Relaxing the Consistency Condition

Relaxing the Consistency Condition

Local Linearizability (CONCUR16)

Local Linearizability main idea

Local Linearizability main idea

- Partition a history into a set of local histories
- Require linearizability per local history

Local Linearizability main idea

Already present in some shared-memory consistency conditions
(not in our form of choice)

- Partition a history into a set of local histories

- Require linearizability per local history

Local Linearizability main idea

Already present in some shared-memory consistency conditions
(not in our form of choice)

- Partition a history into a set of local histories

- Require linearizability per local history

Local sequential consistency… is also possible

Local Linearizability main idea

Already present in some shared-memory consistency conditions
(not in our form of choice)

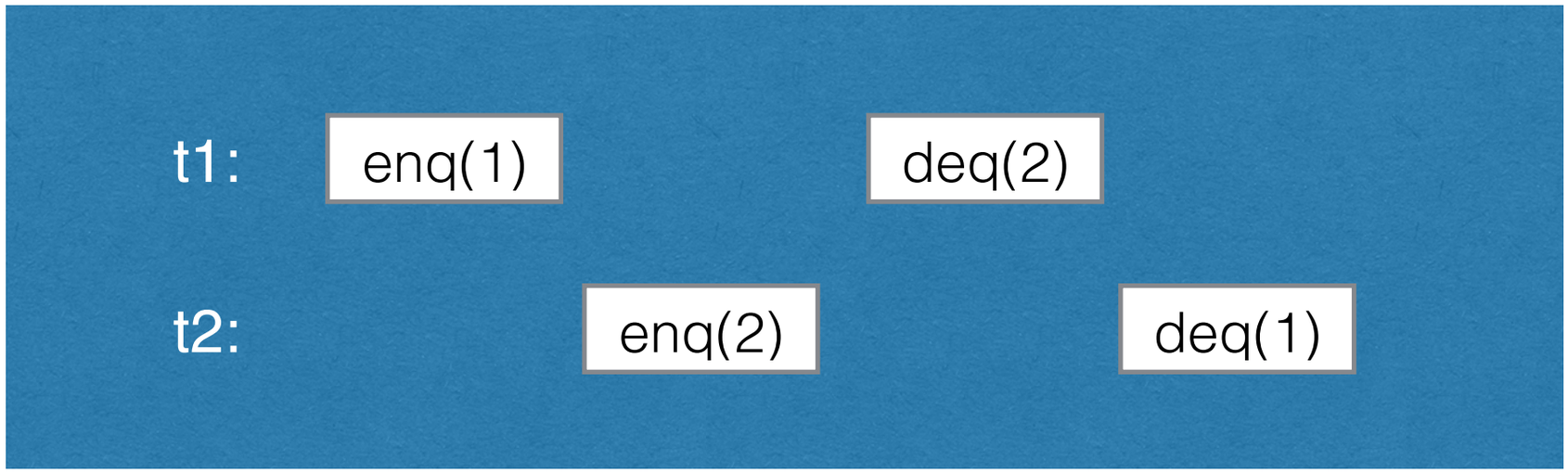
- Partition a history into a set of local histories

- Require linearizability per local history

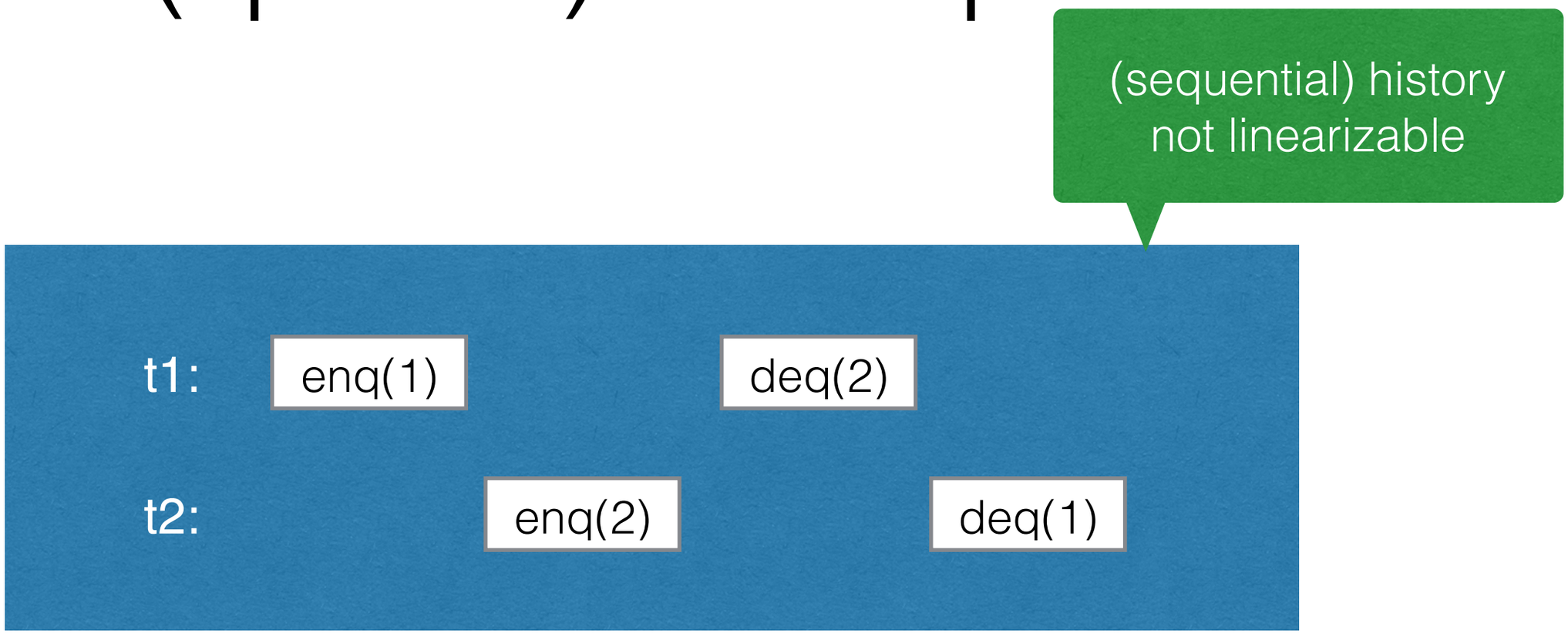
no global witness

Local sequential consistency… is also possible

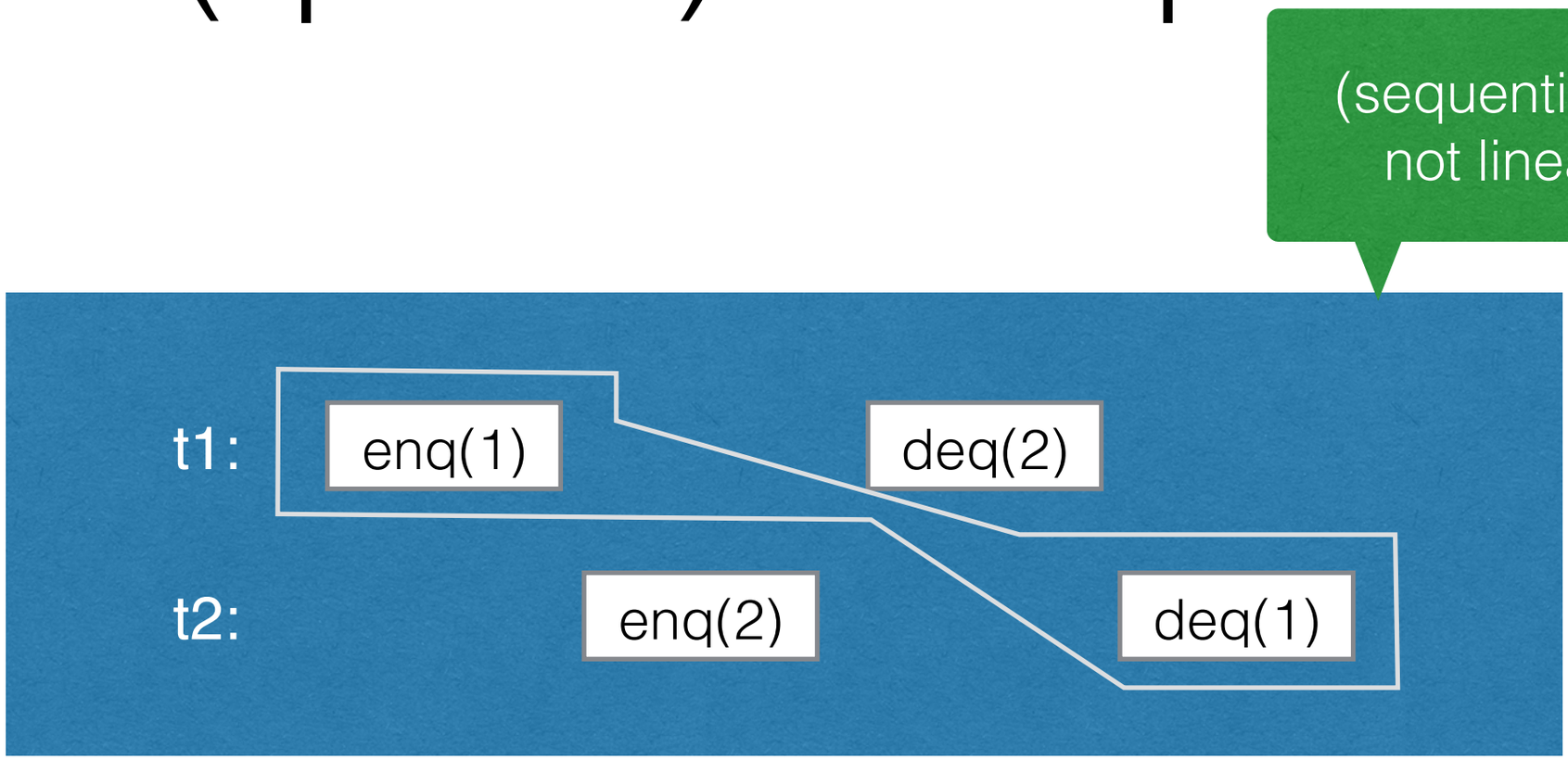
Local Linearizability (queue) example



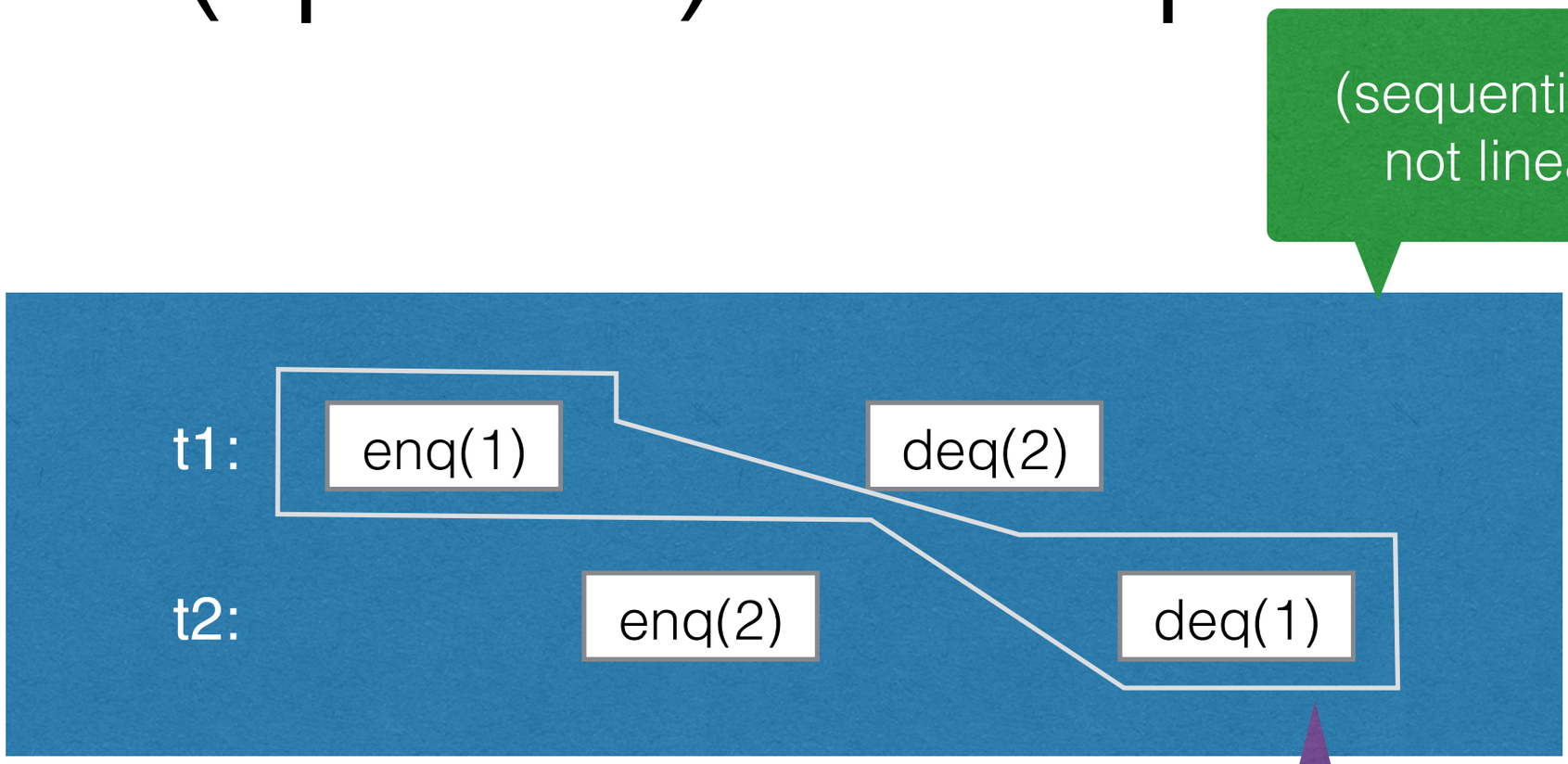
Local Linearizability (queue) example



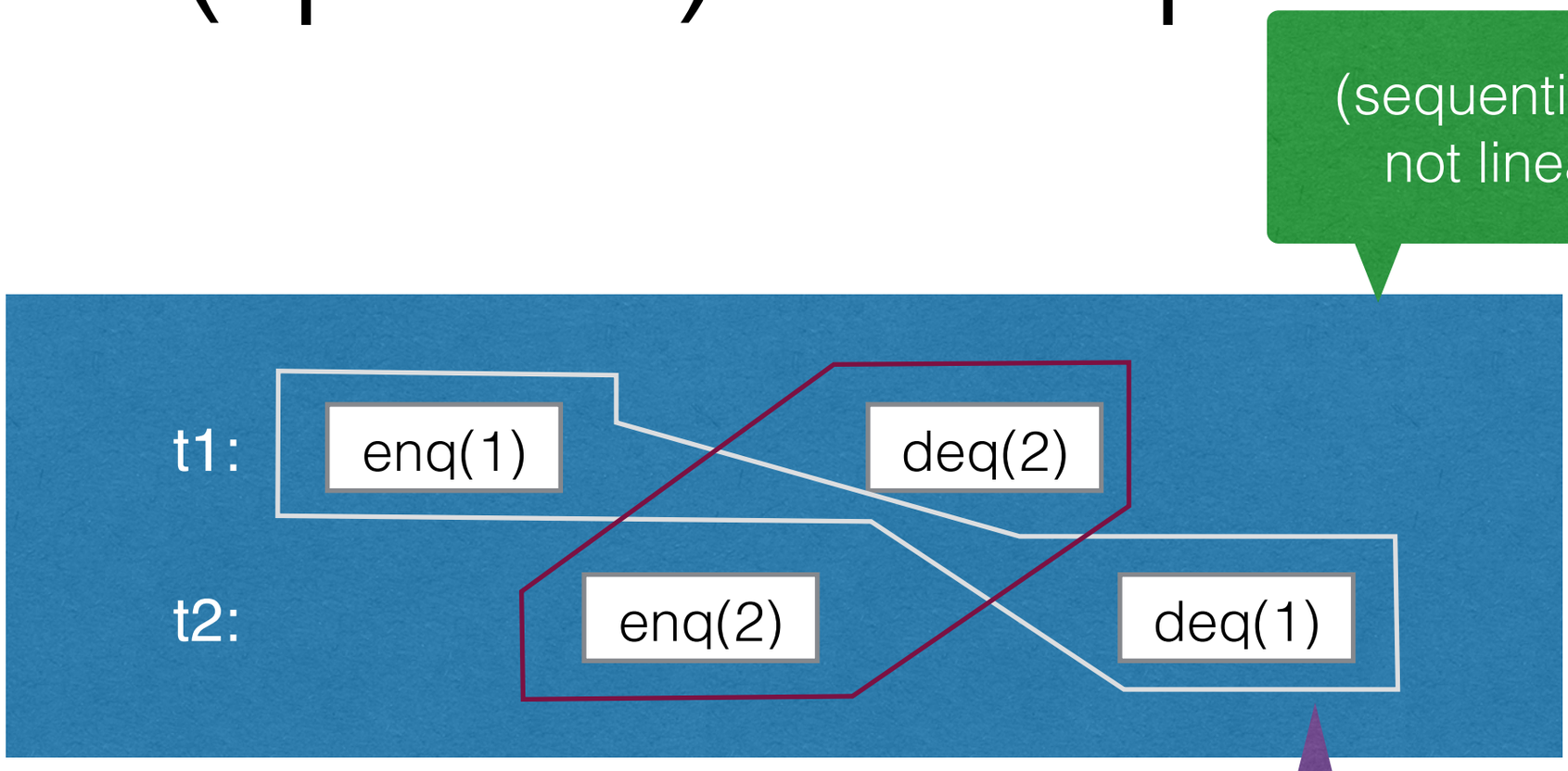
Local Linearizability (queue) example



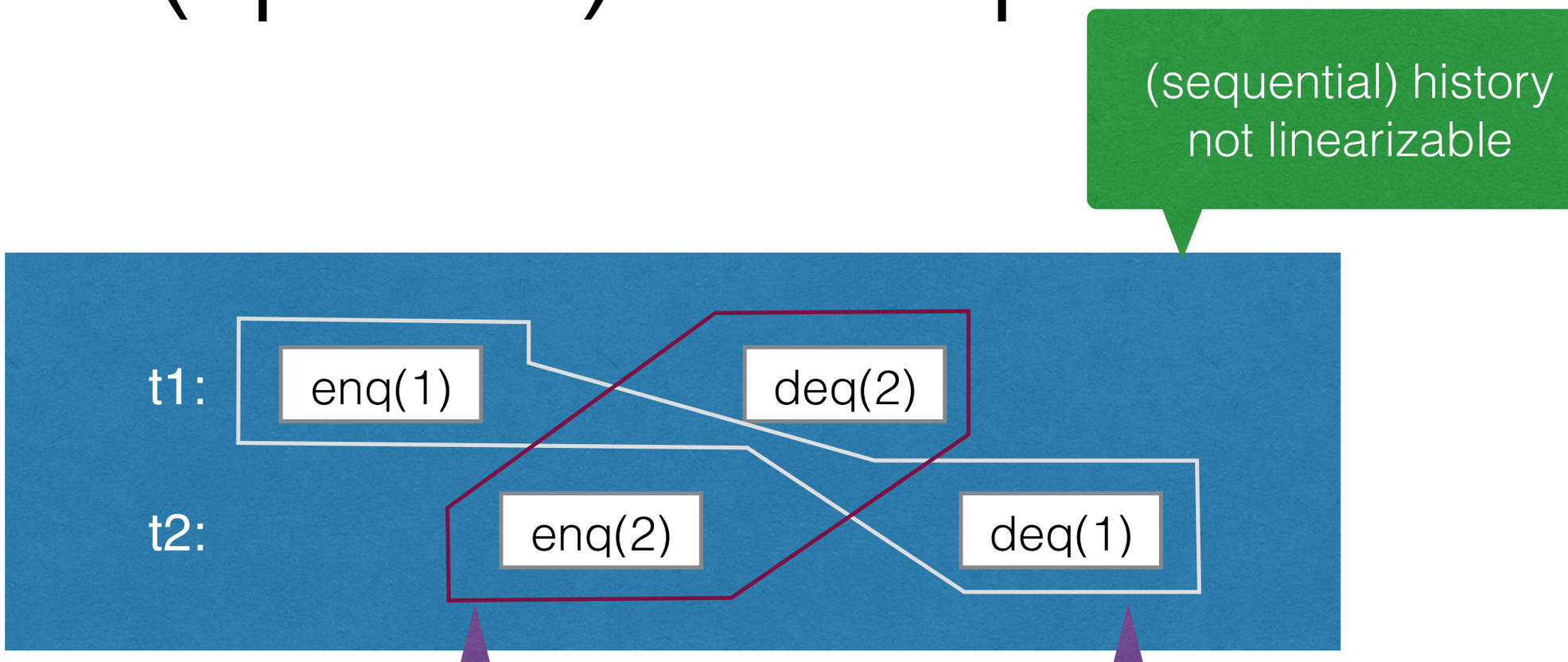
Local Linearizability (queue) example



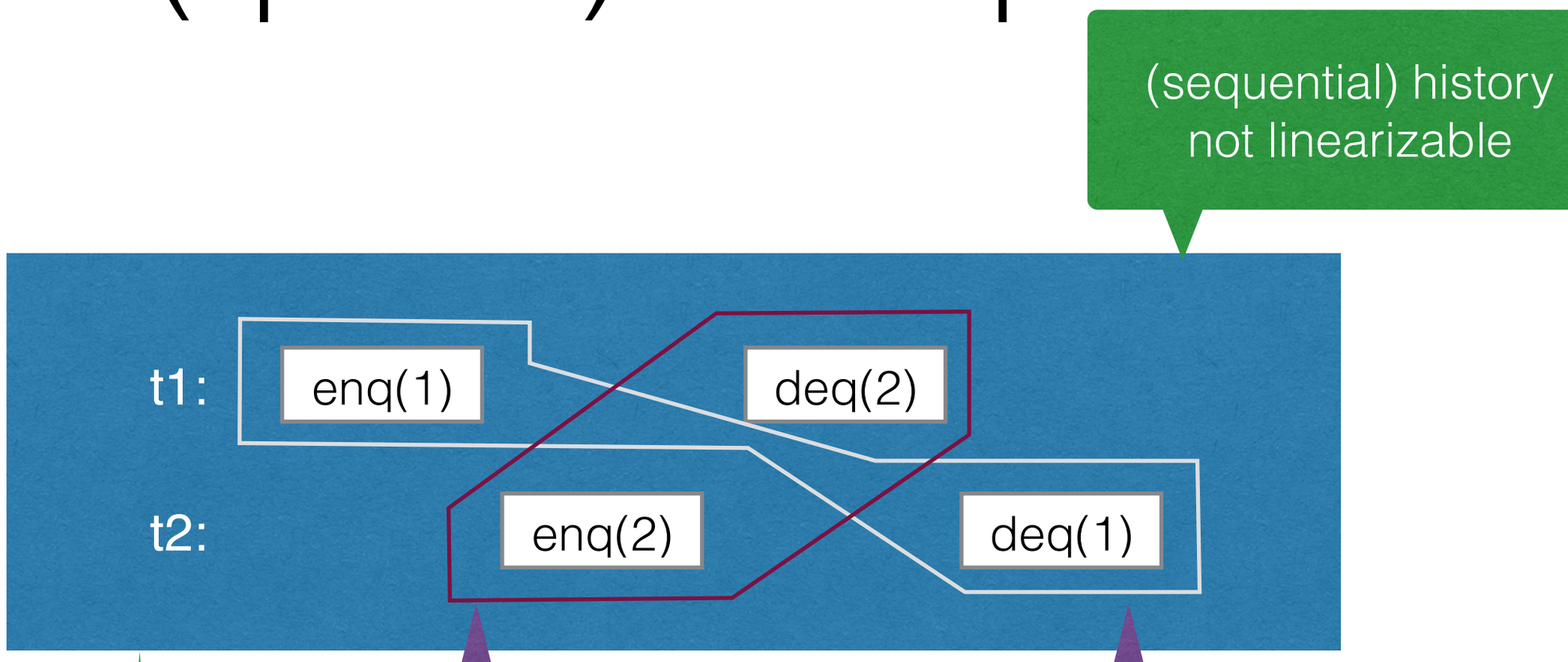
Local Linearizability (queue) example



Local Linearizability (queue) example



Local Linearizability (queue) example



Local Linearizability (queue) definition

Local Linearizability (queue) definition

Queue signature $\Sigma = \{enq(x) \mid x \in V\} \cup \{deq(x) \mid x \in V\} \cup \{deq(empty)\}$

Local Linearizability (queue) definition

Queue signature $\Sigma = \{enq(x) \mid x \in V\} \cup \{deq(x) \mid x \in V\} \cup \{deq(empty)\}$

For a history **h** with a thread T, we put

$$I_T = \{enq(x)^T \in \mathbf{h} \mid x \in V\}$$

$$O_T = \{deq(x)^{T'} \in \mathbf{h} \mid enq(x)^T \in I_T\} \cup \{deq(empty)\}$$

Local Linearizability (queue) definition

Queue signature $\Sigma = \{enq(x) \mid x \in V\} \cup \{deq(x) \mid x \in V\} \cup \{deq(empty)\}$

For a history **h** with a thread T, we put

$I_T = \{enq(x)^T \in \mathbf{h} \mid x \in V\}$

$O_T = \{deq(x)^{T'} \in \mathbf{h} \mid enq(x)^T \in I_T\} \cup \{deq(empty)\}$

in-methods of thread T are enqueues performed by thread T

Local Linearizability (queue) definition

Queue signature $\Sigma = \{enq(x) \mid x \in V\} \cup \{deq(x) \mid x \in V\} \cup \{deq(empty)\}$

For a history **h** with a thread T, we put

$I_T = \{enq(x)^T \in \mathbf{h} \mid x \in V\}$

in-methods of thread T are enqueues performed by thread T

$O_T = \{deq(x)^{T'} \in \mathbf{h} \mid enq(x)^T \in I_T\} \cup \{deq(empty)\}$

out-methods of thread T are dequeues (performed by any thread) corresponding to enqueues that are in-methods

Local Linearizability (queue) definition

Queue signature $\Sigma = \{enq(x) \mid x \in V\} \cup \{deq(x) \mid x \in V\} \cup \{deq(empty)\}$

For a history **h** with a thread T, we put

$I_T = \{enq(x)^T \in \mathbf{h} \mid x \in V\}$

in-methods of thread T are enqueues performed by thread T

$O_T = \{deq(x)^{T'} \in \mathbf{h} \mid enq(x)^T \in I_T\} \cup \{deq(empty)\}$

out-methods of thread T are dequeues (performed by any thread) corresponding to enqueues that are in-methods

h is locally linearizable iff every thread-induced history

$\mathbf{h}_T = \mathbf{h} \mid (I_T \cup O_T)$

is linearizable.

Local Linearizability for Container-Type DS

Local Linearizability for Container-Type DS

Signature $\sum$ = Ins ∪ Rem ∪ SOb ∪ DOb

Local Linearizability for Container-Type DS

Signature $\sum$ = Ins ∪ Rem ∪ SOb ∪ DOb

For a history **h** with a thread T, we put

$$I_T = \{m^T \in \mathbf{h} \mid m \in Ins\}$$

$$O_T = \{m(a) \in \mathbf{h} \cap Rem \mid i(a)^T \in I_T\} \cup \{m(e) \mid e \in Emp\} \cup \{m(a) \in \mathbf{h} \cap DOb \mid i(a)^T \in I_T\}$$

Local Linearizability for Container-Type DS

Signature $\sum$ = Ins ∪ Rem ∪ SOb ∪ DOb

For a history **h** with a thread T, we put

$I_T = \{m^T \in \mathbf{h} \mid m \in Ins\}$

in-methods of thread T are inserts performed by thread T

$O_T = \{m(a) \in \mathbf{h} \cap Rem \mid i(a)^T \in I_T\} \cup \{m(e) \mid e \in Emp\}$
$\cup \{m(a) \in \mathbf{h} \cap DOb \mid i(a)^T \in I_T\}$

Local Linearizability for Container-Type DS

Signature $\sum$ = Ins ∪ Rem ∪ SOb ∪ DOb

For a history **h** with a thread T, we put

$I_T = \{m^T \in \mathbf{h} \mid m \in Ins\}$

in-methods of thread T are inserts performed by thread T

$O_T = \{m(a) \in \mathbf{h} \cap Rem \mid i(a)^T \in I_T\} \cup \{m(e) \mid e \in Emp\}$
$\cup \{m(a) \in \mathbf{h} \cap DOb \mid i(a)^T \in I_T\}$

out-methods of thread T are removes and data-observations (performed by any thread) corresponding to in-methods

Local Linearizability for Container-Type DS

Signature $\sum$ = Ins ∪ Rem ∪ SOb ∪ DOb

For a history **h** with a thread T, we put

$I_T = \{m^T \in \mathbf{h} \mid m \in Ins\}$

in-methods of thread T are inserts performed by thread T

$O_T = \{m(a) \in \mathbf{h} \cap Rem \mid i(a)^T \in I_T\} \cup \{m(e) \mid e \in Emp\}$
$\cup \{m(a) \in \mathbf{h} \cap DOb \mid i(a)^T \in I_T\}$

out-methods of thread T are removes and data-observations (performed by any thread) -methods

h is locally linearizable iff every thread-induced history

$\mathbf{h}_T = \mathbf{h} \mid (I_T \cup O_T)$

is linearizable.

Generalizations of Local Linearizability

Generalizations of Local Linearizability

Signature Σ

Generalizations of Local Linearizability

Signature ∑

For a history **h** with n threads, choose

$In_{\mathbf{h}}(i)$

$Out_{\mathbf{h}}(i)$

Generalizations of Local Linearizability

Signature ∑

For a history **h** with n threads, choose

$In_{\mathbf{h}}(i)$

$Out_{\mathbf{h}}(i)$

in-methods of thread i, methods that go in $\mathbf{h}_i$

Generalizations of Local Linearizability

Signature ∑

For a history **h** with n threads, choose

$In_{\mathbf{h}}(i)$

$Out_{\mathbf{h}}(i)$

in-methods of thread i, methods that go in $\mathbf{h}_i$

out-methods of thread i, dependent methods on the methods in $In_{\mathbf{h}}(i)$ (performed by any thread)

Generalizations of Local Linearizability

Signature ∑

For a history **h** with n threads, choose

$In_{\mathbf{h}}(i)$ — in-methods of thread i, methods that go in $\mathbf{h}_i$

$Out_{\mathbf{h}}(i)$ — out-methods of thread i, dependent methods on the methods in $In_{\mathbf{h}}(i)$ (performed by any thread)

h is locally linearizable iff every thread-induced history

$$\mathbf{h}_i = \mathbf{h} \mid (In_{\mathbf{h}}(i) \cup Out_{\mathbf{h}}(i))$$

is linearizable.

Generalizations of Local Linearizability

Signature ∑

For a history **h** with n threads, choose

$In_{\mathbf{h}}(i)$

$Out_{\mathbf{h}}(i)$

in-methods of thread i, methods that go in $\mathbf{h}_i$

by increasing the in-methods, LL gradually moves to linearizability

out-methods of thread i, dependent methods on the methods in $In_{\mathbf{h}}(i)$ (performed by any thread)

h is locally linearizable iff every thread-induced history

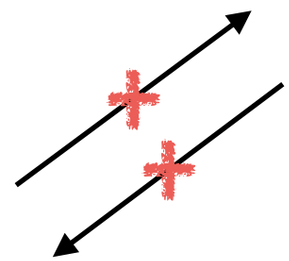
$$\mathbf{h}_i = \mathbf{h} \mid (In_{\mathbf{h}}(i) \cup Out_{\mathbf{h}}(i))$$

is linearizable.

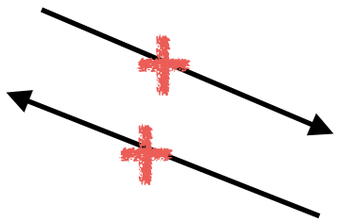
Where do we stand?

In general

Linearizability



Local Linearizability



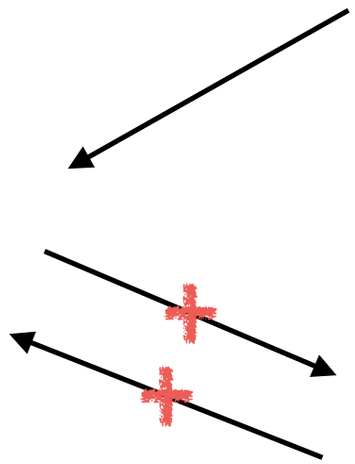
Sequential Consistency

Where do we stand?

For queues (and most container-type data structures)

Linearizability

Local Linearizability



Sequential Consistency

Properties

Properties

Local linearizability is compositional

Properties

Local linearizability is compositional

like linearizability
unlike sequential consistency

Properties

Local linearizability is compositional

like linearizability
unlike sequential consistency

h (over multiple objects) is locally linearizable
iff
each per-object subhistory of **h** is locally linearizable

Properties

Local linearizability is compositional

like linearizability
unlike sequential consistency

h (over multiple objects) is locally linearizable
iff
each per-object subhistory of **h** is locally linearizable

Local linearizability is modular / "decompositional"

Properties

Local linearizability is compositional

like linearizability
unlike sequential consistency

h (over multiple objects) is locally linearizable
iff
each per-object subhistory of **h** is locally linearizable

Local linearizability is modular / "decompositional"

uses decomposition into smaller histories, by definition

Properties

Local linearizability is compositional

like linearizability
unlike sequential consistency

h (over multiple objects) is locally linearizable
iff
each per-object subhistory of **h** is locally linearizable

Local linearizability is modular / "decompositional"

uses decomposition into smaller histories, by definition

may allow for modular verification