

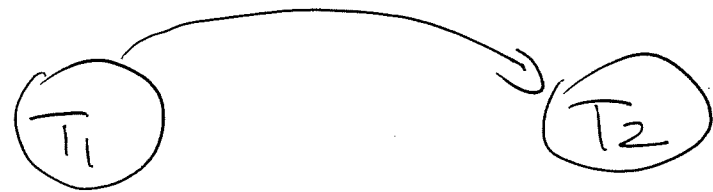
S1: $r_1(x)$ $r_2(y)$ $r_1(w)$ $w_1(x)$ $w_1(w)$ $w_2(y)$ $r_2(w)$ $r_2(x)$ $w_2(x)$ $w_2(w)$

S2: $r_1(x)$ $r_1(w)$ $w_1(x)$ $w_1(w)$ $r_2(y)$ $r_2(w)$ $r_2(x)$ $w_2(x)$ $w_2(w)$

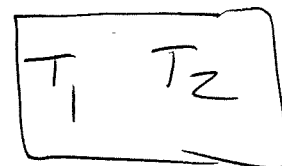
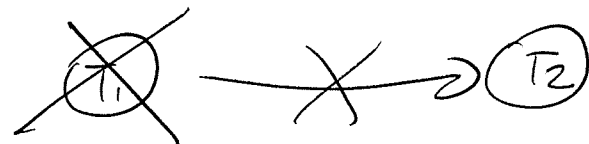
List all conflicts in S1

Conflict graph

$w_1(w)$ $r_2(w)$
 $w_1(x)$ $r_2(x)$
 $w_1(x)$ $w_2(x)$
 $r_1(x)$ $w_2(x)$
 $r_1(w)$ $w_2(w)$
 $w_1(w)$ $w_2(w)$



T_1 T_2



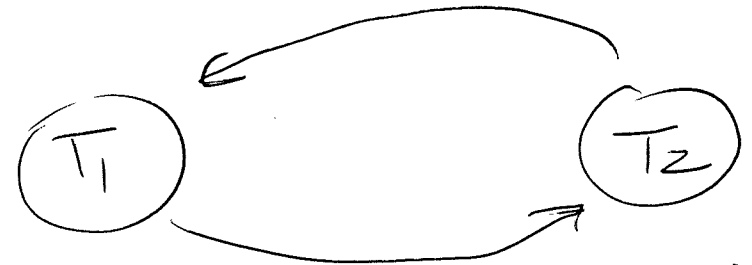
$r_1(x)$ $r_2(x)$ $w_1(x)$ $w_2(x)$

Conflicts

$r_2(x)$ $w_1(x)$

$w_1(x)$ $w_2(x)$

~~$w_1(x)$~~ $w_2(x)$



This schedule is not serializable.

S: $r_1(x)$ $r_3(y)$ $r_2(w)$ $r_3(z)$ ~~$w_3(y)$~~ $r_4(y)$ $r_2(y)$

$r_1(x)$ ~~$w_1(x)$~~ $r_4(x)$ $w_2(w)$ $r_4(w)$ $w_4(x)$

List all conflicts

$w_3(y)$ $r_4(y)$

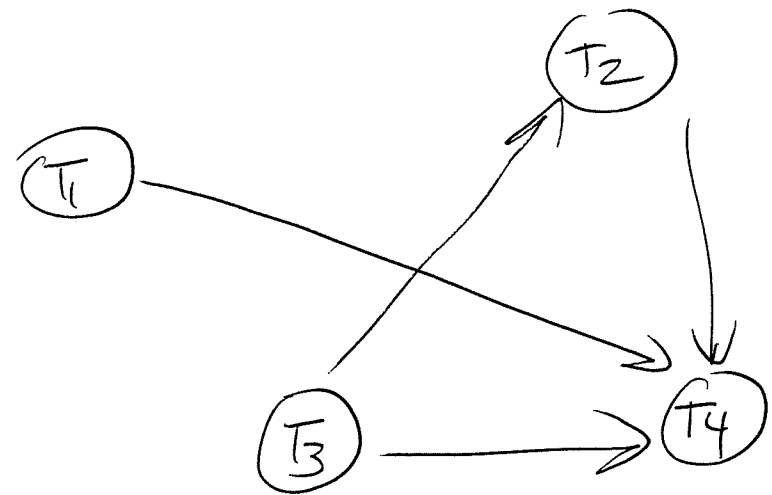
$w_3(y)$ $r_2(y)$

$w_1(x)$ $r_4(x)$

$w_2(w)$ $r_4(w)$

$w_1(x)$ $w_4(x)$

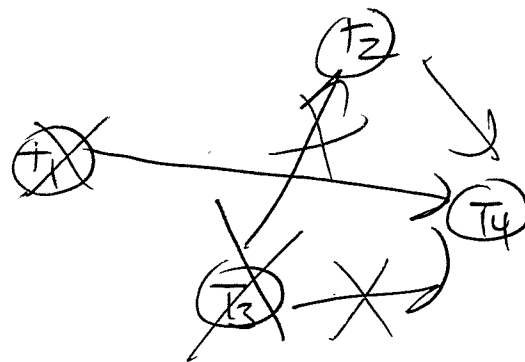
$r_1(x)$ $w_4(x)$



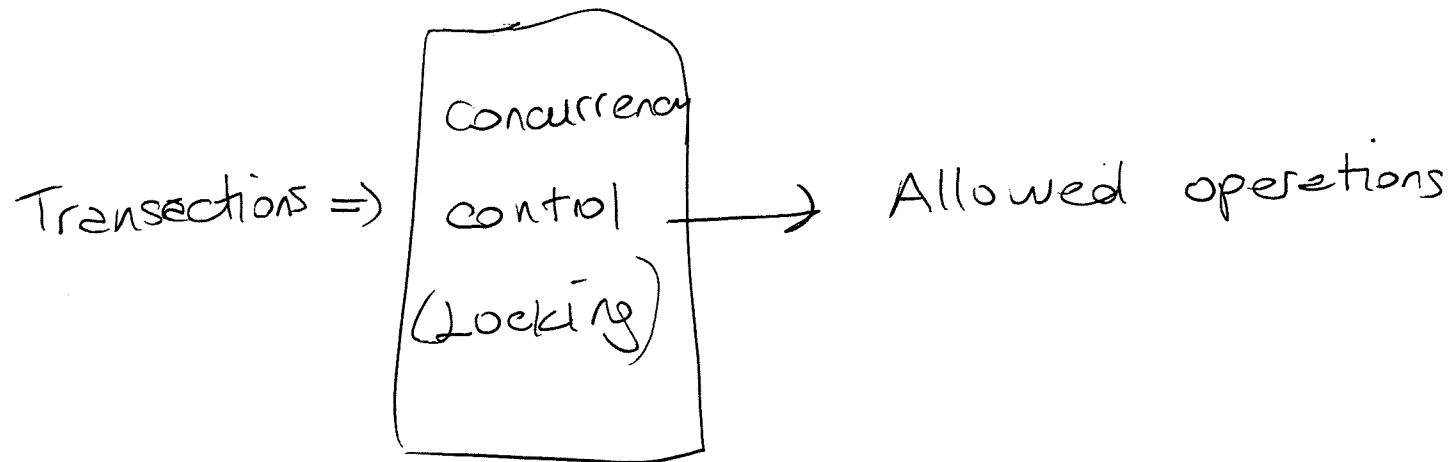
No cycle, serializable

$\begin{Bmatrix} T_3 & T_2 & T_1 & T_4 \\ T_1 & T_3 & T_2 & T_4 \\ T_3 & T_1 & T_2 & T_4 \end{Bmatrix}$

both of these are conflict equivalent to S.



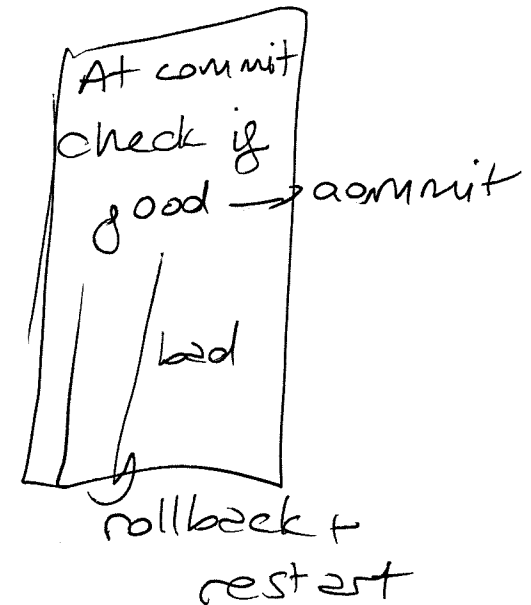
Option 1:



Option 2



Multiversion concurrency
control



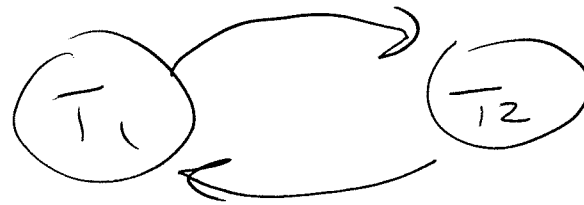
Existing lock on ~~S~~ item A

Requested new lock
item A

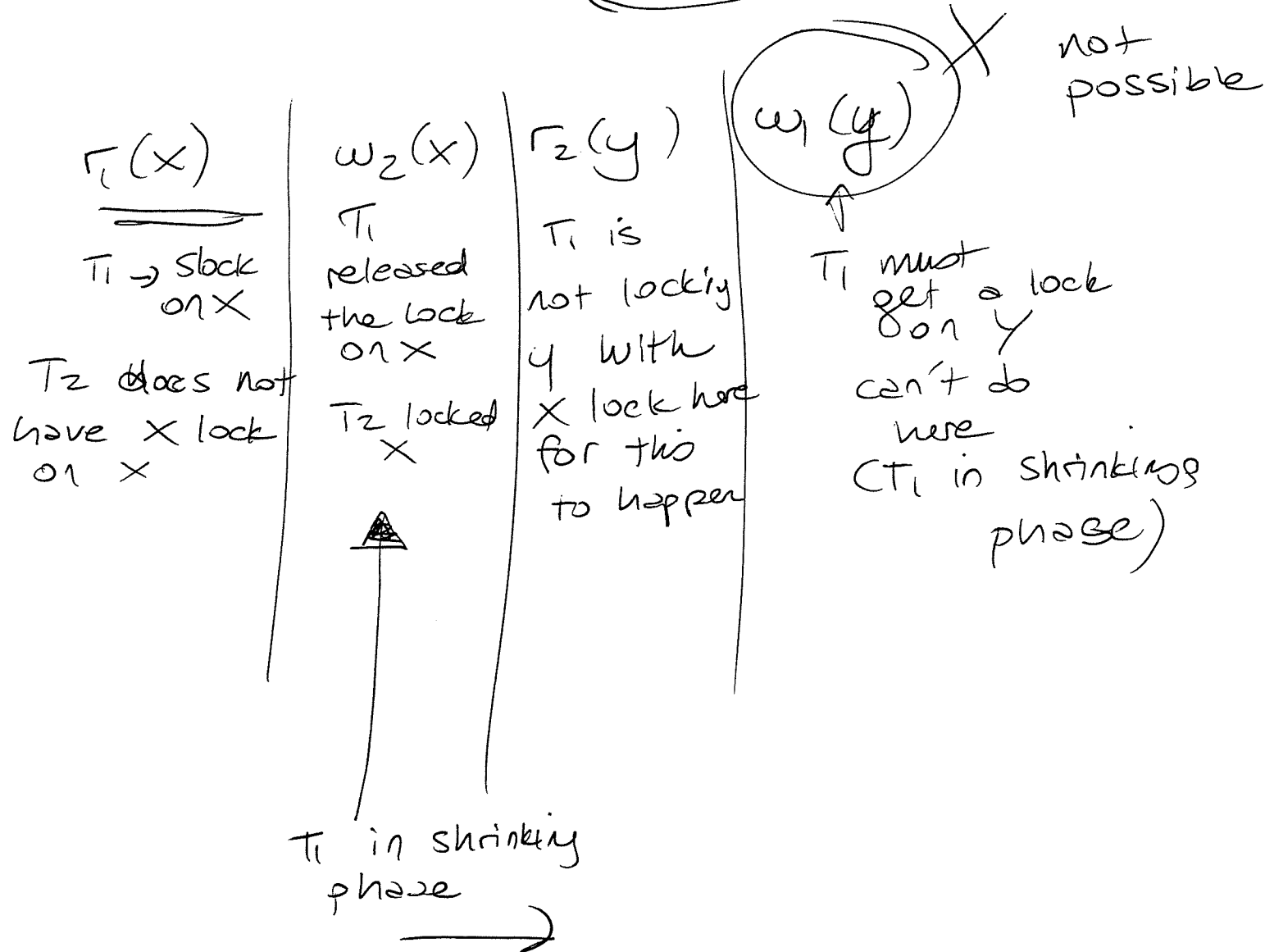
	Nothing	S	X
S	✓	✓	NO NO
X	✓	NO	NO

2PL guarantees serializability!

Assume not:



suppose this was possible!



T_1

$slock(x)$

$r_1(x)$

T_2

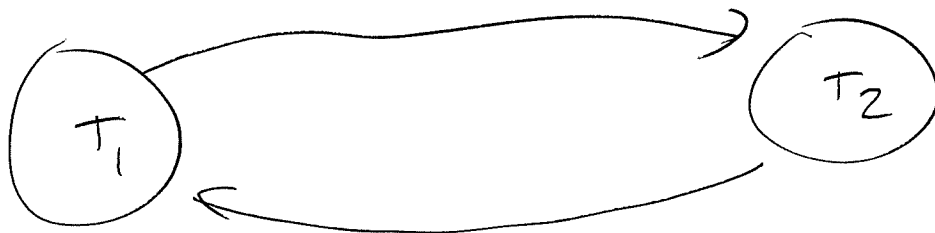
$slock(y)$

$r_2(y)$

$request_xlock_1(y)$
<WAIT MODE>

$request_xlock_2(x)$
<WAIT MODE>

DEADLOCK



WAITING FOR