LINEARIZABILITY

THIS ONE IS IMPORTANT

WHERE WE ARE

- Traces ← What the Paper Called a History
- Safety and Liveness ← Constraints on Correct Traces

TODAY

- One Possible Set of Constraints?

BACKGROUND
Lecture 1: "We will talk about distributed protocols for distributed systems."

What are data types doing here?

Distributed system

Key Value Store

Chord

Everything is really about data and how it evolves.

Linearizability: Order in which events seem to occur.

Two problems:

What orders should we allow (today)?

How to ensure allowed orders?
**Partial Order**

- $e_1 < e_2$ or $e_1 = e_2$

**Total Order**

- $e_1 < e_2$ or $e_2 < e_1$

**Transitive**

- If $e_1 < e_2$ and $e_2 < e_3$, then $e_1 < e_3$

**Asymmetry**

- If $e_1 < e_2$ and $e_2 < e_1$, then $e_1 = e_2$

**Intransitive**

- If $e_1 < e_2$ and $e_2 < e_3$, then $e_1 < e_3$

**Ordering Relations**

- $<$ (strict order)
- $\leq$ (less than or equal to)
- $\geq$ (greater than or equal to)

**Example of Total Order**

- $(e, e) \notin \mathcal{E}$ and $(e, e) \notin \mathcal{E}$

**(Rest of the semester)**
Examples of Partial Order

Linearizability

Sequential Specification [FIFO]

Many Concurrent Users
Linearizability: Given \( h \) (and \( <_h \)) we can

- Extend \( h \) to \( h' \) s.t. \( h' \) is complete

- Exists sequential history \( S(<_s) \)

  \[ \rightarrow \]

  \[ \rightarrow <_h \subseteq <_s \]

  \[ \rightarrow \text{meets sequential spec} \]

1. Complete

   Every Invoke Has a Return.
(2) \( \exists \text{ Sequenital History } S, < \)

\[ \text{ENC(1)} \quad \text{OK(1)} \]
\[ \text{ENC(2)} \quad \text{OK(1)} \quad \text{DEQ(1)} \quad \text{OK(2)} \]

\[ S \]

- \( <_H \subseteq <_S \)
- \( <_S \text{ Total ORDER} \)
- \( S \text{ meets SPEC} \)

Is \( S \) UNIQUE?
**LINEARIZABLE?**

**Detour: Sequential Consistency [Need this to talk about locality]**

\[ \preceq_H \quad \preceq_S \quad \preceq_P : \quad \left\{ \text{event} \right\} \xrightarrow{ \text{H} } \quad \preceq_H \quad \text{when applied only to events from a process} \]

**SEQ CST** \[ \exists \preceq_S \quad \text{such that} \]

\[ \forall \preceq_P : \quad \preceq_P \subseteq \preceq_S \]
**Linearizability vs Seq Cst**

\[
\preceq_s \subseteq \preceq_h
\]

Linearizable

\[
H_p \preceq_p \leq \preceq_s
\]

\[
\preceq_p = \preceq_{H1_p}
\]

**Locality**

Distributed System History \( H \) Linearizable

\[
\iff
\]

History for all objects are Linearizable
**Proof Sketch**

\[ H \text{ LINEARIZABLE} \iff \forall O. H_0 \text{ LINEARIZABLE} \]

**Given**

\[ H, <_H, H_0, <_{H_0}, \ldots \]

For each \( H_0 \) (\( H_0 \)) can find

\[ H'_0 = H_0 + R_0 \]

and \( S_0, <_S, \text{ s.t. } <_S \subseteq <_{H_0} \)

**Construct**

\[ H' = H + \sum R_0 \]

\[ <_{H'} = \text{Closure}(<_H \cup (U, <_{H_0})) \]

\[ \text{L} \rightarrow \text{Add Transitive Edges} \]
CLAIMS

1. < is a partial order

2. Any S constructed by ordering complete (H') st. < ≤ ≤s is a linearization of H

1. <h' is a partial order

   ⇒ Irreflexive ∙ by construction

   ⇒ Transitive ∙ by construction

   ⇒ Antisymmetric ← there are no cycles

Show not the case by contradiction

Assume \( e_1, e_2, \ldots, e_n \) st. \( e_1 < e_2 < e_3 < e_4 < \ldots < e_n \) and \( e_n < e_1 \)

i. \( e_1, e_2, \ldots, e_n \) cannot all be ops on the same object

\[ <_{h'} < < \]

\[ \Rightarrow \text{partial order by assumption} \]

2. Pick the smallest cycle \( e_1, e_2, \ldots, e_n \) st. \( e_1, e_2 \) are ops on different objects

   a. If \( e_1 \) is an operation on \( O \) no other \( e_2, \ldots, e_n \) is an operation on \( O \)

Assume otherwise so \( e_1, e_i \) are both on \( O \)

\[ e_{i-1} < e_i \text{ ReI}(e_{i-1}) \text{ before} \text{ Inv}(e_i) \]
$e_2 \prec e_{i-1}$ \hspace{1cm} Inv$(e_2)$ Before Ret$(e_{i-1})$

$e_i \prec e_2$ \hspace{1cm} Ret$(e_i)$ Before Inv$(e_2)$

$\Rightarrow$ Ret$(e_i)$ Before Ret$(e_{i-1})$ Before Inv$(e_i)$

$\Rightarrow$ $e_i \ prec_h e_i \Rightarrow e_i \prec e_i$

$\Rightarrow$ $e_i, e_i, e_{i+1}, \ldots, e_n$ Is A Smaller Cycle

CONTRADICTION

If $e_i$ is an op on 0 then $e_2, \ldots, e_n$ are not an op on 0

$\Rightarrow$ $e_n \prec_h e_i$

But $e_i \prec_h e_2$ & $\prec_h$ is TRANSITIVE

$\Rightarrow$ $e_n \prec_h e_2$

$\Rightarrow$ $e_2, \ldots, e_n$ Is A Smaller Cycle

CONTRADICTION

SEQ Consistency IS NOT LOCAL

---

A

\[\text{p.enq}(x) \quad q.enq(y) \quad \text{p.deq}()\]

\[\text{ok} \quad \text{ok} \quad \text{OK} \quad \text{OKCY}\]

B

\[\text{q.enq}(y) \quad \text{p.enq}(y) \quad \text{q.deq}()\]

\[\text{OK} \quad \text{OK} \quad \text{OKG}\]

---

\[\text{q.enq}(x)\]

\[\text{q.enq}(y)\]

\[\text{q.deq}()\]

---

A

\[\text{p.enq}(x) \quad q.enq(y) \quad \text{p.deq}()\]

\[\text{ok} \quad \text{ok} \quad \text{OK} \quad \text{OKCY}\]

B

\[\text{q.enq}(y) \quad \text{p.enq}(y) \quad \text{q.deq}()\]

\[\text{OK} \quad \text{OK} \quad \text{OKG}\]

---

\[\text{p.enq}(x)\]

\[\text{q.enq}(y)\]

\[\text{q.deq}()\]
**Non-Blocking**

- Linearizability does not require ops to wait

Alternate: \( I : o_1 < I o_2 \iff \text{Inv}(o_1) \text{ before } \text{Inv}(o_2) \)

Composing linearizable types
- Locality?
- Why hard?
CAP Theorem

Context

Statement

Consistency (Linearizability) ← Network Partitions → Availability

Inktony/Now
Proof by Indistinguishability

Alice \( \xrightarrow{\text{eng}(1)} \) BETH \( \xrightarrow{\text{def} \circ 0 \mid <(1)} \) \( \text{eng}(3) \)

A \( \xrightarrow{\text{eng}(3)} \) B \( \xrightarrow{\text{eng}(1) \circ \text{OK}} \) \( \text{eng}(3) \)

Alice \( \xrightarrow{\text{eng}(1)} \) BETH \( \xrightarrow{\text{OK}} \) \( \text{deq} \)

\( e_1, e_2, d_1 \) \( \times \)