On the Correctness of Transactional Memory

Rachid Guerraoui    Michał Kapałka

EPFL, Switzerland

PPoPP 2008
What is a TM precisely?
Fundamental Question

What is a TM precisely?
When is a TM correct?

Serializability, linearizability, ... do not answer those!
Important Consequences

- Hard to compare TMs
- Hard to prove a TM correct
- Hard to show inherent limitations of TMs
Our Take

Opacity – a correctness condition for TMs (ensured by many TMs)

≈

ACID / serializability for DBs
Opacity

- Compare TMs
  opacity is implementation-agnostic

- Prove a TM correct
  graph interpretation of opacity

- Show inherent limitations of TMs
  time complexity lower bound
Outline

1. TM semantics: intuition
2. Existing correctness criteria: examples
1-lock semantics

atomic {
    a = x
    c . add(5)
    s . push(a)
}
atomic {
    a = x
    c . add(5)
    s . push(a)
}
atomic {
  a = x
  c . \texttt{add}(5)
  s . \texttt{push}(a)
}

1-lock semantics

transaction

opacity
TM Semantics

- Committed: instantaneous
- Aborted: never visible
- All: observe consistent state
Model of Interaction

![Diagram of interaction between App and TM]

- **App**:
  - \( x . \text{add}(5) \)
  - \( \text{try-commit} \)
  - \( \text{ok/aborted} \)

- **TM**:
  - \( \text{ok} \)

---

Michał Kapalęka (EPFL)

Opacity

PPoPP 2008
Outline

1. TM semantics: intuition
2. Existing correctness criteria: examples
Serializability: [Papadimitriou ’79]

- Committed: instantaneous

Recoverability: [Hadzilacos ‘88]

- If $T$ updates $x$, no tx accesses $x$ until $T$ commits/aborts.
Ser. / Recov. Too Weak

\[ x \rightarrow 0 \]

\[ T_1 \]

read

\[ \begin{array}{cc}
0 & 0 \\
x & y
\end{array} \]
Ser. / Recov. Too Weak

\[ x \rightarrow 0 \]

\[ T_1 \]

\[ x = 1 \]

\[ T_2 \]

write

\[ \begin{array}{c}
1 \\
x
\end{array} \quad \begin{array}{c}
0 \\
y
\end{array} \]
Ser. / Recov. Too Weak

\[ x \rightarrow 0 \]

\[ T_1 \]

\[ x = 1 \quad y = 1 \]

\[ T_2 \]

commit

write

\[ 1 \quad 1 \]

\[ x \quad y \]
Ser. / Recov. Too Weak

\[ x \rightarrow 0 \]

\[ T_1 \]

\[ y \rightarrow 1 \]

\[ T_2 \]

\[ x = 1 \quad y = 1 \]

commit

read

\[ \begin{array}{c}
\quad 1 \\
\quad y \\
\end{array} \]

\[ \begin{array}{c}
\quad 1 \\
\quad x \\
\end{array} \]
Ser. / Recov. Too Weak

\[ x \rightarrow 0 \quad y \rightarrow 1 \]

\[ T_1 \]

\[ T_2 \]

\[ x = 1 \quad y = 1 \]

\[ \text{abort} \]

\[ \text{commit} \]

\[ 1 \quad 1 \]

\[ x \quad y \]
$T_1 \xrightarrow{\text{write}} 0$

$x = 1$
Ser. / Recov. Too Strict

\[ x = 1 \]

\[ T_1 \]

\[ x = 2 \]

\[ T_2 \]

write

\[ X \]

\[ 1 \]

\[ X \]
Other Desirable Properties

- Arbitrary objects
- Multiple versions of each object
- Updates at any time
- Real-time order
- User’s perspective
Outline

1. TM semantics: intuition
2. Existing correctness criteria: examples
Given:

- transactional execution
  (sequence of invocations & responses)

Answer:

- is the execution correct?
Opacity: Step by Step

Sequential operations

Sequential transactions

Completed transactions

All transactions
Sequential Operations

\[ s \cdot \text{push}(3) \quad x = 5 \quad s \cdot \text{pop} \rightarrow 3 \quad x \rightarrow 5 \]

Sequential specification of \( s \) and \( x \)
Sequential Transactions

$s \cdot \text{push}(3) \; x = 5$

commit

$s \cdot \text{pop} \rightarrow 3 \; x \rightarrow 5$

commit (abort)

Every transaction is legal
Concurrent, Complete Transactions

\[ s \cdot \text{push}(3) \ x = 5 \]

\[ s \cdot \text{pop} \rightarrow 3 \]

\[ s \cdot \text{pop} \rightarrow 3 \]

\[ x \rightarrow 5 \]

\[ \text{abort} \]
Concurrent, Complete Transactions

\[ s \cdot \text{push}(3) \ x = 5 \]

\[ s \cdot \text{pop} \rightarrow 3 \]

\[ \text{com.} \]

\[ \text{com.} \]

\[ s \cdot \text{pop} \rightarrow 3 \]

\[ x \rightarrow 5 \]

\[ \text{abort} \]
Concurrent, Complete Transactions

\[ s \cdot \text{push}(3) \ x = 5 \quad \text{com.} \quad s \cdot \text{pop} \rightarrow 3 \quad \text{com.} \]

\[ s \cdot \text{pop} \rightarrow 3 \quad x \rightarrow 5 \quad \text{abort} \]
Concurrent, Complete Transactions

$\text{s. push(3)} \ x = 5$

$\text{com.}$

$\text{s. pop} \rightarrow 3$

$\text{com.}$

$\text{s. pop} \rightarrow 3$

$x \rightarrow 5$

abort
Concurrent, Complete Transactions

\[ s \cdot \text{push}(3) \ x = 5 \]

\[ s \cdot \text{pop} \rightarrow 3 \]

\[ s \cdot \text{pop} \rightarrow 3 \]

\[ x \rightarrow 5 \]

abort
Concurrent, Complete Transactions

\[ s \cdot \text{push}(3) \ x = 5 \]
\[ \text{com.} \]
\[ s \cdot \text{pop} \rightarrow 3 \]
\[ \text{com.} \]

\[ s \cdot \text{pop} \rightarrow 3 \]
\[ x \rightarrow 5 \]
\[ \text{abort} \]
Concurrent, Complete Transactions

\[ s \cdot \text{push}(3) \ x = 5 \]
\[ s \cdot \text{pop} \rightarrow 3 \]
\[ x \rightarrow 5 \]
\[ \text{abort} \]
Live Transactions

\[ s . pop \rightarrow 3 \]
Live Transactions

\[ s \cdot \text{pop} \rightarrow 3 \]

\[ \text{aborted} \]

\[ s \cdot \text{pop} \rightarrow 3 \]
Live Transactions

\[
s \cdot \text{pop} \rightarrow 3
\]

\[
\text{try-commit}
\]

committed or aborted
The Definition

A history $H$ ensures opacity if there exists a sequential history $S$ equivalent to some history in set $\text{Complete}(H)$, such that:

1. $S$ preserves the real-time order of $H$,
2. every transaction $T_i \in S$ is legal in $S$. 

Michał Kapalka (EPFL)
Opacity: Captures TM Semantics

- Committed: instantaneous
- Aborted: never visible
- All: observe consistent state
Opacity: Benefits

- Compare TMs
  - opacity is implementation-agnostic

- Prove a TM correct
  - graph interpretation of opacity

- Show inherent limitations of TMs
  - time complexity lower bound