# ‘Cause I’m Strong Enough: Reasoning about Consistency Choices in Distributed Systems

Presented By:
Aldrin Montana

# What are the takeaways?

# Example

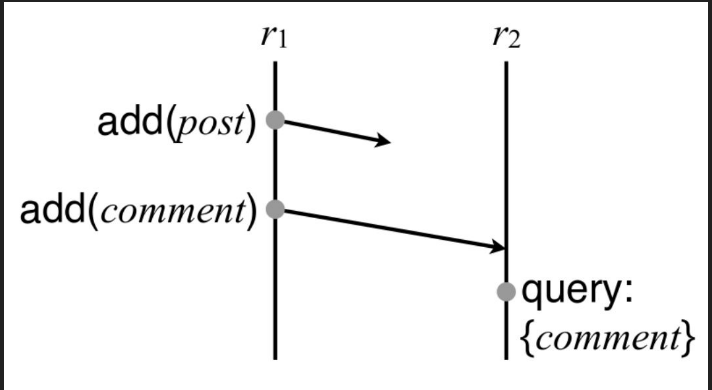



Figure 1A
Illustration of Add and Query

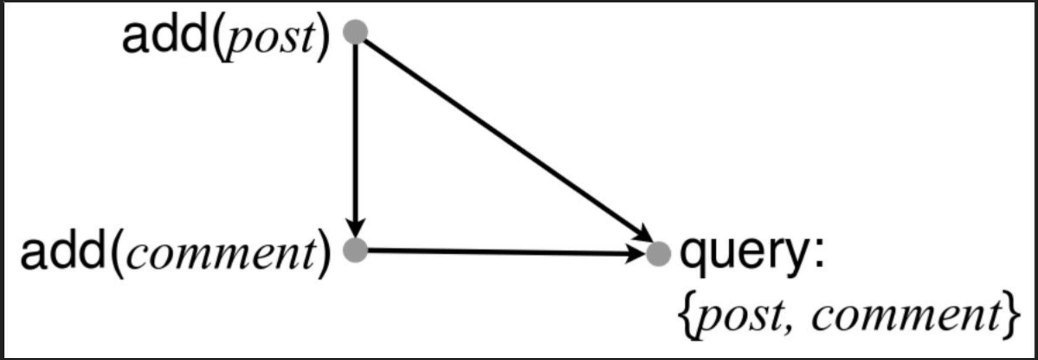



Figure 2A
Example of Definition 1
for Add and Query

# Example

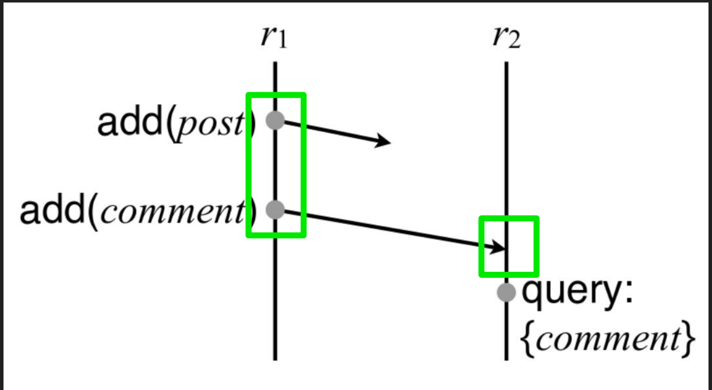



Figure 1A
Illustration of Add and Query

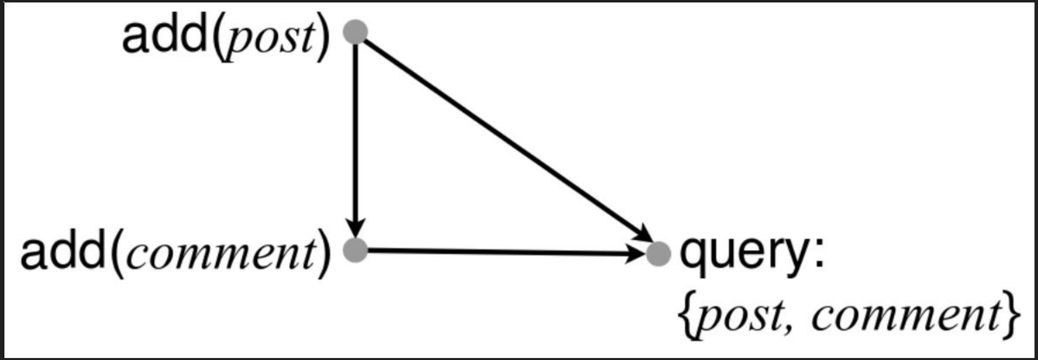



Figure 2A
Example of Definition 1
for Add and Query

# Example

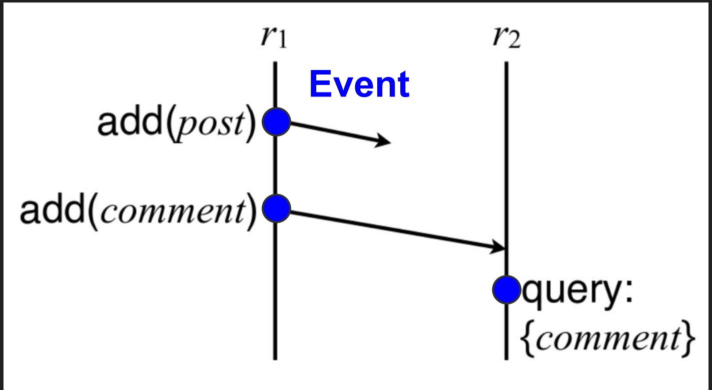



Figure 1A
Illustration of Add and Query

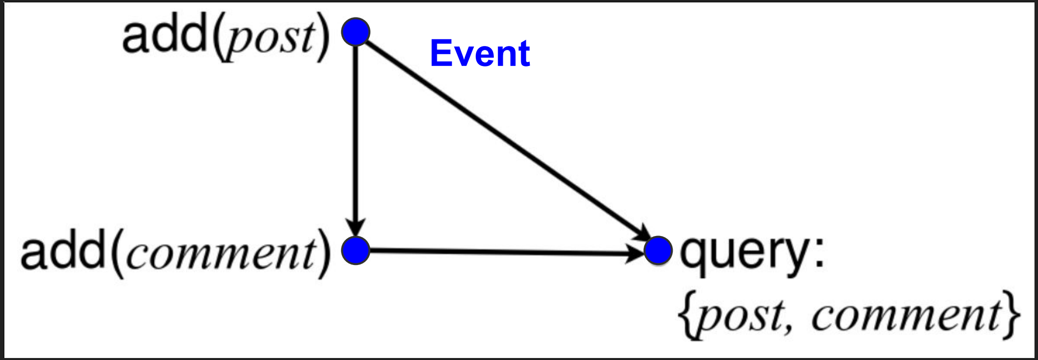



Figure 2A
Example of Definition 1
for Add and Query

# Example

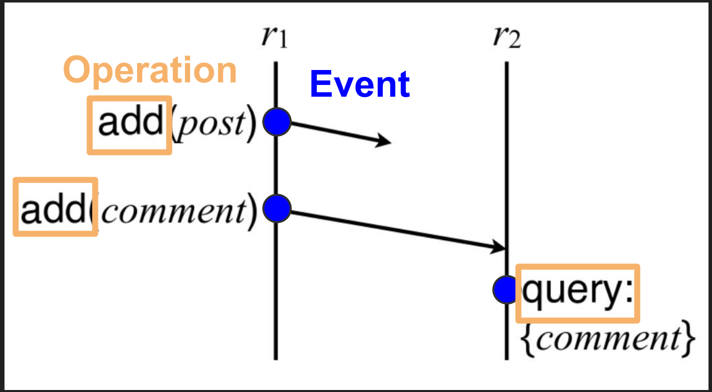



Figure 1A
Illustration of Add and Query

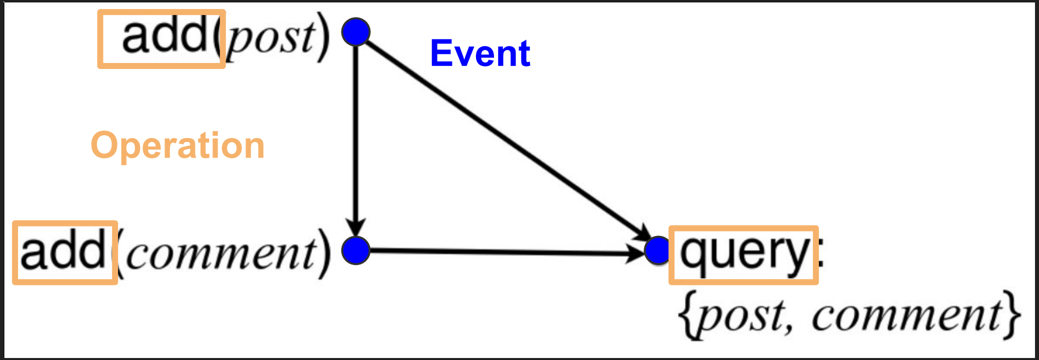



Figure 2A
Example of Definition 1
for Add and Query

# Example

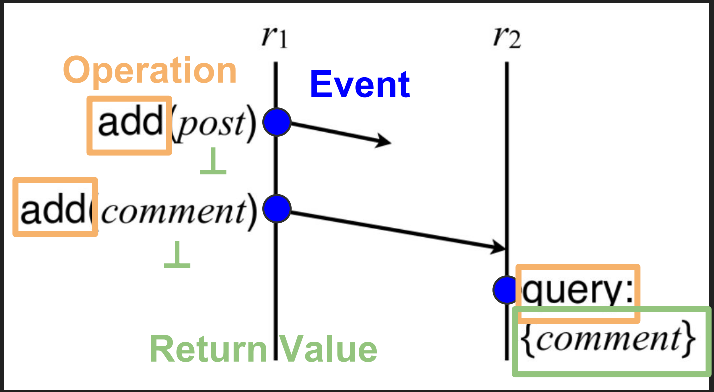



Figure 1A
Illustration of Add and Query

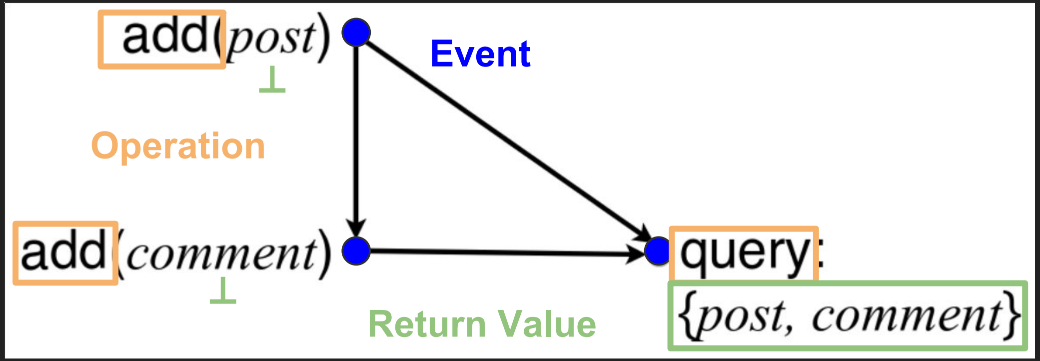



Figure 2A
Example of Definition 1
for Add and Query

# Example

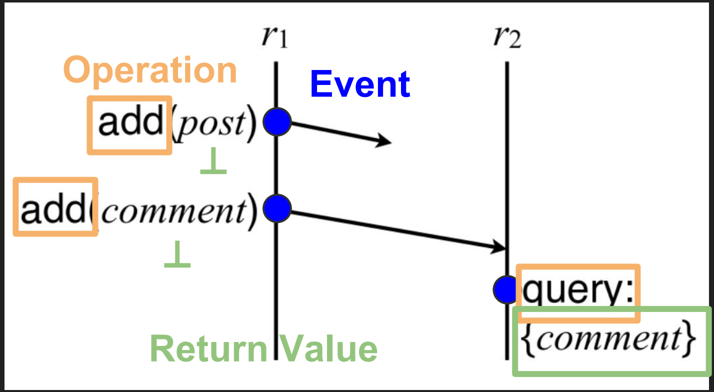



Figure 1A
Illustration of Add and Query

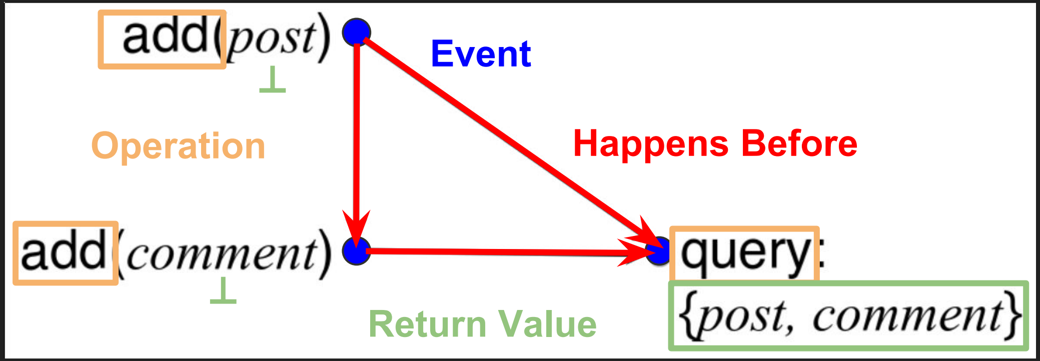



Figure 2A
Example of Definition 1
for Add and Query

# Example - What is the effect?

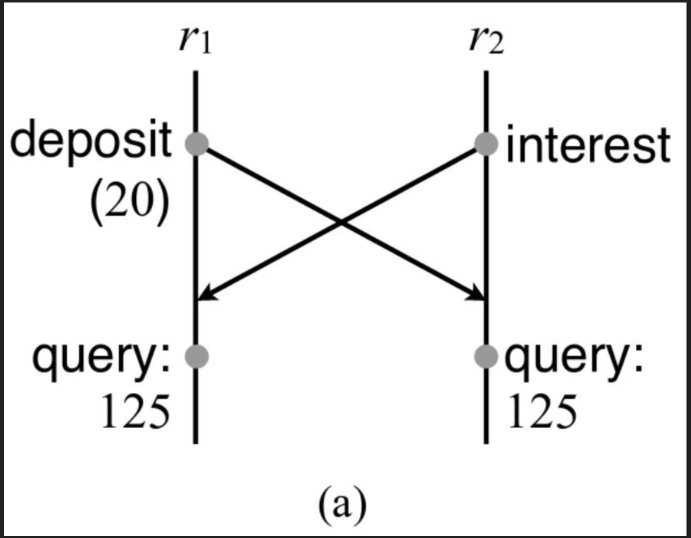



Figure 3C
Illustration of Deposit, Interest and Query

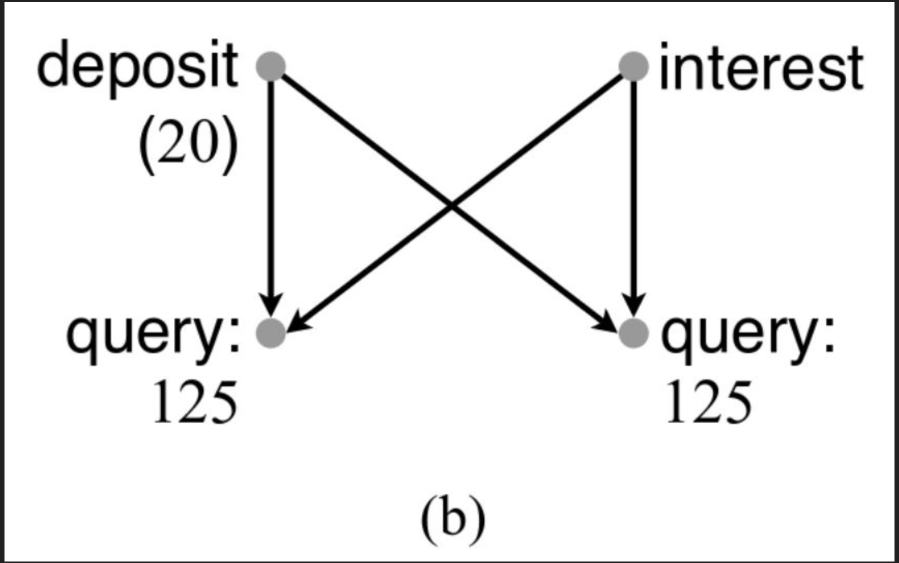



Figure 3C
Example of Definition 1
for Deposit, Interest and Query

# Example - What is the effect?

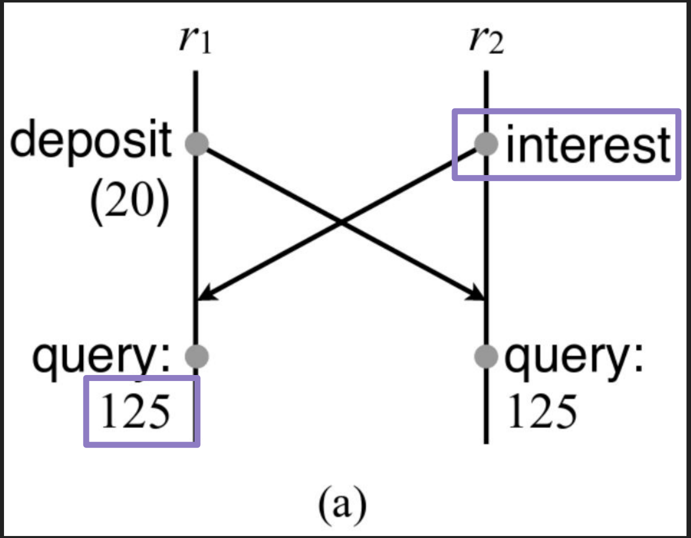



Figure 3C
Illustration of Deposit, Interest and Query

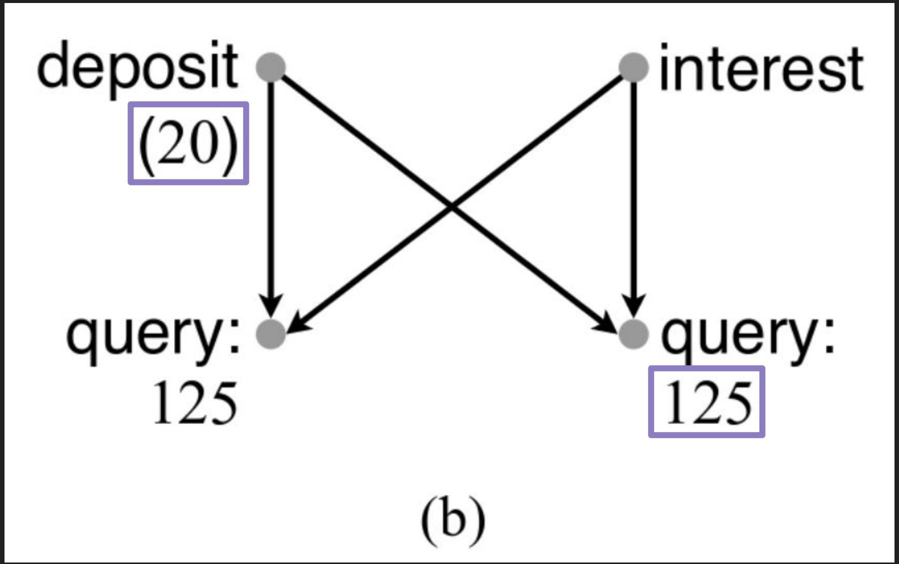



Figure 3C
Example of Definition 1
for Deposit, Interest and Query

# Definitions and Notations

$$F \in Op \to (State \to (Val \times (State \to State)))$$

$$F_o(\sigma) \quad = \quad (Val\ ,\ (State \to State))$$

$$F_o(\sigma) \quad = \quad (F_o^{val}(\sigma)\ ,\ (\ F_o^{eff}(\sigma)\qquad)))$$

# Example

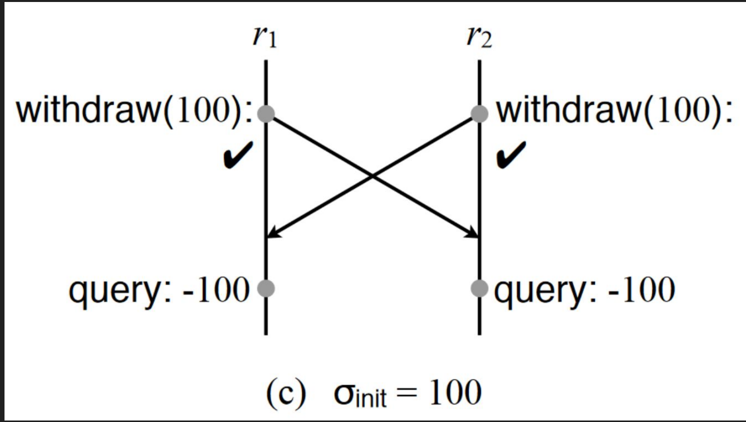



Figure 1C
Illustration of Withdraw and Query

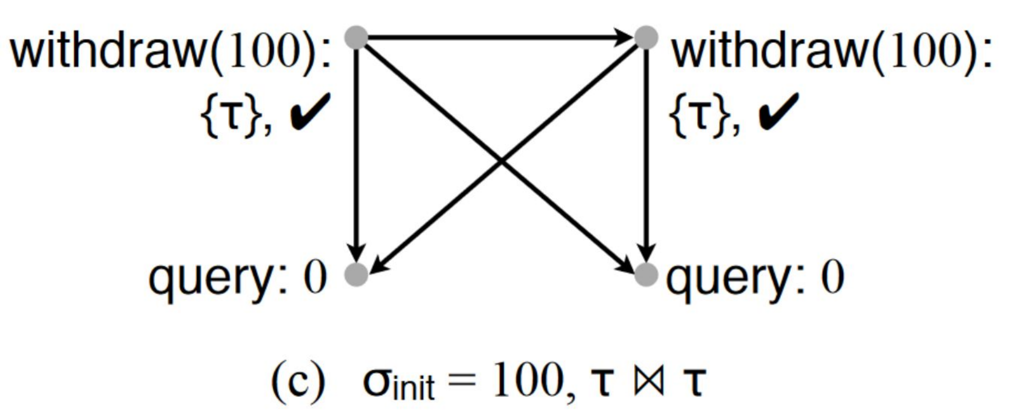



Figure 2C
Example of Definition 1
for Withdraw and Query

# Example

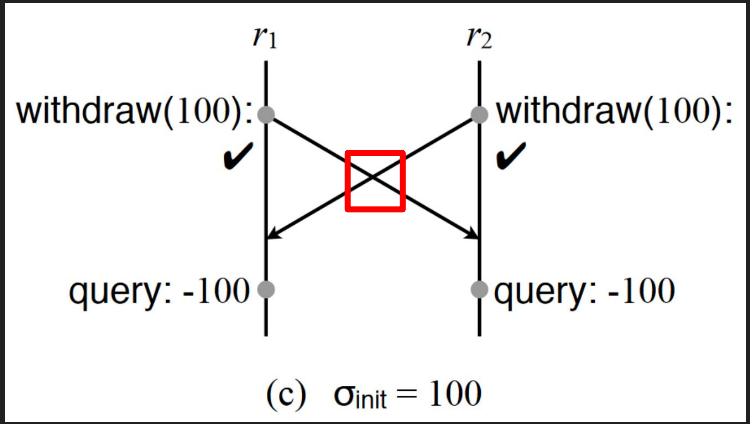



Figure 1C
Illustration of Withdraw and Query

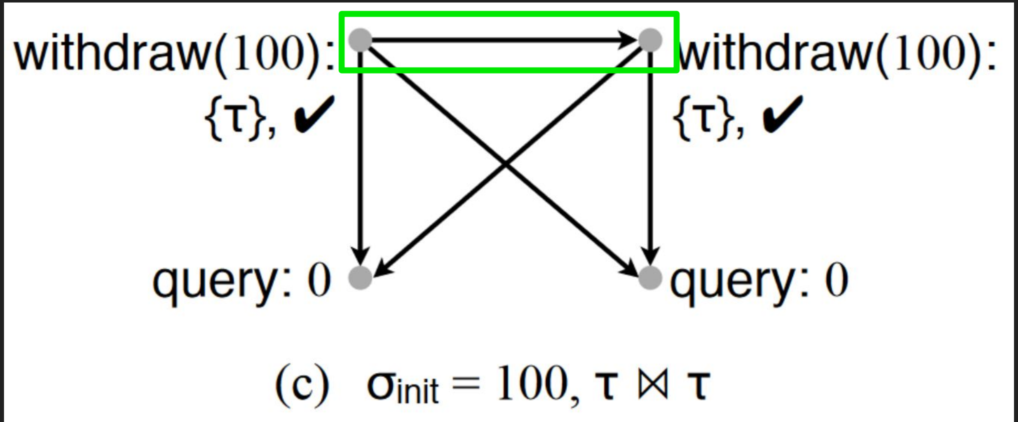



Figure 2C
Example of Definition 1
for Withdraw and Query

# Example

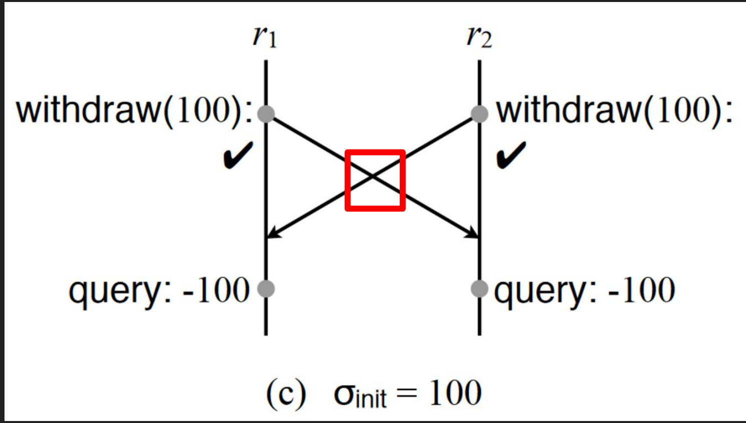



Figure 1C
Illustration of Withdraw and Query

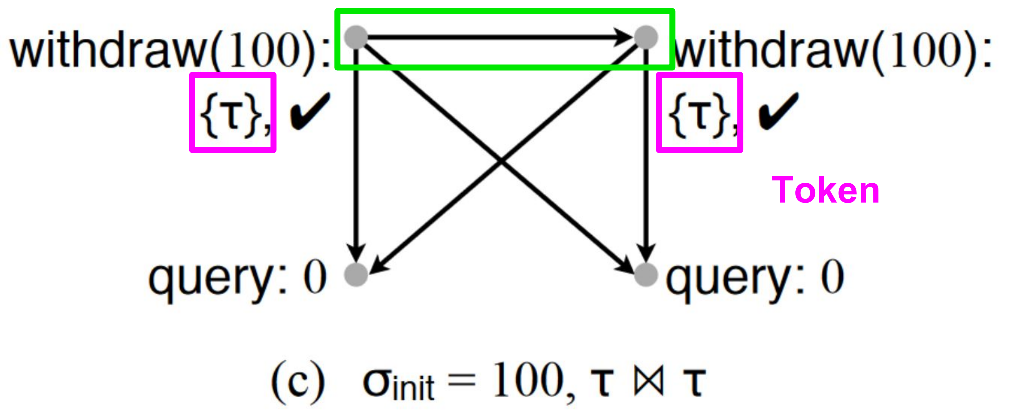



Figure 2C
Example of Definition 1
for Withdraw and Query

# Definitions and Notations - Extensions

$$I = \{\text{subset of State}\} \qquad\qquad T = (\text{Token}, \bowtie)$$

$$F \in \text{Op} \rightarrow (\text{State} \rightarrow (\text{Val} \times (\text{State} \rightarrow \text{State}) \times \mathbb{P}(\text{Token})))$$

$$F_o(\sigma) \quad = \quad (\text{Val}\,,\ (\text{State} \rightarrow \text{State})\,,\ \mathbb{P}(\text{Token}))$$

$$F_o(\sigma) \quad = \quad (F_o^{val}(\sigma)\,,\ (\ F_o^{eff}(\sigma)\ )\ ,\ \ F_o^{tok}(\sigma)\ )$$

# Definitions and Notations - Commutativity

$$F^{eff}_{o1}(\sigma 1) \circ F^{eff}_{o2}(\sigma 2) = F^{eff}_{o2}(\sigma 2) \circ F^{eff}_{o1}(\sigma 1)$$

# Definitions and Notations - Commutativity

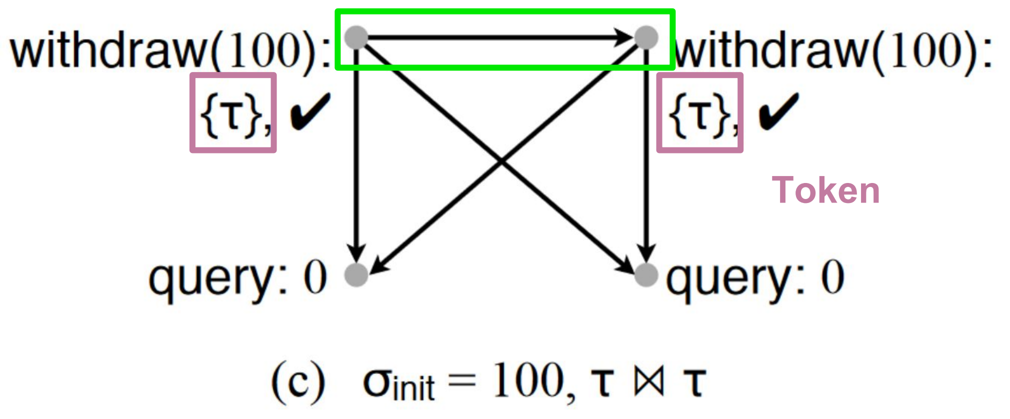
$$\left( F^{tok}_{o1}(\sigma 1) \bowtie F^{tok}_{o2}(\sigma 2) \right) \vee$$

$$\left[ F^{eff}_{o1}(\sigma 1) \circ F^{eff}_{o2}(\sigma 2) = F^{eff}_{o2}(\sigma 2) \circ F^{eff}_{o1}(\sigma 1) \right]$$

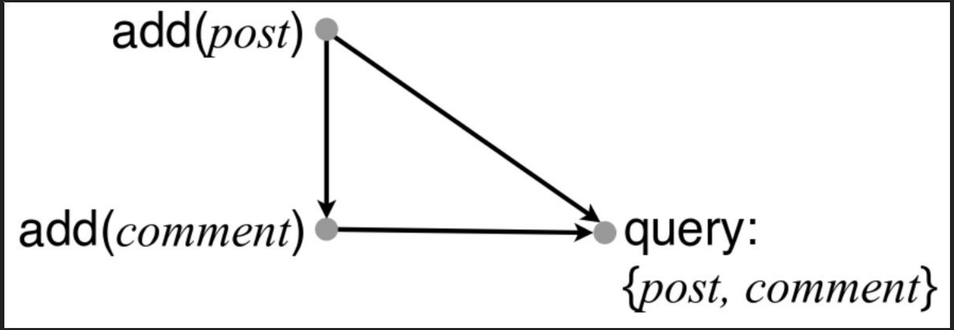
# Definitions and Notations - Extensions

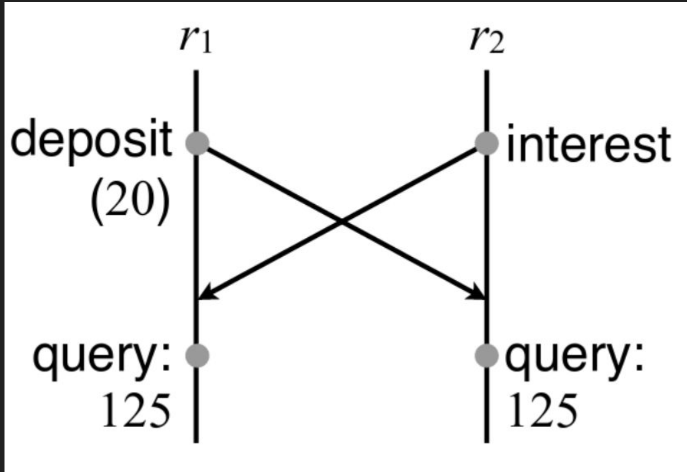
$$F_{withdraw(a)}(\sigma) = \begin{cases} (✓, (\lambda\sigma'.\sigma' - a), T_w), & \text{if } \sigma \geq a \\ (✗, skip, T_w), & \text{else} \end{cases}$$

Figure 2C
Example of Definition 1
for Withdraw and Query

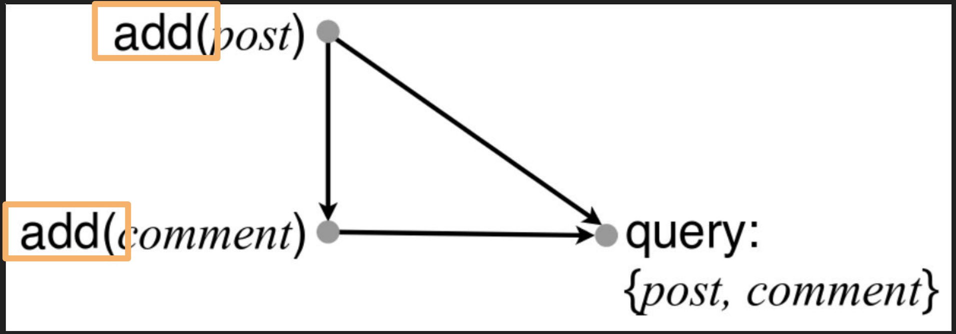


# Intuition

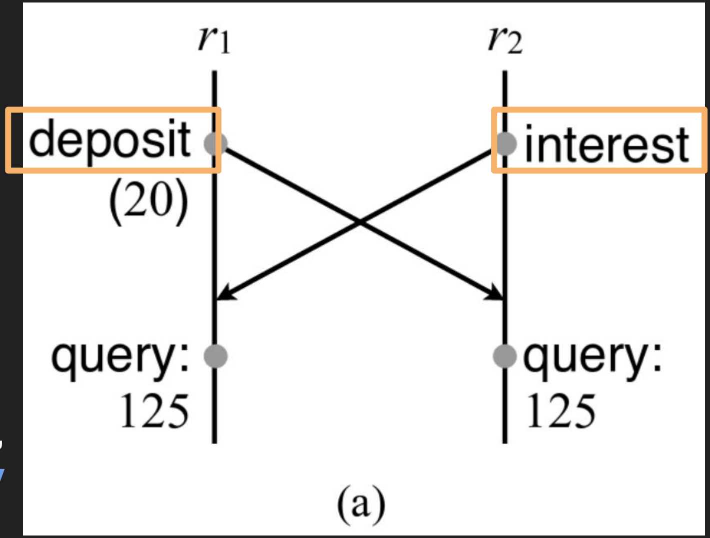

add(*post*)
add(*comment*)
query:
{*post*, *comment*}


$r_1$
$r_2$
deposit
(20)
interest
query:
125
query:
125
(a)

# Intuition




If operations **are commutative**,
then tokens **are not necessary**




If operations **are not commutative**,
then tokens **are necessary**

# State-based Proof

$\exists G_0 \in \mathcal{P}(\mathsf{State} \times \mathsf{State}), G \in \mathsf{Token} \to \mathcal{P}(\mathsf{State} \times \mathsf{State})$
such that

S1. $\sigma_{\mathsf{init}} \in I$

S2. $G_0(I) \subseteq I \wedge \forall \tau.\ G(\tau)(I) \subseteq I$

S3. $\forall o, \sigma, \sigma'.\ (\sigma \in I \wedge (\sigma, \sigma') \in (G_0 \cup G((\mathcal{F}_o^{\mathsf{tok}}(\sigma))^{\perp}))^*)$
$$\implies (\sigma', \mathcal{F}_o^{\mathsf{eff}}(\sigma)(\sigma')) \in G_0 \cup G(\mathcal{F}_o^{\mathsf{tok}}(\sigma))$$

---

$$\mathsf{Exec}(\mathcal{T}, \mathcal{F}) \subseteq \mathsf{eval}_{\mathcal{F}}^{-1}(I)$$

# State-based Proof

$$\exists G_0 \in \mathcal{P}(\mathsf{State} \times \mathsf{State}), G \in \mathsf{Token} \to \mathcal{P}(\mathsf{State} \times \mathsf{State})$$

such that

S1. $\sigma_{\mathsf{init}} \in I$

S2. $G_0(I) \subseteq I \wedge \forall \tau.\ G(\tau)(I) \subseteq I$

S3. $\forall o, \sigma, \sigma'.\ (\sigma \in I \wedge (\sigma, \sigma') \in (G_0 \cup G((\mathcal{F}_o^{\mathsf{tok}}(\sigma))^{\perp}))^*)$

$$\Longrightarrow (\sigma', \mathcal{F}_o^{\mathsf{eff}}(\sigma)(\sigma')) \in G_0 \cup G(\mathcal{F}_o^{\mathsf{tok}}(\sigma))$$

$$\overline{\mathsf{Exec}(\mathcal{T}, \mathcal{F}) \subseteq \mathsf{eval}_{\mathcal{F}}^{-1}(I)}$$

S1. The initial state satisfies the invariant

S2. Causally consistent operations satisfy the invariant
**AND** all possible state changes that use synchronization satisfy the invariant

S3. **IF** the origin state and replica state are in the guaranteed possible state changes,
**THEN** the state change from the effect function must be a guaranteed possible state change

# State-based Proof

$\exists G_0 \in \mathcal{P}(\mathsf{State} \times \mathsf{State}), G \in \mathsf{Token} \to \mathcal{P}(\mathsf{State} \times \mathsf{State})$
such that

S1. $\sigma_{\mathsf{init}} \in I$

S2. $G_0(I) \subseteq I \land \forall \tau.\ G(\tau)(I) \subseteq I$

S3. $\forall o, \sigma, \sigma'.\ (\sigma \in I \land (\sigma, \sigma') \in (G_0 \cup G(\boxed{(\mathcal{F}_o^{\mathsf{tok}}(\sigma))^{\perp}}))^*)$
$\implies (\sigma', \mathcal{F}_o^{\mathsf{eff}}(\sigma)(\sigma')) \in G_0 \cup G(\mathcal{F}_o^{\mathsf{tok}}(\sigma))$

$$\frac{}{\mathsf{Exec}(\mathcal{T}, \mathcal{F}) \subseteq \mathsf{eval}_{\mathcal{F}}^{-1}(I)}$$

$$\boxed{T^{\perp} = \{\tau \mid \tau \in \mathsf{Token} \land \neg\exists \tau' \in T.\ \tau \bowtie \tau'\}}$$

# State-based Proof

$\exists G_0 \in \mathcal{P}(\mathsf{State} \times \mathsf{State}), G \in \mathsf{Token} \to \mathcal{P}(\mathsf{State} \times \mathsf{State})$
such that

$$\frac{\begin{array}{l} \text{S1. } \sigma_{\mathsf{init}} \in I \\ \text{S2. } G_0(I) \subseteq I \land \forall \tau.\ G(\tau)(I) \subseteq I \\ \text{S3. } \forall o, \sigma, \sigma'.\ (\sigma \in I \land (\sigma, \sigma') \in (G_0 \cup \boxed{G((\mathcal{F}_o^{\mathsf{tok}}(\sigma))^{\perp})})^*) \\ \qquad \implies (\sigma', \mathcal{F}_o^{\mathsf{eff}}(\sigma)(\sigma')) \in G_0 \cup G(\mathcal{F}_o^{\mathsf{tok}}(\sigma)) \end{array}}{\mathsf{Exec}(\mathcal{T}, \mathcal{F}) \subseteq \mathsf{eval}_{\mathcal{F}}^{-1}(I)}$$

$$T^{\perp} = \{\tau \mid \tau \in \mathsf{Token} \land \neg\exists \tau' \in T.\ \tau \bowtie \tau'\}$$

$$\boxed{G(T) = \bigcup_{\tau \in T} G(\tau)}$$

# State-based Proof

$\exists G_0 \in \mathcal{P}(\mathsf{State} \times \mathsf{State}), G \in \mathsf{Token} \to \mathcal{P}(\mathsf{State} \times \mathsf{State})$
such that

S1. $\sigma_{\mathsf{init}} \in I$

S2. $G_0(I) \subseteq I \wedge \forall \tau.\ G(\tau)(I) \subseteq I$

S3. $\forall o, \sigma, \sigma'.\ (\sigma \in I \wedge (\sigma, \sigma') \in \boxed{(G_0 \cup G((\mathcal{F}^{\mathsf{tok}}_o(\sigma))^{\perp}))}^{*}) \implies (\sigma', \mathcal{F}^{\mathsf{eff}}_o(\sigma)(\sigma')) \in G_0 \cup G(\mathcal{F}^{\mathsf{tok}}_o(\sigma))$

Reflexive and transitive closure

$$\overline{\mathsf{Exec}(\mathcal{T}, \mathcal{F}) \subseteq \mathsf{eval}^{-1}_{\mathcal{F}}(I)}$$

$$T^{\perp} = \{\tau \mid \tau \in \mathsf{Token} \wedge \neg\exists \tau' \in T.\ \tau \bowtie \tau'\}$$

$$G(T) = \bigcup_{\tau \in T} G(\tau)$$

# Event-based Proof

$\exists \mathbb{G} \in \mathcal{P}(\mathsf{Exec}(\mathcal{T}) \times \mathsf{Exec}(\mathcal{T}))$ such that

E1. $X_{\mathsf{init}} \in \mathbb{I}$

E2. $\mathbb{G}(\mathbb{I}) \subseteq \mathbb{I}$

E3. $\forall X, X', X''.\ \forall e \in X''.E.$

$$(X \in \mathbb{I} \wedge X' = X''|_{X''.E-\{e\}} \wedge X'' \in \mathsf{Exec}(\mathcal{T}, \mathcal{F}) \wedge e \in \max(X'') \wedge X = \mathsf{ctxt}(e, X'') \wedge (X, X') \in \mathbb{G}^*) \implies (X', X'') \in \mathbb{G}$$

$$\overline{\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad}$$

$$\mathsf{Exec}(\mathcal{T}, \mathcal{F}) \subseteq \mathbb{I}$$

# What are the takeaways?