

# Transaction Chopping

# Example: Simple Purchases

- Chopping solution:
  1. If  $\text{Cash} < P$  then roll back.  
     $\text{Cash} := \text{Cash} - P$ .
  2.  $\text{Inventory}(I) := \text{inventory}(I) + P$
- Chopping execution:
  - P11:  $100 > 50$ .  $\text{Cash} := 50$ .
  - P21:  $75 > 50$ . Rollback.
  - P12:  $\text{inventory} := \text{inventory} + 50$ .

# Transaction Chopping

- Execution rules:
  - When pieces execute, they follow the partial order defined by the transactions.
  - If a piece is aborted because of a conflict, it will be resubmitted until it commits
  - If a piece is aborted because of an abort, no other pieces for that transaction will execute.

# Transaction Chopping

- Let  $T_1, T_2, \dots, T_n$  be a set of transactions. A chopping partitions each  $T_i$  into pieces  $c_{i1}, c_{i2}, \dots, c_{ik}$ .
- A chopping of  $T$  is rollback-safe if (a)  $T$  does not contain any abort commands or (b) if the abort commands are in the first piece.

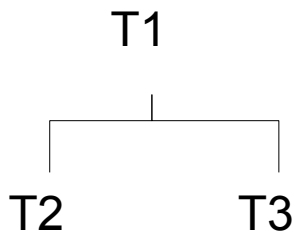
# Correct Chopping

- Chopping graph (variation of the serialization graph):
  - Nodes are pieces
  - Edges:
    - C-edges: C stands for conflict. There is a C-edge between two pieces from different transactions if they contain operations that access the same data item and one operation is a write.
    - S-edges: S stands for siblings. There is an S-edge between two pieces, iff they come from the same transaction.
- A chopping graph contains an S-C cycle if it contains a cycle that includes at least one S-edge and one C-edge.

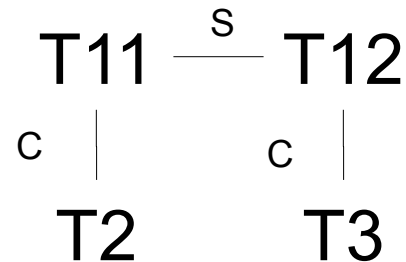
# Correct Chopping

- A chopping is correct if it is rollback safe and its chopping graph contains no SC-cycle.

T1: r(x) w(x) r(y) w(y)  
 T2: r(x) w(x)  
 T3: r(y) w(y)

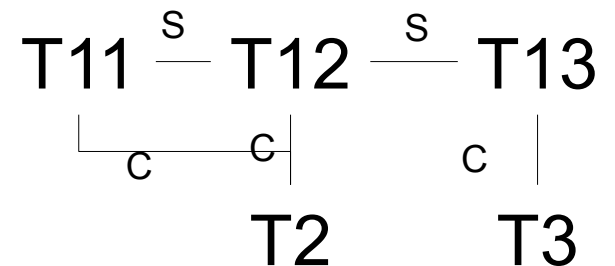


T11: r(x) w(x)  
 T12: r(y) w(y)



CORRECT

T11: r(x)  
 T12: w(x)  
 T13: r(y) w(y)



NOT CORRECT

# Chopping Example

T1: RW(A) RW (B)

T2: RW(D) RW(B)

T3: RW(E) RW(C)

T4: R(F)

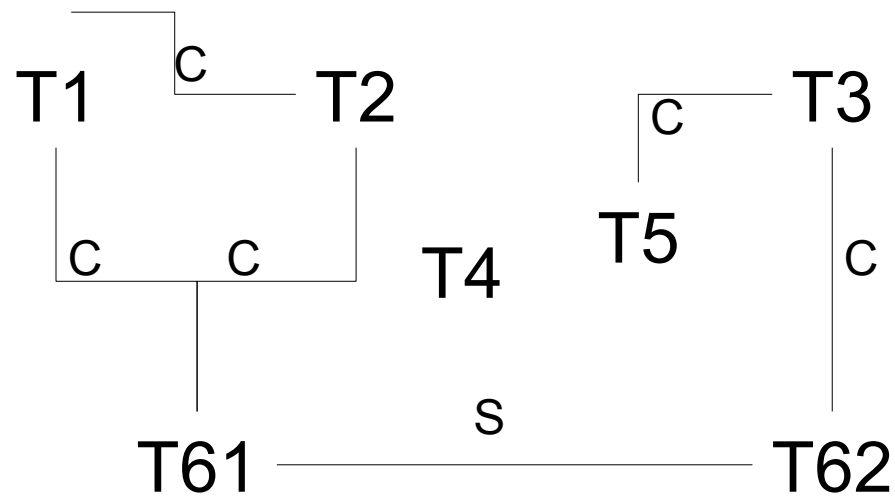
T5: R(E)

T6: R(A) R(F) R(D) R(B) R(E) R(G) R(C)

# Chopping Example

T61: R(A) R(F) R(D) R(B)

T62: R(E) R(G) R(C)



# Finest Chopping

- A private chopping of transaction  $T_i$ , denoted  $\text{private}(T_i)$  is a set of pieces  $\{c_{i1}, c_{i2}, \dots, c_{ik}\}$  such that:
  - $\{c_{i1}, c_{i2}, \dots, c_{ik}\}$  is a rollback safe chopping
  - There is no SC-cycle in the graph whose nodes are  $\{T_1, \dots, T_{i-1}, c_{i1}, c_{i2}, \dots, c_{ik}, T_{i+1}, \dots, T_n\}$
- The chopping consisting of  $\{\text{private}(T_1), \text{private}(T_2), \dots, \text{private}(T_n)\}$  is rollback-safe and has no SC-cycles.

# Finest Chopping

- In:  $T, \{T_1, \dots, T_{n-1}\}$
- Initialization
  - If there are abort commands
    - then  $p_1 :=$  all writes of  $T$  (and all non swappable reads) that may occur before or concurrently with any abort command in  $T$
    - else  $p_1 :=$  first database access
  - $P := \{x \mid x \text{ is a database operation not in } p_1\}$
  - $P := P \setminus \{p_1\}$

# Finest Chopping

- Merging pieces
  - Construct the connected components of the graph induced by  $C$  edges alone on all transactions  $\{T_1, \dots, T_{n-1}\}$  and on the pieces in  $P$ .
  - Update  $P$  based on the following rule:
    - If  $p_j$  and  $p_k$  are in the same connected component and  $j < k$ , then
      - add the accesses from  $p_k$  to  $p_j$
      - delete  $p_k$  from  $P$