CS 448 Database Systems

Serializability Theory

Serializability Theory

- We will develop a theory that determines what types of interleaving of transactions are acceptable (i.e. serializable).
- Introduce a new model with
  - transactions
  - histories

Transactions

- A transaction consists of read and write operations on database objects.
- It also specifies an order in which the operations are executed. This may be a partial order, i.e., some pairs of operations are not strictly ordered in time. The order describes the *"happened-before"* relationship.

Transactions

- Each operation of a transaction will be represented by the following symbols:
  - $r_1[x]$ – txn T_1 reads data item x
  - $w_1[x]$ – txn T_1 writes data item x
  - c_1 – txn T_1 commits
  - a_1 – txn T_1 aborts
  - The start of a txn is implicit

Transactions

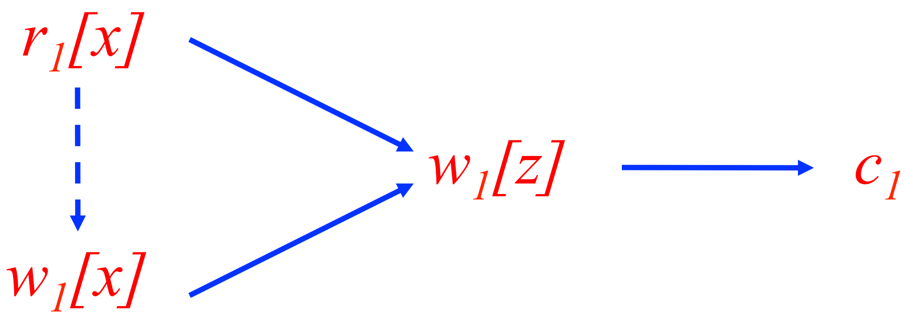
- A txn is represented as a partial order $(\Sigma,<)$, where Σ is the set of elements being ordered (i.e., the operations of the transaction), and $<$ is the ordering relation.
- The final operation in a transaction must be a commit or abort.

Transaction

- A transaction T_i is a partial order with ordering relation $<_i$, where
  - $T_i \subseteq \{r_i[x], w_i[x] \mid x$ is a data item$\} \cup \{ai, ci\}$;
  - $a_i \in T_i$ iff $c_i \notin T_i$;
  - If t_i is c_i or a_i (whichever is in T_i), for any other operation $p \in T_i$, $p <_i t_i$;
  - If $r_i[x], w_i[x] \in T_i$, then either $r_i[x] <_i w_i[x]$ or $w_i[x] <_i r_i[x]$.

Transactions

- A partial order can be represented by a directed acyclic graph (DAG).
- E.g.



Transactions

- Ignore all other actions of txns
- Can model input values as read statements.

Histories

- A history captures the execution of several transactions.
- Histories are collections of the partial orders of txns and are partial orders too.
- They need to be more that just the sum of the partial orders of their constituent transactions though – they MUST order conflicting operations
- A pair of operations *conflict* if they both operate on the same data item and at least one of them is a write.

Complete Histories

- Let $T = \{T_1, T_2, \ldots, T_n\}$ be a set of transactions. A *complete* history H over T is a partial order with ordering relation $<_H$ where:
  - $H = \bigcup_{i=1}^{n} T_i$;
  - $<_H \supseteq \bigcup_{i=1}^{n} <_i$; and
  - For any two conflicting operation, $p, q \in H$, either $p <_H q$ or $q <_H p$.
- A *history* is simply a prefix of a complete history.

Example

- $T_1 = r_1[x] \rightarrow w_1[x] \rightarrow c_1$
- $T_2 = r_2[x] \rightarrow w_2[y] \rightarrow w_2[x] \rightarrow c_2$
- $T_3 = r_3[y] \rightarrow w_3[x] \rightarrow w_3[y] \rightarrow c_3$

$$
\begin{tikzcd}
r_1[x] \arrow[r] & w_1[x] \arrow[r] \arrow[d] & c_1 & & \\
r_3[y] \arrow[r] & w_3[x] \arrow[r] \arrow[d] & w_3[y] \arrow[r] \arrow[d] & c_3 & \\
 & r_2[x] \arrow[r] & w_2[y] \arrow[r] & w_2[x] \arrow[r] & c_2
\end{tikzcd}
$$

Order implied by transitivity are omitted.

For total orders, we can drop the arrows.

Committed Projection of a History

- The committed projection of a history H, denoted *C(H),* is the history obtained from *H* by deleting all operations that do not belong to committed txns.
- This is important for the definition of serializable histories.

Equivalent Histories

- We want to allow only those histories that are EQUIVALENT to some serial history.
- We define two histories H and H' to be equivalent ($\equiv$) if
  - They are defined over the same set of transactions and have the same operations; and
  - They order conflicting operations of non-aborted transactions in the same way; that is, for any conflicting operations p_i and q_j belonging to transactions T_i and T_j (respectively) where $a_j, a_i \notin H$, if $p_i <_H q_j$ then $p_i <_{H'} q_j$.

Serializable Histories

- Because only the complete execution of txns represents a consistent state, we define a history to be serializable (SR) if its committed projection, *C(H),* is equivalent to some serial history H_s.
- A serialization graph can be used to determine whether a history is serializable.

Serialization Graph (SG)

- The *SG(H)* is a directed graph whose nodes are committed txns in *H*, and whose edges are $T_i \rightarrow T_j$ such that one of T_i's operations precedes and conflicts with one of T_j's operations in *H*.

E.g.

$r_3[x] \rightarrow w_3[x] \rightarrow c_3$

$r_1[x] \rightarrow w_1[x] \rightarrow w_1[y] \rightarrow c_1$

$r_2[x] \rightarrow w_2[y] \rightarrow c_2$

$T_2 \rightarrow T_1 \rightarrow T_3$

NOTE: SG may not be transitive!

Serializability Theorem

A History H is serializable iff $SG(H)$ is acyclic