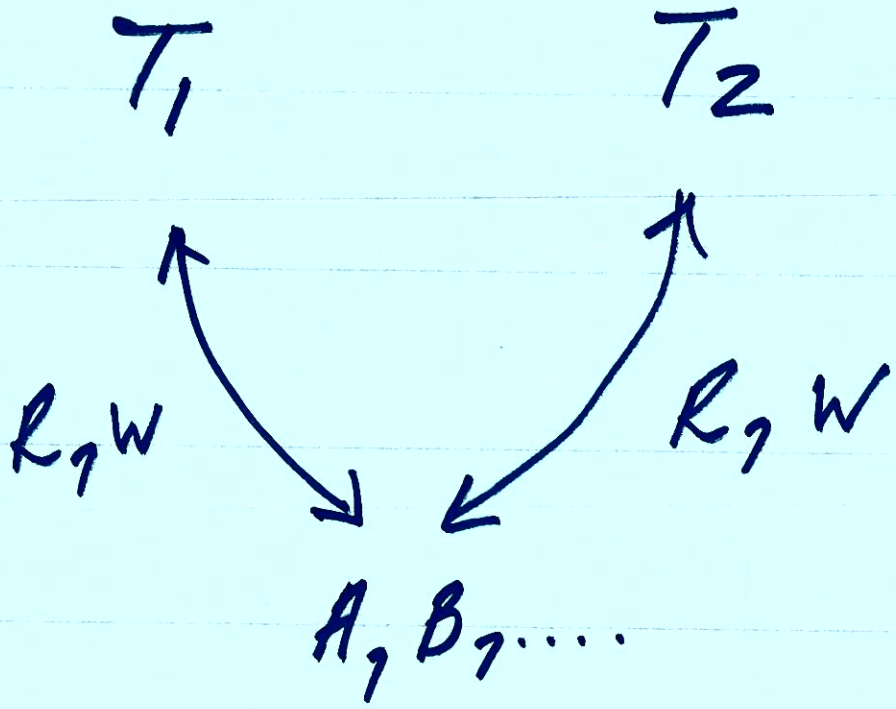


Safe interleaving of accesses by T_1 and T_2 to A, B, ...

When we swap read/write/commit operations in Schedules

– we will never permit swaps that change the order of operations within an individual transaction in the Schedule

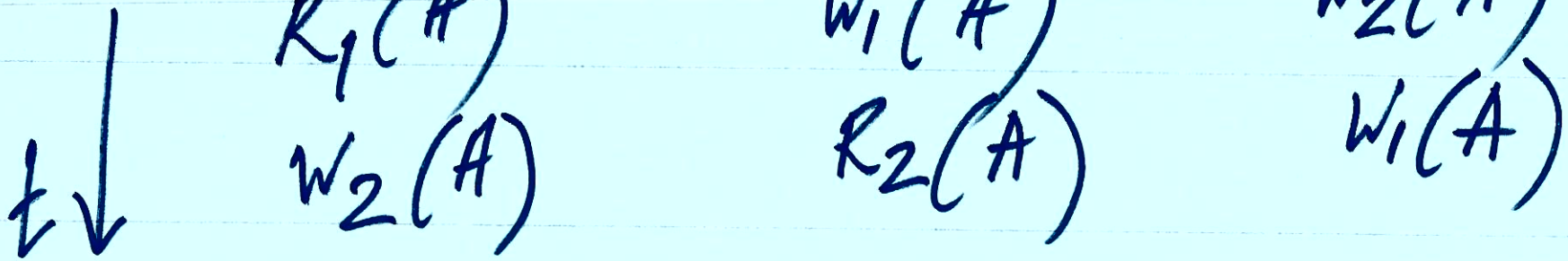
Main idea:

- try to swap R/W of xactions in schedule

= so as to arrive at a sequential execution (= no overlaps)

⇒ we cannot do swaps that change the effects of the R/W on any individual data item

Conflicting pairs:



Schedule S of slide 18:

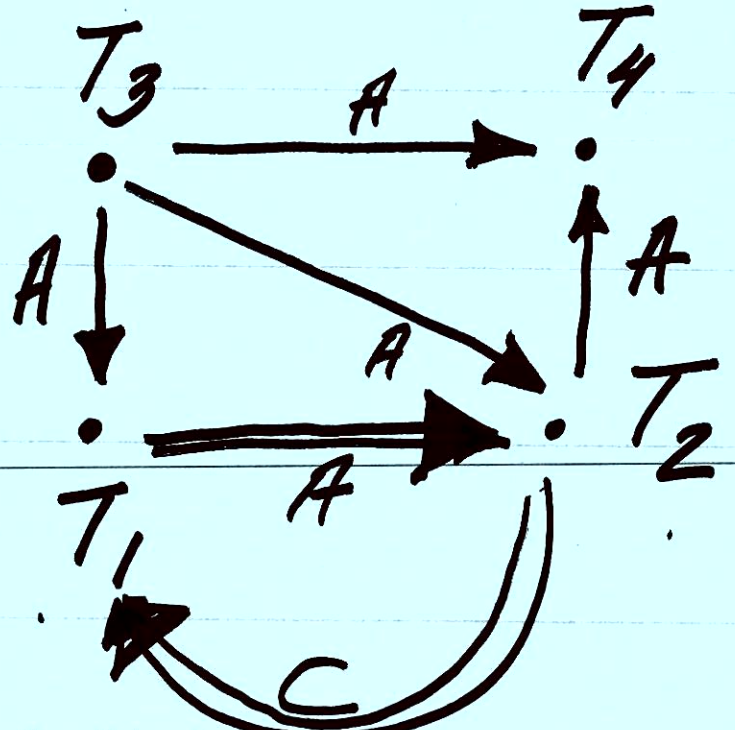
$S = W_3(A)\ W_2(C)\ R_1(A)\ W_1(B)\ R_1(C)\ W_2(A)\ R_4(A)\ W_4(D)$

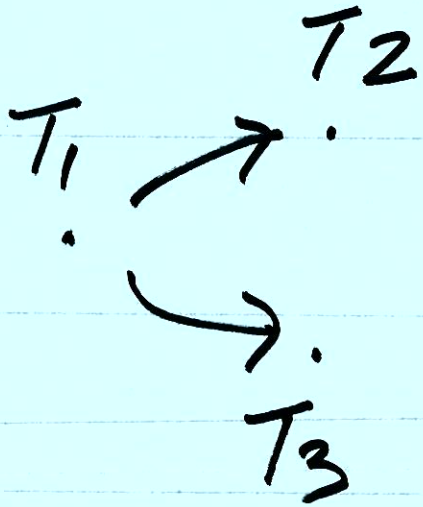
✓ A: $W_3(A)$ $R_1(A)$ $W_2(A)$ $R_4(A)$

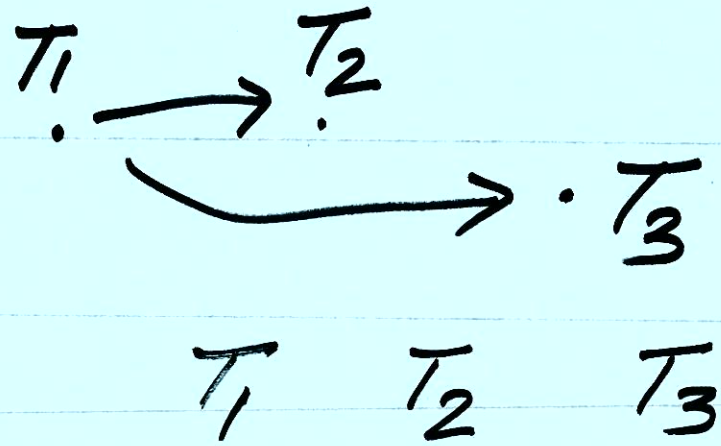
✓ B: $W_1(B)$

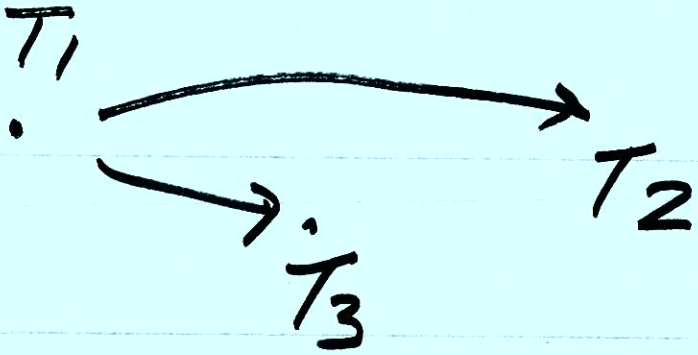
✓ C: $W_2(C)$ $R_1(C)$

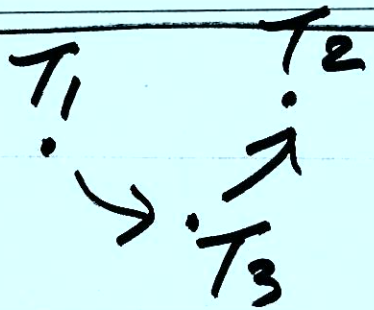
✓ D: $W_4(D)$

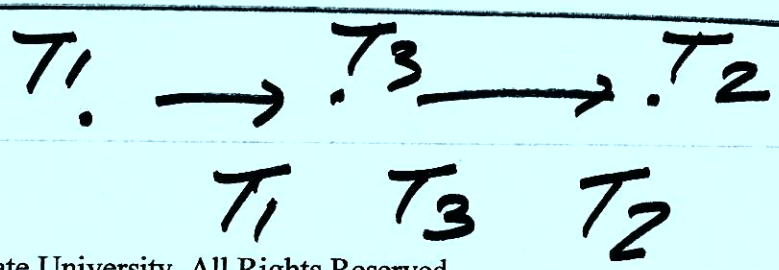












commit for T_2

commit for T_1

S: $w_1(A), R_2(A), c_2, c_1$

T_2 reads from T_1

– if T_1 does not commit

$\Rightarrow S$ is not recoverable

T_1 reads from T_2 in sched S

(intuition: S: ..... $W_2(A) R_1(A)$)

if, for some data item A:

$R_1(A)$ is the first read operation on A after the latest write operation on A by T_2

S: .... $W_3(B) R_1(C) W_2(B) R_1(B)$

S: $R_3(C)$ $W_4(C)$ $W_5(C)$ $W_3(C)$

$R_5(C)$ $W_5(C)$ $R_4(C)$

$R_5(C)$

T_5 reads from T_3 (on C);

T_4 reads from T_5

* Recoverable schedule:
  - each T commits only after each xaction from which it had read has committed

yes ✓ S_1: $W_1(A), C_1, R_2(A), C_2$ (& ACR)

yes ✓ S_2: $W_1(A), R_2(A), C_1, C_2$ (not ACR)

yes ✓ S_3: $W_1(A), R_3(A), C_1, R_2(A), C_3, C_2$

no - S_4: $W_1(A), R_3(A), C_3, C_1, R_2(A), C_2$

(T_1 - T_3)

- Schedules that avoid cascading rollbacks (ACR):
  - each T reads from a committed transaction

s: $R_1(A), R_2(A), R_3(B), W_1(A), R_2(C), R_2(B), W_2(B), W_1(C)$

1. Maximize concurrency:

Assume each xaction
- takes a lock on each data item immediately before it first uses it, and
- releases the lock immediately after the last time it accesses the data item

$S(T_1)$: $R_1(A)$, $W_1(A)$, $W_1(C)$

$S(T_2)$: $R_2(A)$, $R_2(C)$, $R_2(B)$, $W_2(B)$

$S(T_3)$: $R_3(B)$

⇓

$S^{lu}(T_3)$: $l_3(B)$, $R_3(B)$, $u_3(B)$

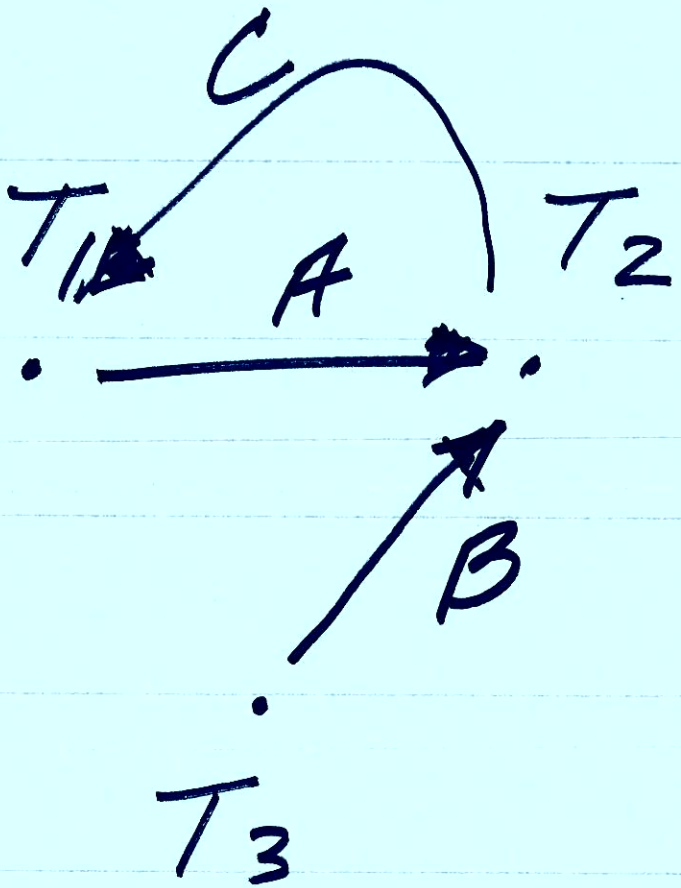
$S^{lu}(T_1)$: $l_1(A)$, $R_1(A)$, $W_1(A)$, $u_1(A)$, $l_1(C)$, $W_1(C)$, $u_1(C)$

$S^{lu}(T_2)$: $l_2(A)$, $R_2(A)$, $u_2(A)$, $l_2(C)$, $R_2(C)$, $u_2(C)$, $l_2(B)$, $R_2(B)$, $W_2(B)$, $u_2(B)$

Resulting schedule:

S': $l_1(A)$, $R_1(A)$, $W_1(A)$, $u_1(A)$,
$l_2(A)$, $R_2(A)$, $l_3(B)$, $R_3(B)$, $u_3(B)$,
$u_2(A)$,
$l_2(C)$, $R_2(C)$, $u_2(C)$, $l_2(B)$,
$R_2(B)$, $W_2(B)$, $u_2(B)$, $l_1(C)$,
$W_1(C)$, $u_1(C)$

A: R_1, W_1, R_2
B: R_3, R_2, W_2
C: R_2, W_1



2. 2PL - two-phase locking on ✓

$S(T_1)$: $R_1(A)$, [redacted] $W_1(A)$, $W_1(C)$

$S(T_2)$: $R_2(A)$, $R_2(C)$, $R_2(B)$, $W_2(B)$

$S(T_3)$: $R_3(B)$

$S^{2PL}(T_1)$: $L_1(A)$, $R_1(A)$, $W_1(A)$, $L_1(C)$, $U_1(A)$, $W_1(C)$, $U_1(C)$

$S^{2PL}(T_2)$: $L_2(A)$, $R_2(A)$, $L_2(C)$, $R_2(C)$, $L_2(B)$, [redacted] $U_2(C)$, $U_2(A)$, $R_2(B)$, $W_2(B)$, $U_2(B)$

$\int^{2PL}(T_3)$: $l_3(B), R_3(B), u_3(B)$

S^{2PL}: $l_1(A), R_1(A), l_3(B), R_3(B),$
$u_3(B), W_1(A), l_1(C), u_1(A),$
$l_2(A), R_2(A), W_1(C), u_1(C),$
$l_2(C), R_2(C), l_2(B), u_2(C),$
$u_2(A), R_2(B), W_2(B), u_2(B)$

A: R_1 W_1 R_2

B: R_3 R_2 W_2

C: W_1 R_2