

Disjoint-Access Parallelism: Impossibility, Possibility, and Cost of Transactional Memory Implementations

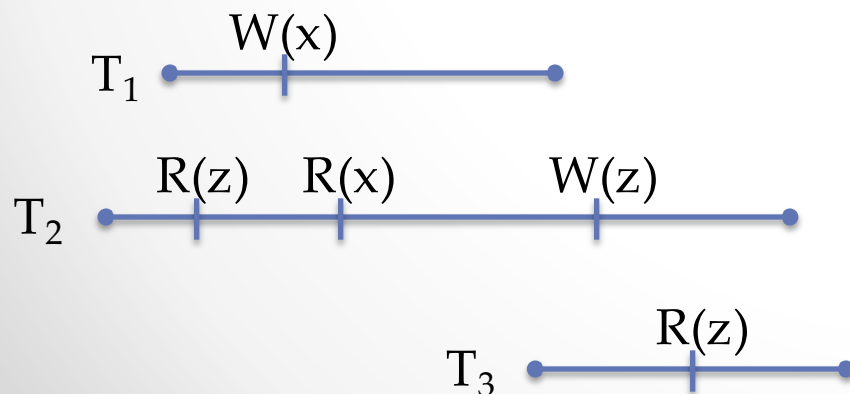
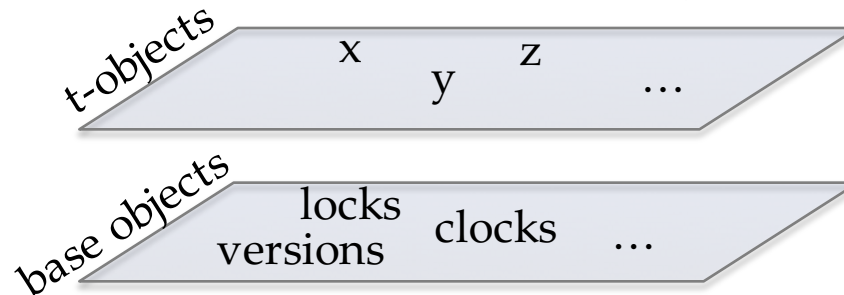
Sebastiano Peluso¹, Roberto Palmieri¹, Paolo Romano²,
Binoy Ravindran¹ and Francesco Quaglia³



Degrees of Parallelism

[SPAA08, SPAA09]

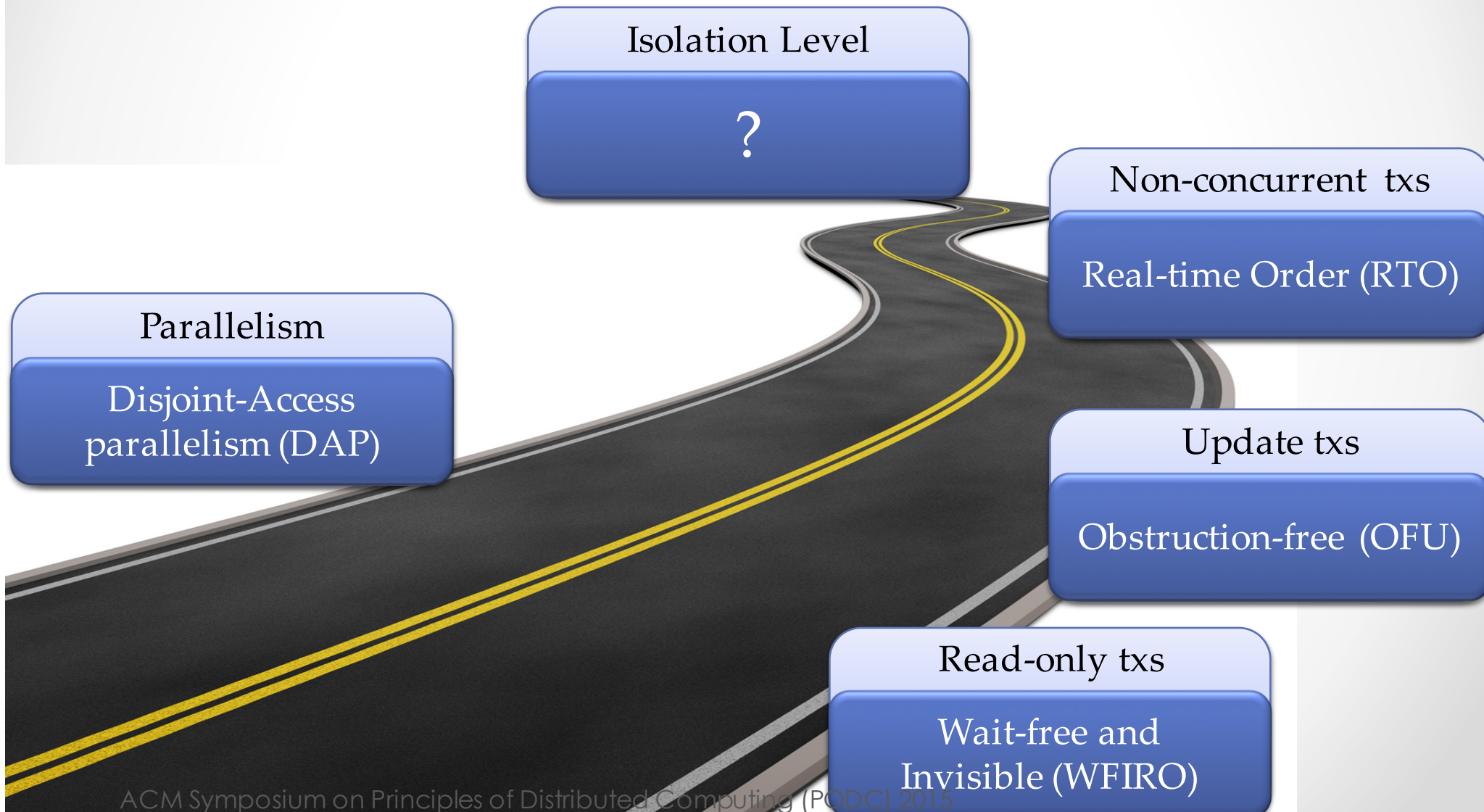
- **Strict Disjoint-Access Parallelism (S-DAP)**: Two transactions do not contend on a common base object if they do not access any common transactional object.
- **Weak Disjoint-Access Parallelism (W-DAP)**: Two transactions do not concurrently contend on a common base object if there is no path between them in the conflict graph.



S-DAP: T₁ and T₃ **cannot contend** on a common base object.

W-DAP: T₁ and T₃ **can concurrently contend** on a common base object.

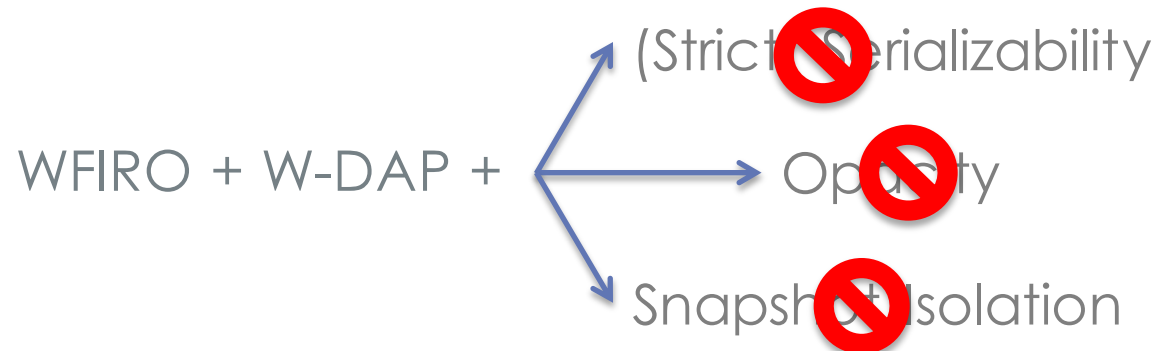
Desirable properties in TM



Existing Impossibility Results

Impossibility on Wait-free Invisible Read-Only (WFIRO)

[SPAA09, TCS11]



Lower Bound: Wait-free Read-only txs should write on at least $t-1$ base objects when $t+1$ processes execute.

Impossibility on Obstruction-free Updates (OFU)

[SPAA08, SPAA14]

OFU + S-DAP + \longrightarrow Serializability

The diagram shows the text 'OFU + S-DAP +' followed by a blue arrow pointing to the word 'Serializability'. The word 'Serializability' is crossed out with a large red 'X' inside a circle, indicating that the combination is impossible.

PCL Theorem: No TM can ensure S-DAP, Obstruction-freedom and Weak Adaptive Consistency (weaker than Snapshot Isolation and Processor Consistency).

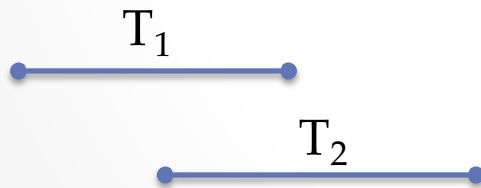
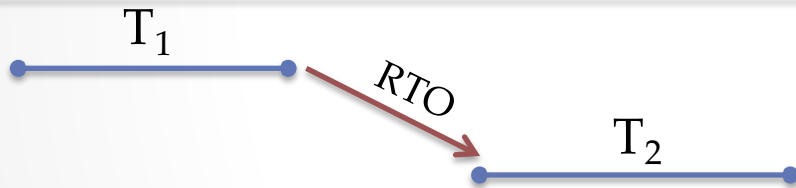
Roadmap of Results

- Impossibility Results
 - Real-Time Order +
 - Wait-free Read-only Txs + WDAP +
 1. Obstruction-free Update Txs
 2. Invisible Read-only Txs + Weakly Progressive Update Txs
 - Possibility Result
 - Witnessable Real-Time Order +
 - Extended Update Serializability +
 - SDAP +
 - Wait-free Invisible Read-only Txs +
 - Strongly Progressive Update Transactions
 - ... and corresponding costs
 - Space Complexity: $\Omega(\min(N_o, N_p))$ per data-version
 - Time Complexity: $\Omega(k * N_o)$ per read operation
- N_p = number of processes
 N_o = number of objects
 k = number of versions per object

What about Preserving Real-Time Order?

- **Real-time order (RTO) relation:** T_1 is ordered before T_2 if the commit of T_1 precedes the begin of T_2 .
- Transactions should appear as executed without violating the Real-time order [PPoPP08].

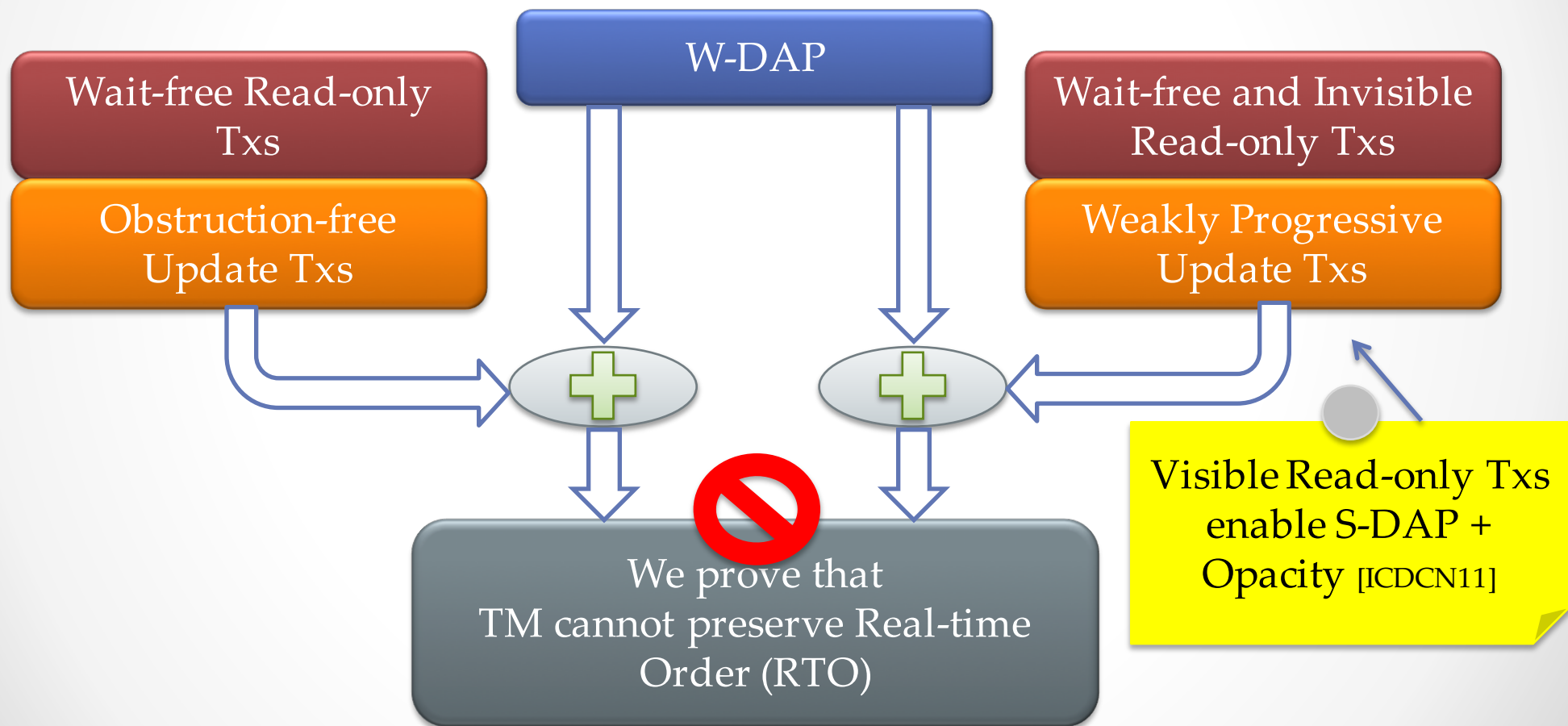
Executions



Possible Serialization Orders



Impossibility Results on Real-Time Order

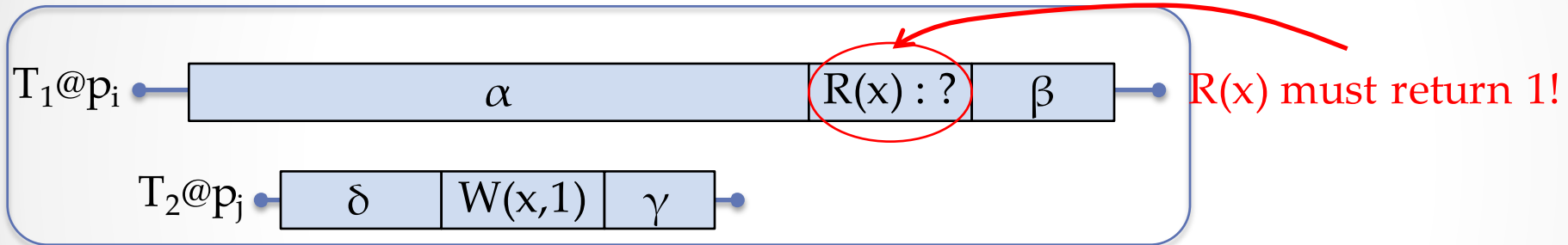


Sketch on the Impossibility:

What Should a Transaction Read?

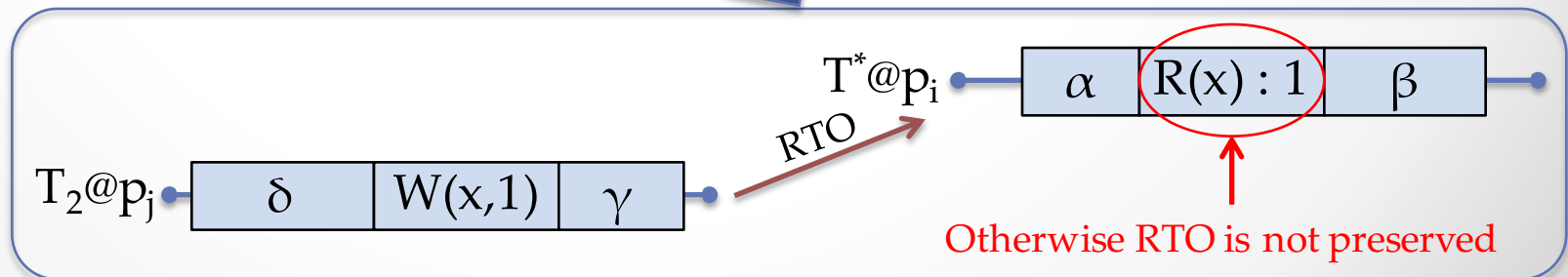
[Lemma 3 simplified] WDAP TM + WFRO + RTO

Read the last committed value!



- $\alpha, \beta, \delta, \gamma$ are non-conflicting intervals
- T_1 is a read-only transaction
- T_1 cannot abort because of WFRO

Indistinguishable to process p_i because of WDAP (and Lemmas 1, 2)



Sketch on the Impossibility: Violating the Real-Time Order

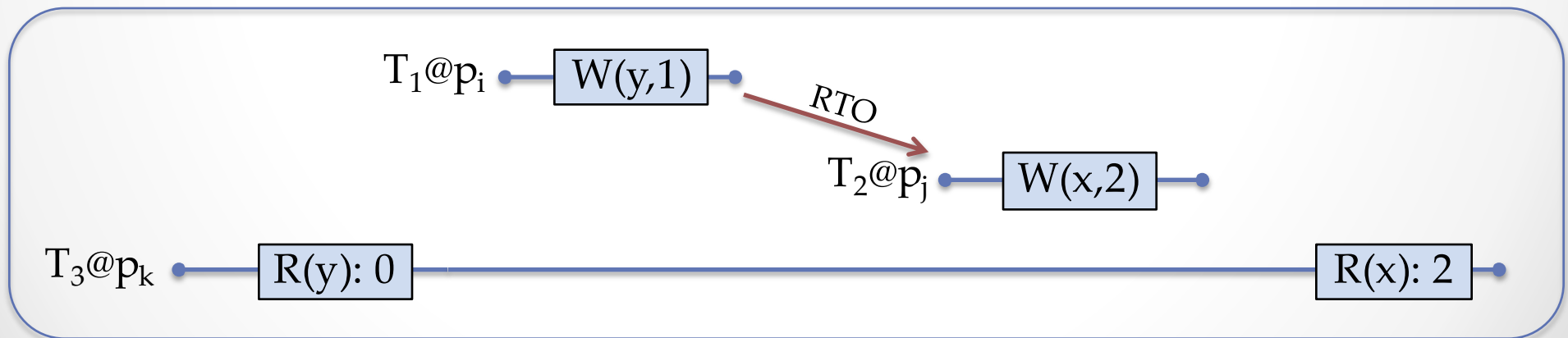
By contradiction we assume that WDAP + WFRO + RTO is possible, then...

Obstruction-free Update Txns

- T_1 commits because it runs solo and it cannot wait for T_3 's outcome to commit
- T_2 commits because it runs solo

Weakly Progressive Update Txns & Invisible Read-only Txns

- T_1 commits because it cannot detect the conflict with T_3
- T_2 commits because it does not encounter any conflict



- $R(x)$ must return 2 because of Lemma 3
- **Real-Time Order violated: T_2 appears as executed before T_1 !**

Roadmap of Results

- Impossibility Results
 - Real-Time Order +
 - Wait-free Read-only Txs + WDAP +
 1. Obstruction-free Update Txs
 2. Invisible Read-only Txs + Weakly Progressive Update Txs

- Possibility Result
 - Witnessable Real-Time Order +
 - Extended Update Serializability +
 - SDAP +
 - Wait-free Invisible Read-only Txs +
 - Strongly Progressive Update Transactions

- ... and corresponding costs
 - Space Complexity: $\Omega(\min(N_o, N_p))$ per data-version
 - Time Complexity: $\Omega(k \cdot N_o)$ per read operation

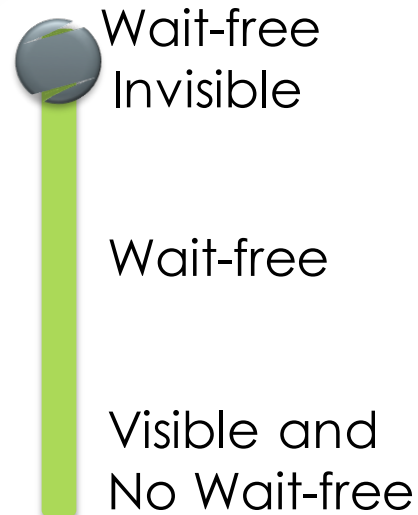
N_p = number of processes
 N_o = number of objects
 k = number of versions per object

A Strictly DAP TM Design

PARALLELISM



READ-ONLY



UPDATES



ISOLATION



Witnessable Real-Time Order (WRTO)

Real-Time Order preserved only among directly conflicting transactions, and transactions executed by the same process

- Properties of Extended Update Serializability:
 - All transactions observe a serializable history (like Opacity!) ☒
 - Committed Update Transactions are serializable (like Opacity!) ☒
 - Two transactions T_1, T_2 can observe two non-compatible serialization orders... ☒
 - ...but only if T_1 and T_2 will never commit any write operation ☐



Roadmap of Results

- Impossibility Results
 - Real-Time Order +
 - Wait-free Read-only Txs + WDAP +
 1. Obstruction-free Update Txs
 2. Invisible Read-only Txs + Weakly Progressive Update Txs
 - Possibility Result
 - Witnessable Real-Time Order +
 - Extended Update Serializability +
 - SDAP +
 - Wait-free Invisible Read-only Txs +
 - Strongly Progressive Update Transactions
 - ... and corresponding costs
 - Space Complexity: $\Omega(\min(N_o, N_p))$ per data-version
 - Time Complexity: $\Omega(k \cdot N_o)$ per read operation
- N_p = number of processes
 N_o = number of objects
 k = number of versions per object

Detected Costs



N_p = number of processes

N_o = number of objects

k = number of versions per object

- The TM requires to store a vector clock of size N_p per data-version.
- A read operation by a read-only T may validate T 's read-set to check if it can return a data-version.

Space Complexity

$O(N_p)$ integers for each version of a transactional object

Time Complexity

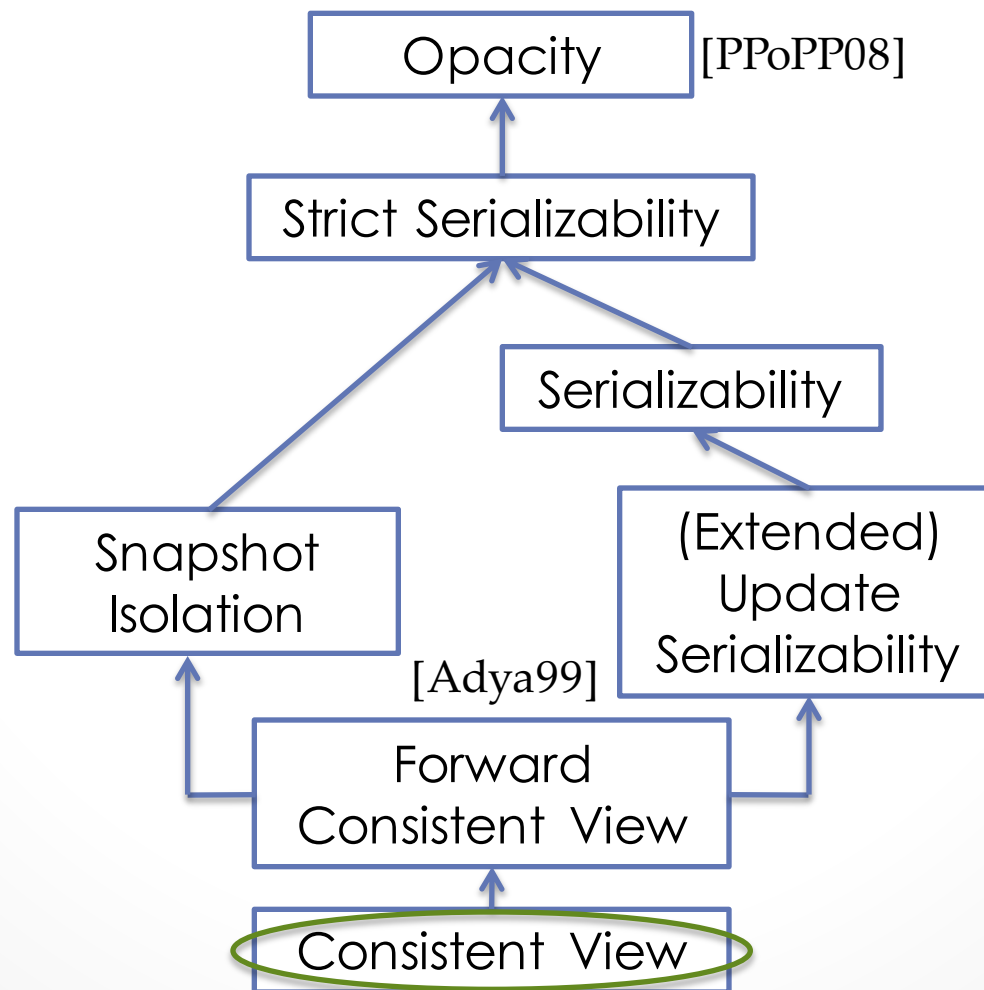
$O(k*N_o)$ steps to execute a read operation of a read-only tx

Are these costs necessary?



Consistent View

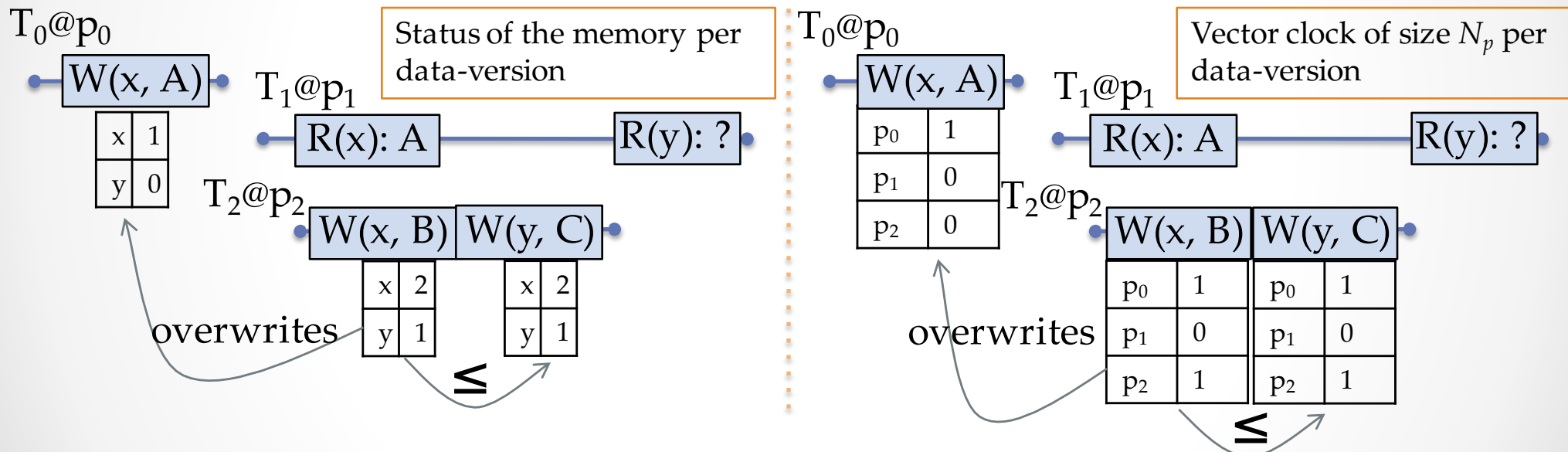
- **Consistent View:** all transactions read from a causally consistent snapshot. → It is weaker than EUS.



Lower Bound on the Space Complexity

- Implementation of reads: return version V if V “does not depend on” a version V^* that **overwrites** the read-set.

Intuition on possible S-DAP implementations

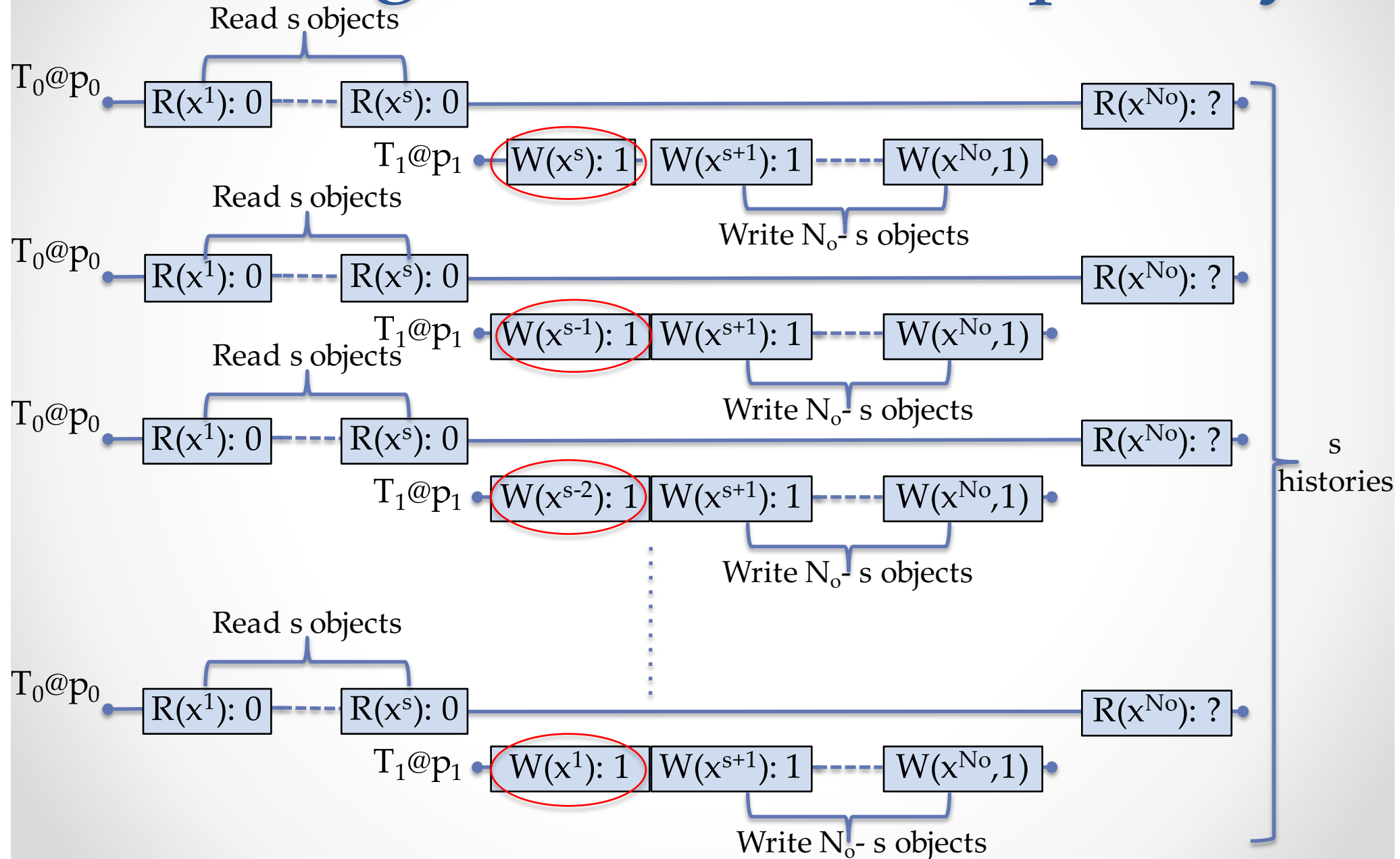


Theorem. SDAP TM that guarantees WFIRO, Consistent View, WRTO and either obstruction-free or weakly progressive update transactions



The space complexity for each version of a transactional object is
 $\Omega(\min(N_o, N_p))$

Proving the Time Complexity



Lower Bound on the Time Complexity

- In a WDAP TM with invisible read-only transactions $R(x^{N_o})$ cannot distinguish the s histories unless it analyzes T_0 's read-set.
- Consistent View + WRT0 + WFIRO: T_0 has to perform s steps to check if $R(x^{N_o})$ can return 1.
- s can be equal to $N_o - 1$...and a read can check k versions...

Theorem. WDAP multi-version TM that guarantees WFIRO, Consistent View, WRT0 and either obstruction-free or weakly progressive update transactions



The read-only time complexity is $\Omega(k * N_o)$

Read-only time complexity: the maximum number of steps performed by any read operation of any read-only transaction

Related Results

Lower Bound on the Number of Visible Reads [SPAA09, TCS11]

Hypothesis: WDAP TM + Strict Serializability + Minimal Progressive Update txs.

Result: Wait-free Read-only txs should write on at least $t-1$ base objects when $t+1$ processes execute in a WDAP TM.

Relation: The result is only a necessary condition in case Update txs are Obstruction-free since RTO cannot be preserved. The result is sufficient in case Update txs are Weakly Progressive [ICDCN11].

Impossibility Result in Multi-Version-Permissive TM [PODC10]

Hypothesis: WDAP TM + MV-Permissiveness

Result: No Strict Serializability.

Relation: In a parasitic-free environment MV-Permissiveness could provide Wait-free and Invisible Read-only txs and Weakly Progressive Update Txs. In that case RTO cannot be preserved.

Time Complexity in Progressive TM [PPoPP08]

Hypothesis: Progressive single-version TM + Opacity

Result: The maximum number of steps performed by any operation is $\Omega(N_o)$

Relation: Same cost to check if a version is observable in a multi-version WDAP TM with WRTO, wait-free and invisible read-only txs and Consistent View.

Conclusions

- Two **impossibility results** ruling DAP TM that guarantees Real-Time Order and a set of desirable progress properties.
- **Possibility result**: an SDAP TM that provides a strong correctness property (i.e., EUS), Witnessable Real-Time Order, and the same set of progress properties as before.
- **Lower bounds** on the space and time complexity of such a DAP TM.

Our Message

In a DAP TM lowering the consistency property is not enough. Rather either invisible reads or wait-freedom must be sacrificed!

Thanks for the attention



peluso@vt.edu

References

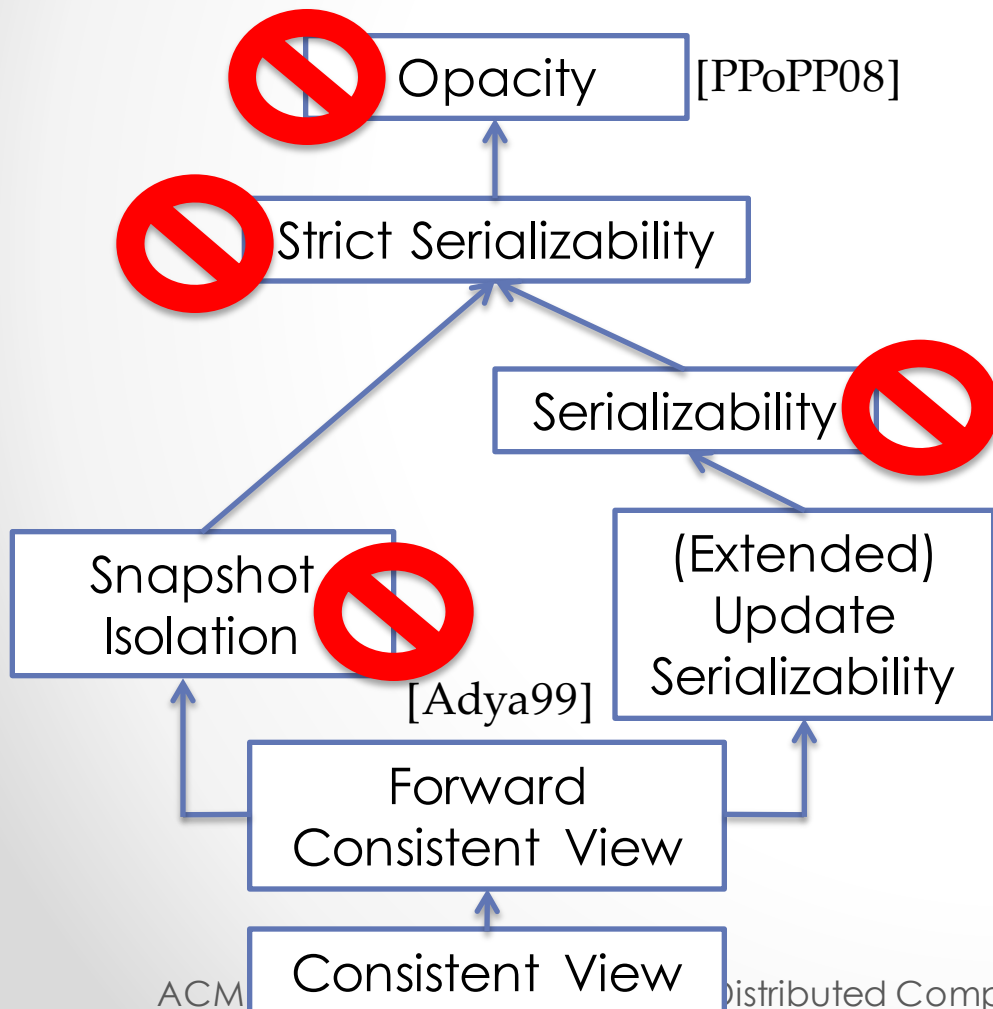
- [Adya99] A. Adya. “Weak Consistency: A Generalized Theory and Optimistic Implementations for Distributed Transactions”. PhD thesis, 1999. AAI0800775.
- [ICDCN11] H. Attiya and E. Hillel. “Single version STMs can be multi-version permissive”. In proc. of the 12th International Conference on Distributed Computing and Networking, ICDCN, 2011.
- [PODC10]
- [POPL09] R. Guerraoui and M. Kapalka. “The Semantics of Progress in Lock-based Transactional Memory”. In proc. of the 36th annual ACM SIGPLAN-SIGACT symposium on Principles of programming languages (POPL), 2009.
- [PPoPP08] R. Guerraoui and M. Kapalka. “On the Correctness of Transactional Memory”. In proc. of the 13th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming (PPoPP), 2008.
- [SPAA08] R. Guerraoui and M. Kapalka. “On Obstruction-free Transactions.” In proc. of the 20th annual Symposium on Parallelism in Algorithms and Architectures (SPAA), 2008.
- [SPAA09] Hagit Attiya, Eshcar Hillel, and Alessia Milani. “Inherent limitations on disjoint-access parallel implementations of transactional memory.” In proc. of the 21st annual Symposium on Parallelism in Algorithms and Architectures (SPAA), 2009.
- [SPAA14] Victor Bushkov, Dmytro Dziuina, Panagiota Fatourou, Rachid Guerraoui, “The PCL Theorem. Transactions cannot be Parallel, Consistent and Live.”. In proc. of the 26th annual Symposium on Parallelism in Algorithms and Architectures (SPAA), 2014.
- [TCS11] Hagit Attiya, Eshcar Hillel, and Alessia Milani. “Inherent limitations on disjoint-access parallel implementations of transactional memory”. Theory Comput. Syst., 49(4):688–718, 2011.

Backup slides

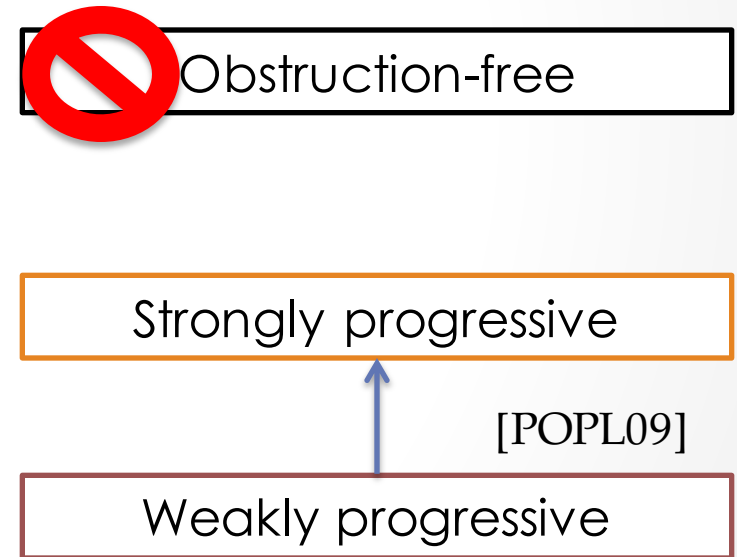
Seeking a Sweet Spot in Disjoint-Access Parallel TM

- Objective: S-DAP + WFIRO

Isolation Levels



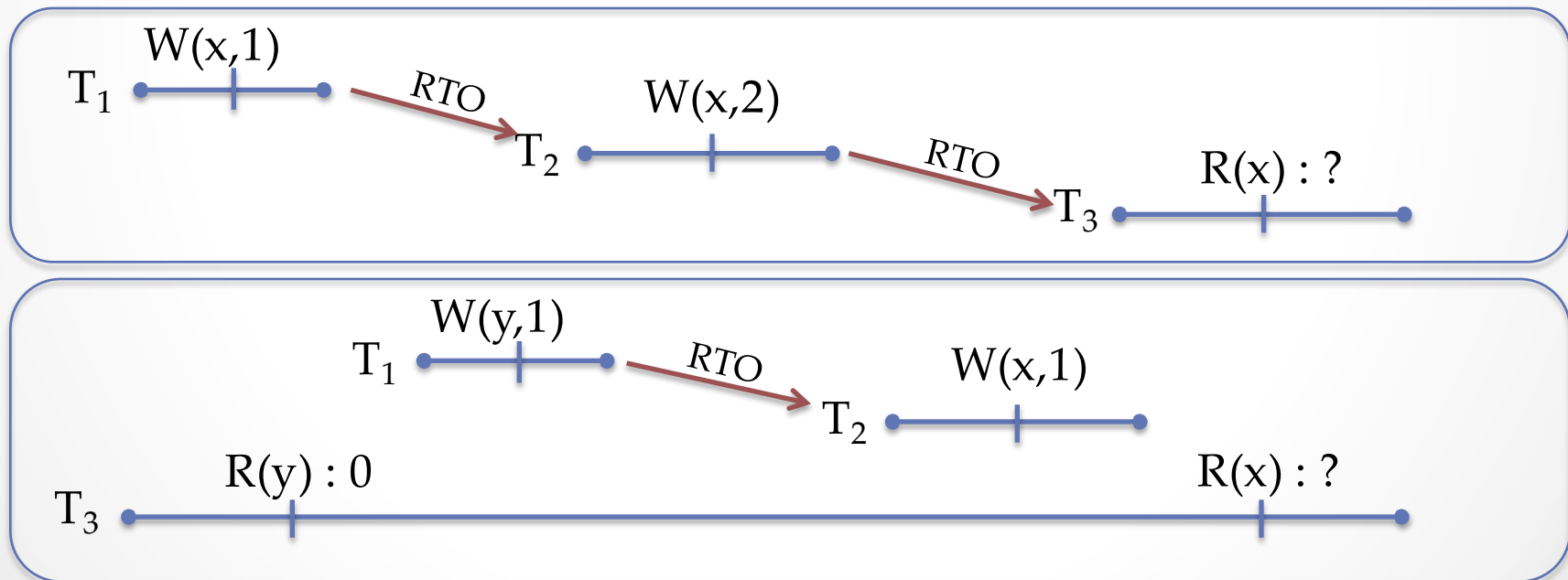
Progress Guarantees for Update Txs



What about Preserving Real-Time Order?

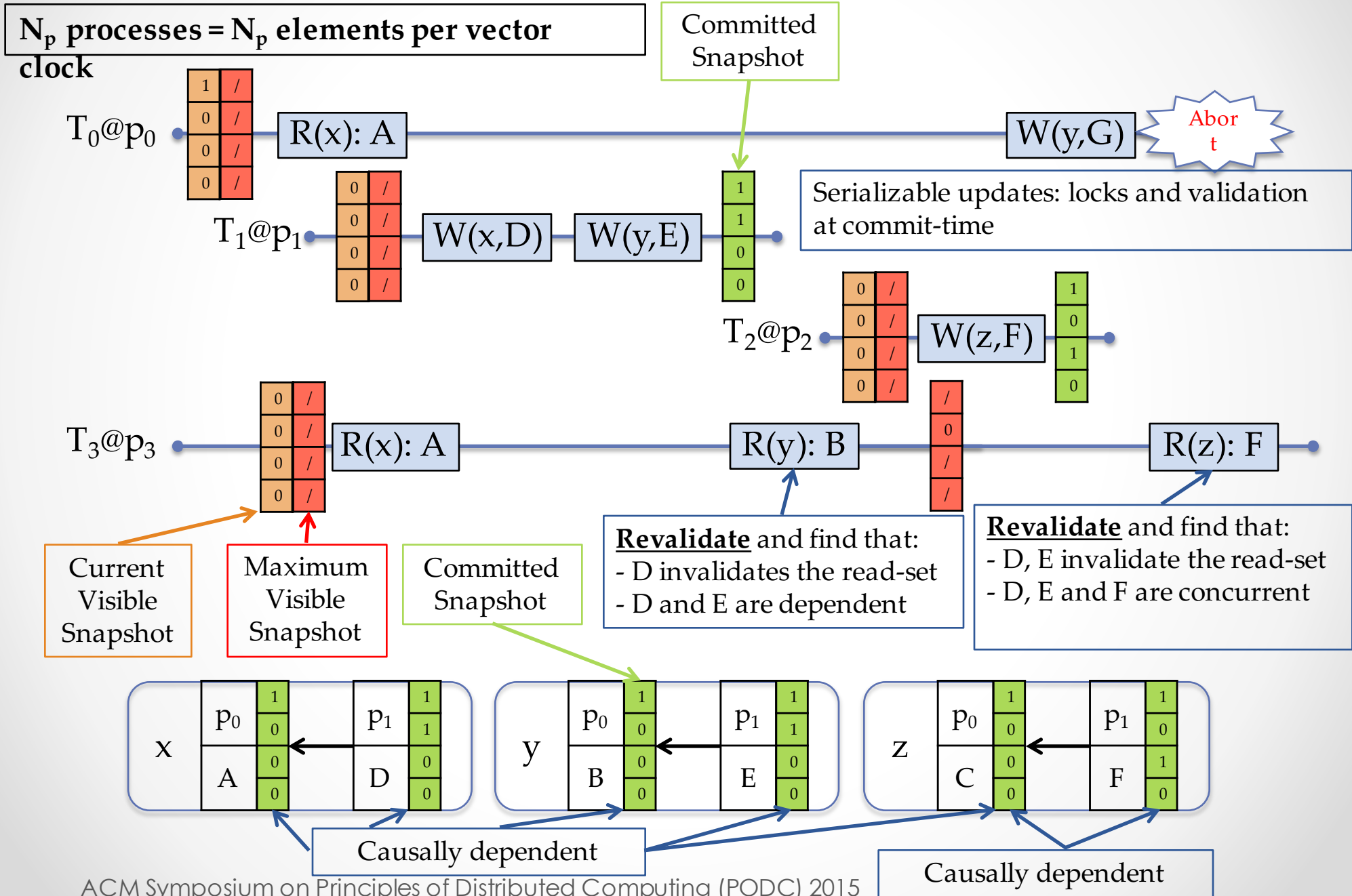
- **Real-time order (RTO) relation:** T_1 is ordered before T_2 if the commit of T_1 precedes the begin of T_2 .
- Transactions should appear as executed without violating the Real-time order [PPoPP08].

Examples



Real-time order violated if $R(x)$ returned 1.

Details about the SDAP TM



Consistent View

- **Consistent View:** all transactions read from a causally consistent snapshot. \rightarrow It is weaker than EUS.
- **Forbidden Read:** the read creates an oriented cycle with exactly one write-after-read edge in the conflict graph [Adya99].

Example

