CS3DB3/SE4DB3/ SE6DB3 TUTORIAL
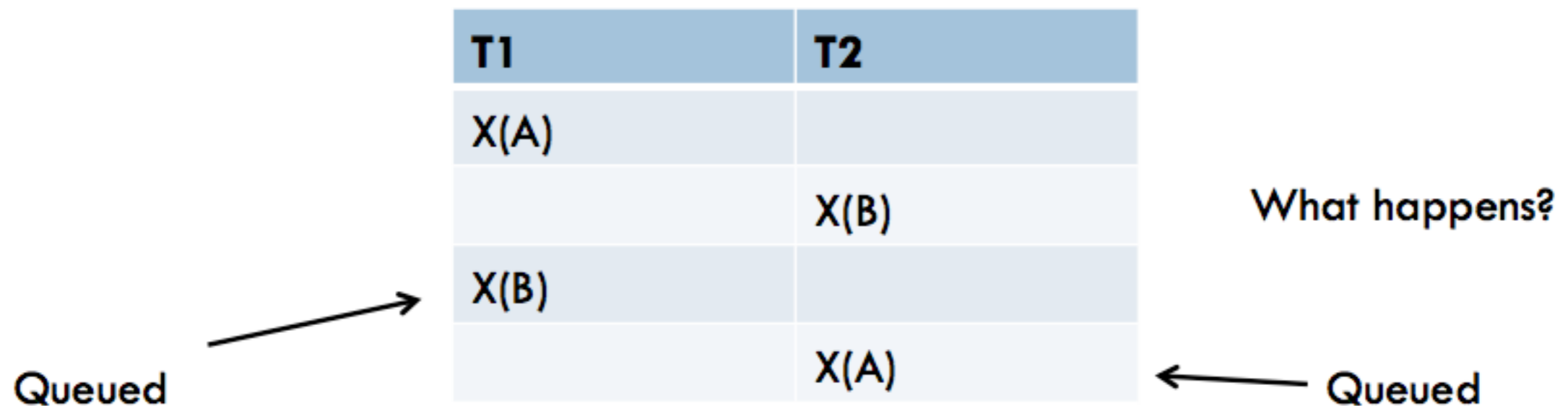
YU HUANG
2015-04-01

Outline

- Deadlock

- Deadlock Detection

- Wait-Die

- Wound-Wait

- Intention locks

Deadlock

- Deadlock: Cycle of transactions waiting for locks to be released by each other.

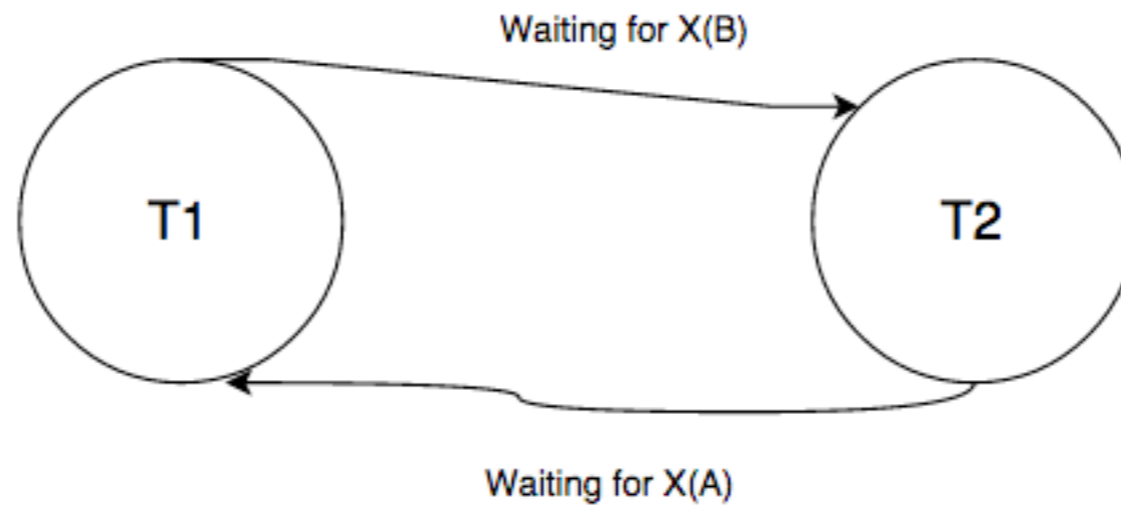| T1	T2
X(A)	
	X(B)
X(B)	
	X(A)

Queued →
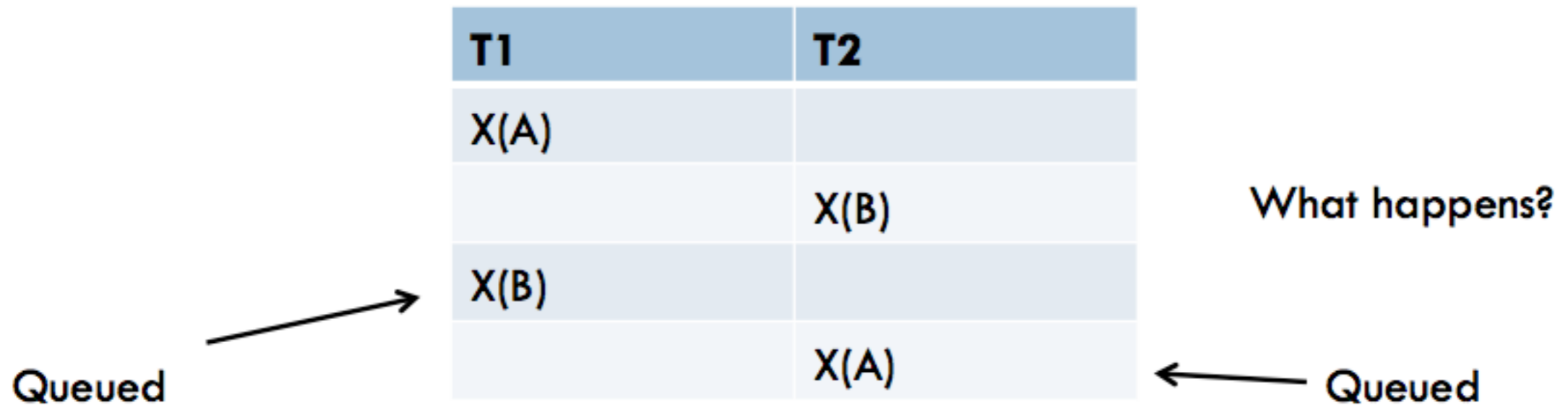
← What happens?

← Queued

T1 is waiting for T2 to release its lock
T2 is waiting for T1 to release its lock

Deadlock Detection

Waits-for graph

- Nodes are transactions

- T1 points to T2 means T1 is waiting for T2 to release a lock

- a cycle means that deadlock exists

- example

T1	T2
X(A)	
	X(B)
X(B)	
	X(A)

What happens?

Queued

Queued

Waiting for X(B)

T1   T2

Waiting for X(A)
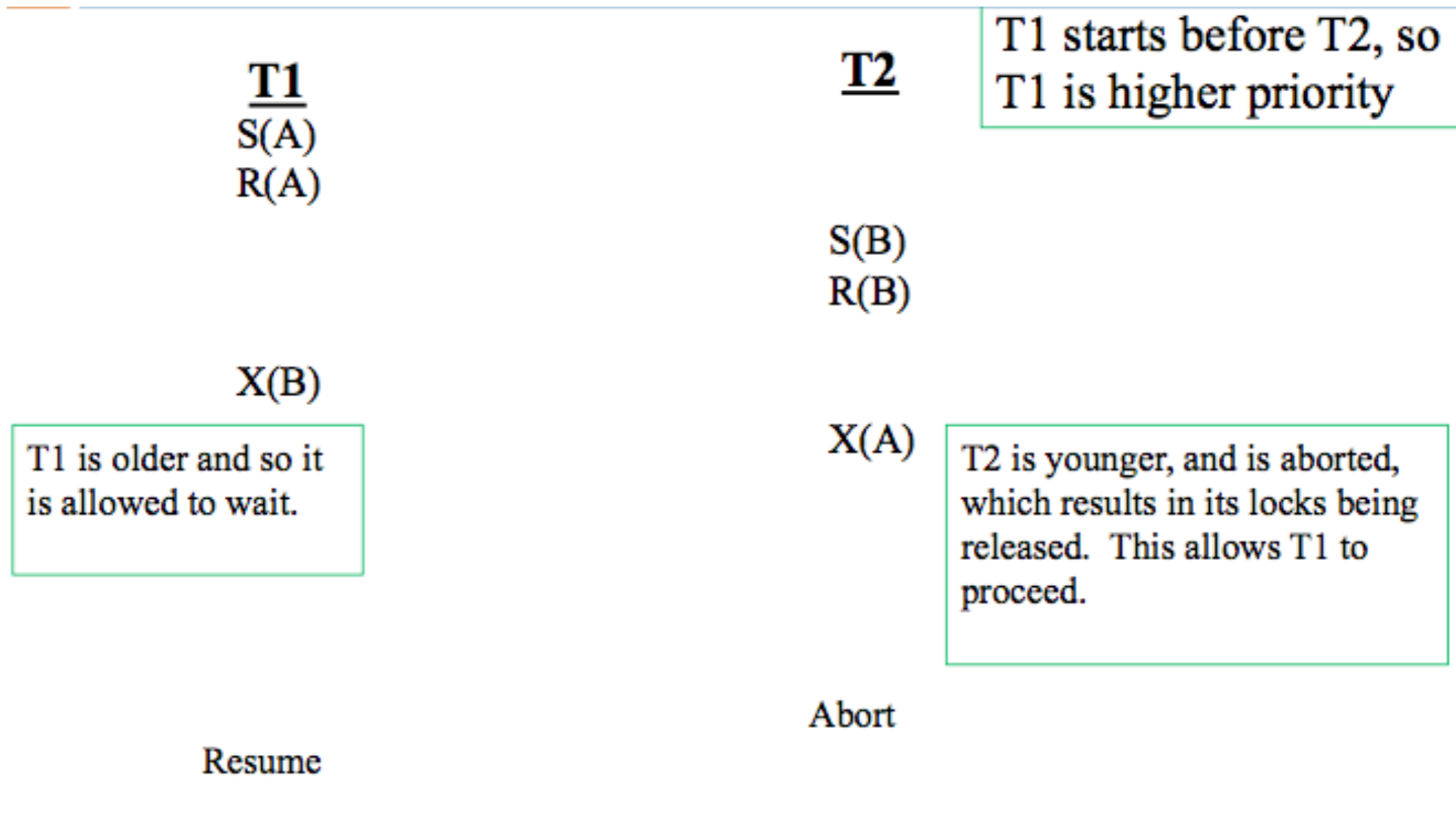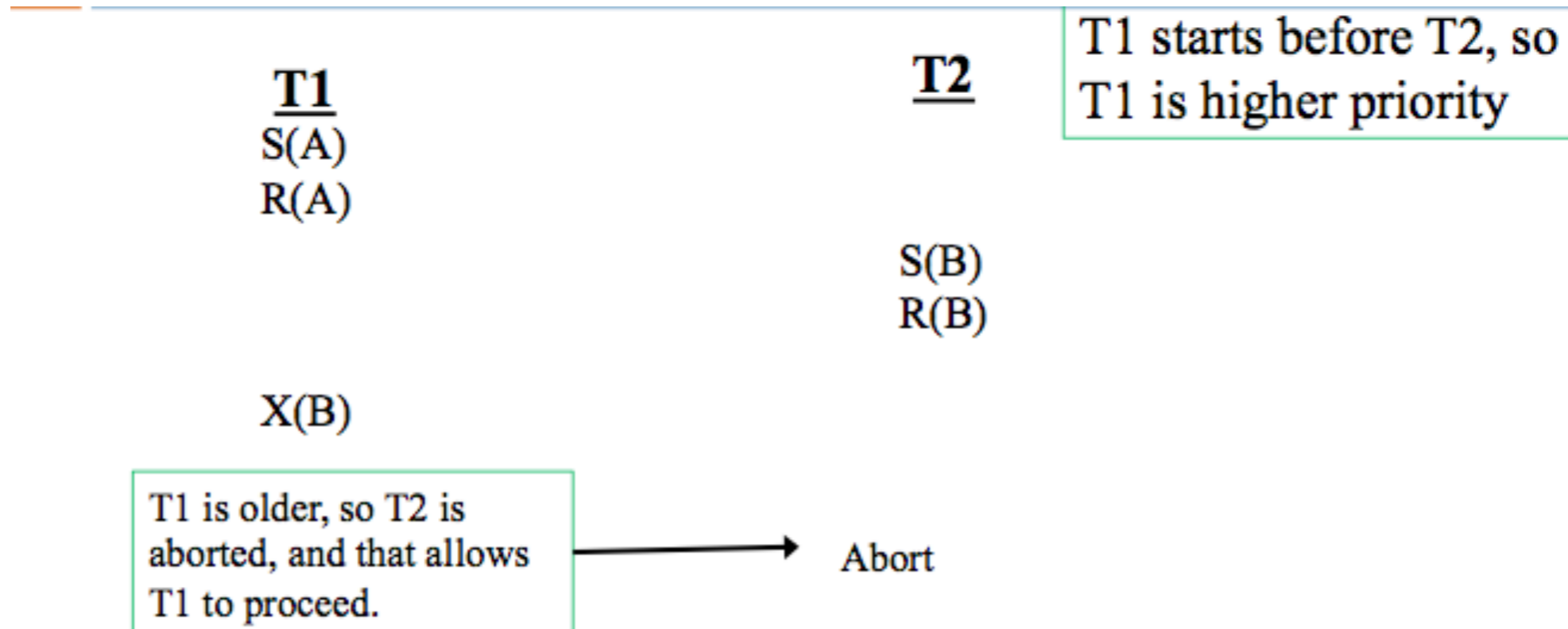
Deadlock Prevention

- Wait-Die: If Ti has higher priority, Ti waits for Tj; otherwise Ti aborts.

- Wound-Wait: If Ti has higher priority, Tj aborts; otherwise Ti waits.

- Assign priorities based on timestamps: oldest transaction has the highest priority.

Example: wait-die

T1	**T2**	T1 starts before T2, so T1 is higher priority
S(A)		
R(A)		
	S(B)	
	R(B)	
X(B)		
	X(A)	

T1 is older and so it is allowed to wait.

T2 is younger, and is aborted, which results in its locks being released. This allows T1 to proceed.

Abort

Resume

Example:wound-wait

T1

S(A)
R(A)

T2

| T1 starts before T2, so |
| T1 is higher priority |

S(B)
R(B)

X(B)

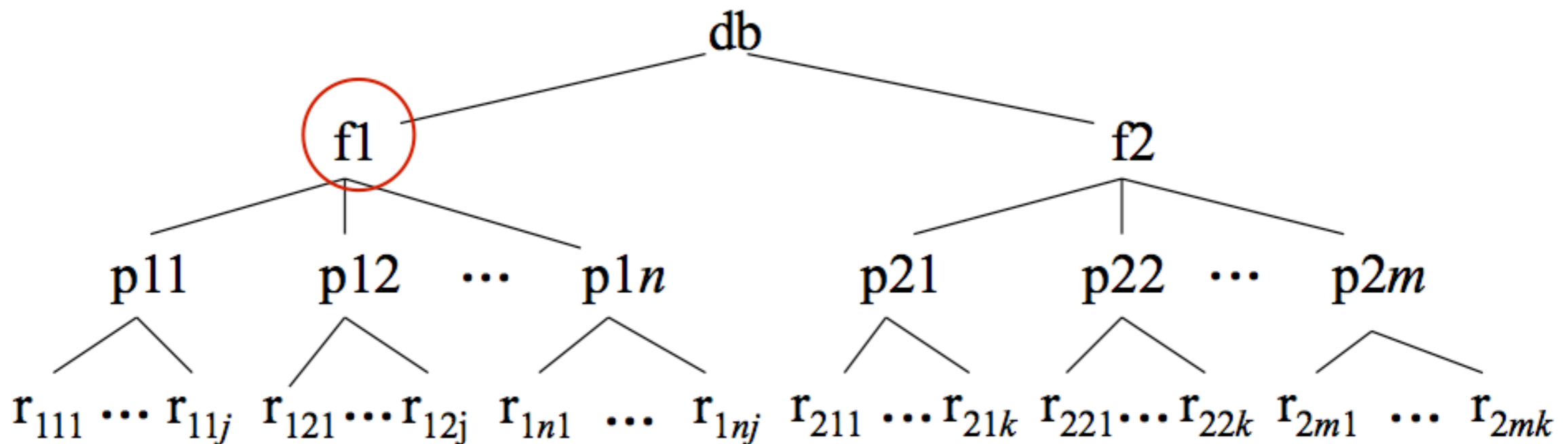| T1 is older, so T2 is |
| aborted, and that allows |
| T1 to proceed. | → Abort

Intention Locks

T1: updates all the records in file f1.
T2: read record r_{1nj}.

Assume T1 comes before T2;
Assume T2 comes before T1

To improve the efficiency of the Database Engine in detecting lock conflicts at the higher level of granularity.