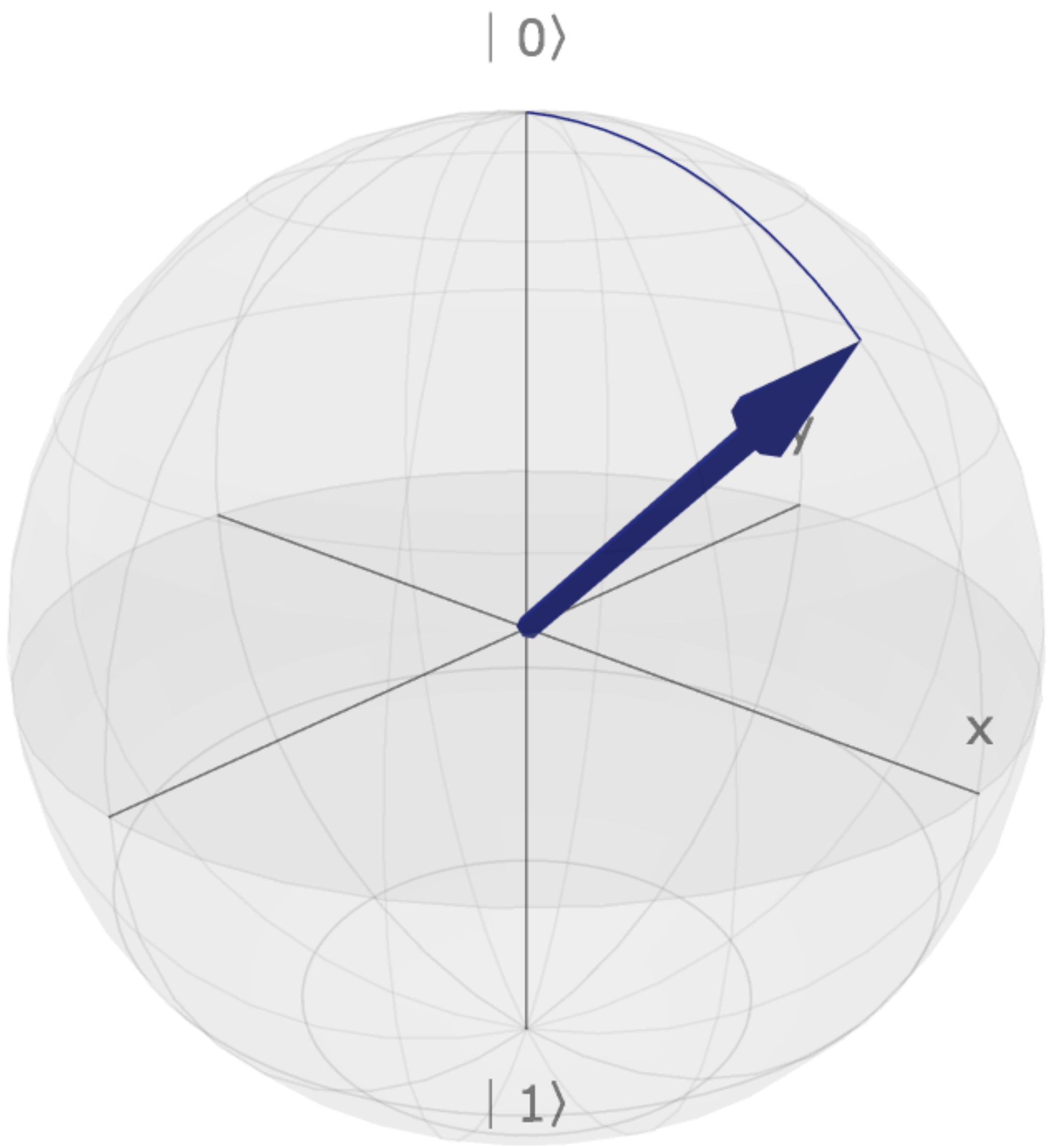




bloch.kherb.io

Pulses & Full-Analog Signals

timely computation

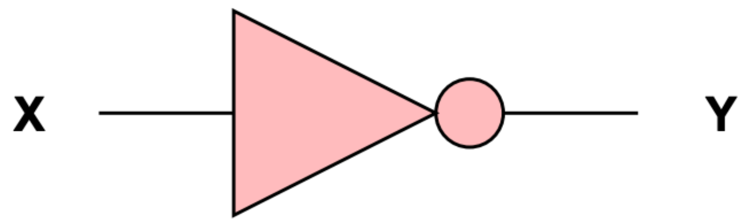


bloch.kherb.io

Pulses & Full-Analog Signals

static timing analysis of circuits

- Delay, rising/falling delay



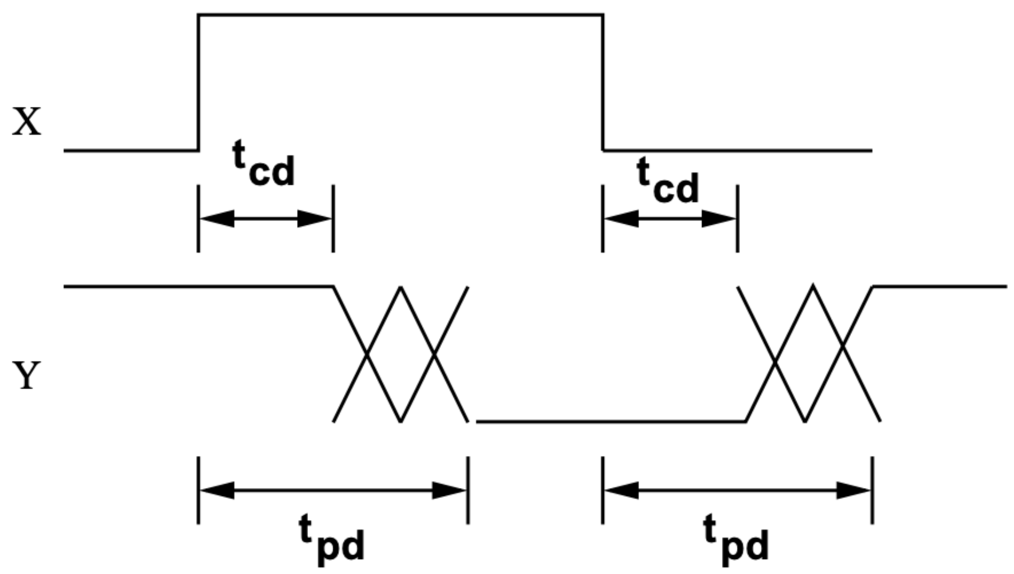
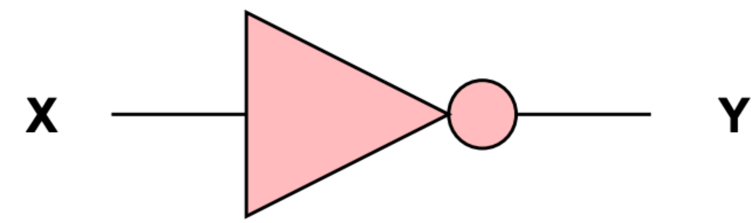


Figure 2: Combinational Propagation and Contamination Delay

Pulses & Full-Analog Signals

static timing analysis of circuits

- Functions are abstract and discrete, while circuits are concrete and continuous
- Full-analog signals/circuits have delays, rising/falling delays…
- Only stable signals/intervals are friendly for measurements
- Generally, how is the square roots idea connected to "physical reality"?



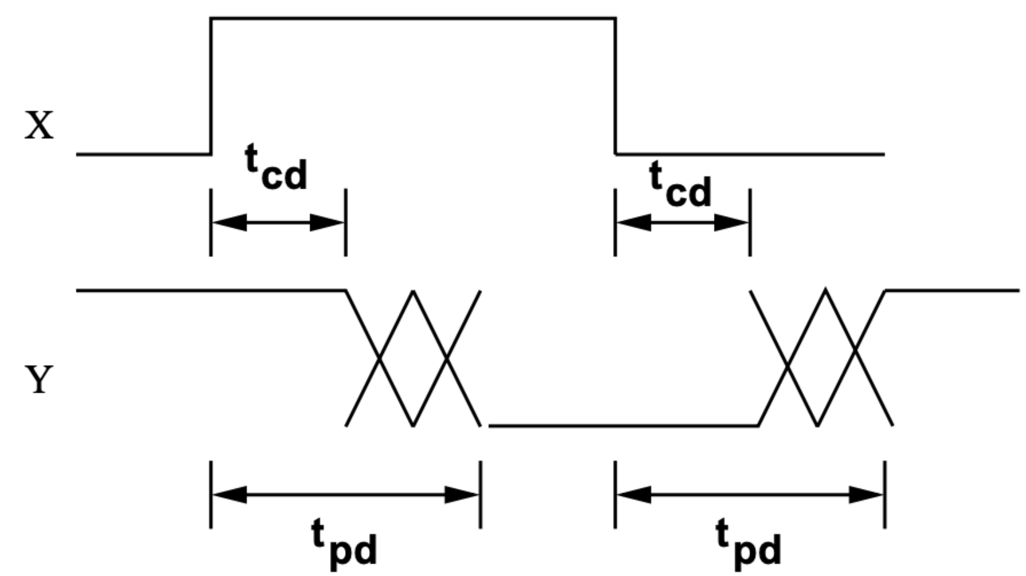


Figure 2: Combinational Propagation and Contamination Delay

Why Cubical?

A “natural” connection…

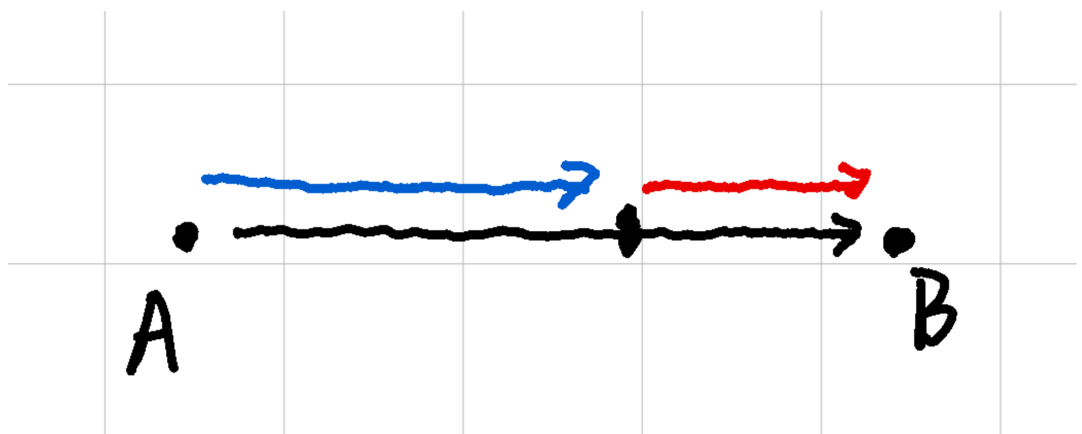
- In Cubical, only endpoints are “tangible” (i.e. only i0 and i1 are instances of the Interval type), but we can parameterize over the Interval and…

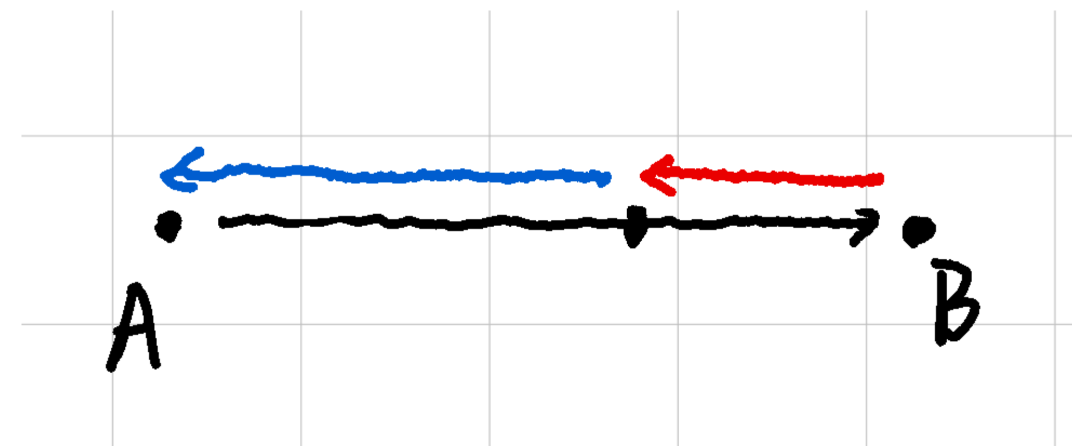
- In Quantum Computing, single measurement will either produce 0 or 1, but there are superpositions of states “in-between” and…

Sounds similar but has a “tangible” gap “in-between”

Why Cubical?

What I tried with Cubical Agda…



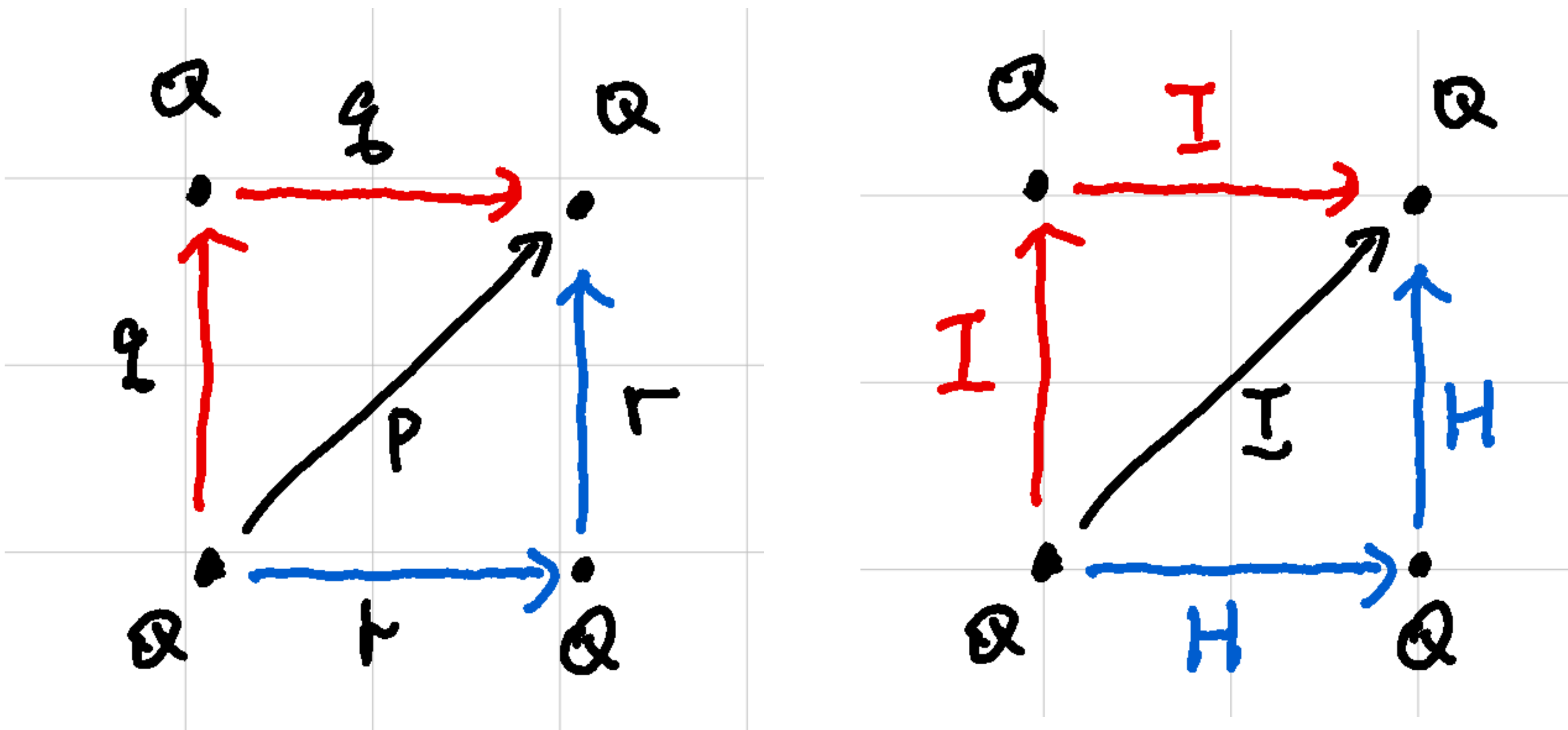



```
sqrt← : ∀ {ℓ} {A B : Type ℓ} (p : A ≡ B) (i : I)
      → (p i1 → p i) × (p i → p i0)
sqrt← p i = (λ b → transp (λ j → p (i ∨ ~ j)) i0 b)
          , (λ b → transp (λ j → p (i ∧ ~ j)) i0 b)
```


```
sqrt← : ∀ {ℓ} {A B : Type ℓ} (p : A ≡ B) (i : I)
      → (p i1 → p i) × (p i → p i0)
sqrt← p i = (λ b → transp (λ j → p (i ∨ ~ j)) i0 b)
          , (λ b → transp (λ j → p (i ∧ ~ j)) i0 b)
```

Why Cubical?

don’t know how to construct from diagnals…



No need for full-fledged “Qubit” type :
one of the aims is to get rid of matrices and complex numbers, and obtain a nice programming language model with combinators