10.2 Minimal Violations and Locality:

Goal: Show that in a minimal violation only a single thread reorders its actions.

Definition (Minimal violation):

Consider $\tau = \alpha . a . \beta . b . \gamma \in C_{TSO}(P)$ with $thread(a) = t = thread(b)$.

- Then the distance of a and b in τ is
$$d_\tau(a, b) := |\beta \downarrow t|.$$
- The number of delays in τ is
$$\#(\tau) := \sum_{\substack{\text{corresponding} \\ isu,\, st \in \tau}} d_\tau(isu, st).$$
- A violating computation τ is minimal if $\#(\tau)$ is minimal among all violating computations.

Note:

Program P is not robust iff it has a minimal violation.

Lemma 1 (Delays in minimal violations are required):

Consider $\tau = \alpha . isu . \beta . st . \gamma \in C_{TSO}(P)$ a minimal violation with isu, st from the same instruction of thread t.

Then $\beta \downarrow t = \varepsilon$

or $\beta \downarrow t = \beta' . ld . \beta''$ with $addr(ld) \neq addr(st)$ and β'' contains only stores.

Proof: Suppose β contains one or more actions of thread t.

- If all actions of t in β are stores, then also $\tau' := \alpha . \beta . isu . st . \gamma \in C_{TSO}(P)$.

The computation has the same trace as τ,
but

$$\#(\tau') < \#(\tau). \quad \lightning \text{ Minimality.}$$

. Let a be the last non-store action in $\beta \downarrow t$:

$$\beta = \beta_1 . a . \beta_2 .$$

This means all actions of t in β_2 are stores.
The remaining actions belong to other threads.

. Since store actions cannot be delayed past a fence,
a is (1) issue, (2) local action, (3) or load.

In case (1), (2), and (3) with early load ($addr(a) = addr(st)$),
delaying st past a can be avoided:

$$\tau' := \alpha . isu . \beta_1 . \beta_2 . st . a . \gamma \in C_{TSO}(P).$$

Again $Tr(\tau') = Tr(\tau)$ and $\#(\tau') < \#(\tau)$. $\lightning$

$\square$

<u>Goal</u>: Detect happens-before cycles in a trace (graph structure)
on the computation (linear structure).

<u>Definition (Happens before through)</u>:
Let $\tau = \alpha . a . \beta . b . \gamma \in C_{TSO}(P)$.
Then <u>a happens before b through β</u>
if there is a subsequence $c_1, \dots, c_n$ of β
with $c_i \rightarrow_{hb} c_{i+1}$ or $c_i \rightarrow_{po}^{*} c_{i+1}$ for $0 \leq i \leq n$
with $c_0 := a$
and $c_{n+1} := b$.

Lemma 2 (Happens before through is stable under insertion):

Consider $\tau = \alpha . a . \beta b . \gamma$

and $\tau' = \alpha' a \beta' b \gamma'$ $\in C_{TSO}(P)$

so that $\tau \Downarrow t = \tau' \Downarrow t$ for all $t \in TID$.

Moreover, assume β is a subsequence of β'.

If $a \rightarrow_{hb}^{+} b$ through β, then $a \rightarrow_{hb}^{+} b$ through β'.

Proposition (Dichotomy):

Consider a minimal violation $\tau = \alpha . a \, \beta . b . \gamma \in C_{TSO}(P)$.

Then

(1) $a \rightarrow_{hb}^{+} b$ through β

or (2) $\exists \tau' = \alpha . \beta_1 . b \, a \, \beta_2 \, \gamma \in C_{TSO}(P)$

so that $Tr(\tau') = Tr(\tau)$

and $\tau' \Downarrow t = \tau \Downarrow t$ for all $t \in TID$.

Proof:

Showing $(1) \vee (2)$ is equivalent to $\neg(1) \Rightarrow (2)$,

which is what we prove.

We proceed by induction on the length of β

and strengthen the hypothesis:

We additionally show that β_2 is a subsequence of β.

IA: Then $\tau = \alpha a b \gamma$ and $a \not\rightarrow_{hb} b$.

$|\beta| = 0$

- If $thread(a) = thread(b)$, then $b \rightarrow_{po}^{+} a$.

Therefore, b is a store action which has been delayed past a.

Swapping a and b will save the delay

without changing the trace. $\lightning$ minimality.

- If $thread(a) \neq thread(b)$, then either
  - ↳ one of the actions is local,
  - ↳ the actions access different addresses, or
  - ↳ both are loads.

In all three cases, swapping the actions produces τ' as required.

<u>IS</u>: Assume the statement holds for all β' with $|\beta'| < n$.
Consider $\tau = \alpha . a . \beta . c . b . \gamma$ with $|\beta . c| = n+1$.

Since we assume $a \not\to^+_{hb} b$ through $\beta . c$,
we have $a \not\to^+_{hb} c$ through β or $c \not\to_{hb} b$.

<u>Let $a \not\to^+_{hb} c$ through β:</u>
We apply the induction hypothesis to a and c.
This gives $\tau' = \alpha . \beta_1 . c . a . \beta_2 . b . \gamma$
with $Tr(\tau') = Tr(\tau)$ and β_2 a subsequence of β.
and $\tau' \downarrow t = \tau \downarrow t$ for all $t \in TID$

If we had $a \to^+_{hb} b$ through β_2 in τ',
then also $a \to^+_{hb} b$ through $\beta . c$ in τ.
This holds by the induction hypothesis
(that β_2 is a subsequence of β)
together with Lemma 2,
and contradicts the assumption $a \not\to^+_{hb} b$ through $\beta . c$.
Hence, we can apply the hypothesis to a and b in τ'.
This yields
$\tau'' = \alpha . \beta_1 . c . \beta_{21}\, b\, a\, \beta_{22}\, \gamma$.

Again
$Tr(\tau'') = Tr(\tau')$ and β_{22} a subsequence of β_2.
and $\tau'' \downarrow t = \tau' \downarrow t$ for all $t \in TID$

Together:
- $Tr(\tau'') = Tr(\tau)$
- $\tau'' \downarrow t = \tau \downarrow t$ for all $t \in TID$
- β_{22} is a subsequence of β_2,
  which is a subsequence of β,
  which is a subsequence of $\beta.c$,

so β_{22} is a subsequence of $\beta.c$.

Let $c \to_{hb} b$:

We apply the induction hypothesis to b and c.
This yields
$$\tau' = \alpha\, a\, \beta . b . c\, \gamma$$
with $Tr(\tau') = Tr(\tau)$ and $\tau' \downarrow t = \tau \downarrow t$ for all $t \in TID$.

We now apply the hypothesis to a and b in τ'
and get
$$\tau'' = \alpha . \beta_1 . b\, a . \beta_2 . c\, \gamma$$
with $Tr(\tau'') = Tr(\tau')$, $\tau'' \downarrow t = \tau' \downarrow t$ for all $t \in TID$,
and β_2 a subsequence of β.

Together, $Tr(\tau'') = Tr(\tau)$, $\tau'' \downarrow t = \tau \downarrow t$ for all $t \in TID$,
and $\beta_2 . c$ is a subsequence of $\beta . c$.

$\square$