

Brief Announcement: Breaching the Wall of Impossibility Results on Disjoint-Access Parallel TM

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

¹



VirginiaTech
Invent the Future

²



³

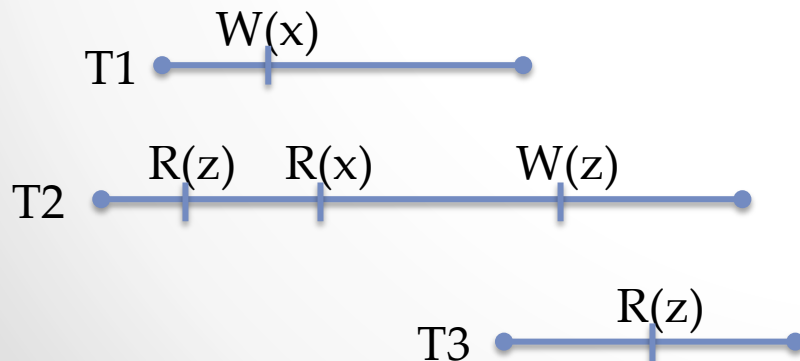


SAPIENZA
UNIVERSITÀ DI ROMA

Degrees of Parallelism

[SPAA08, SPAA09]

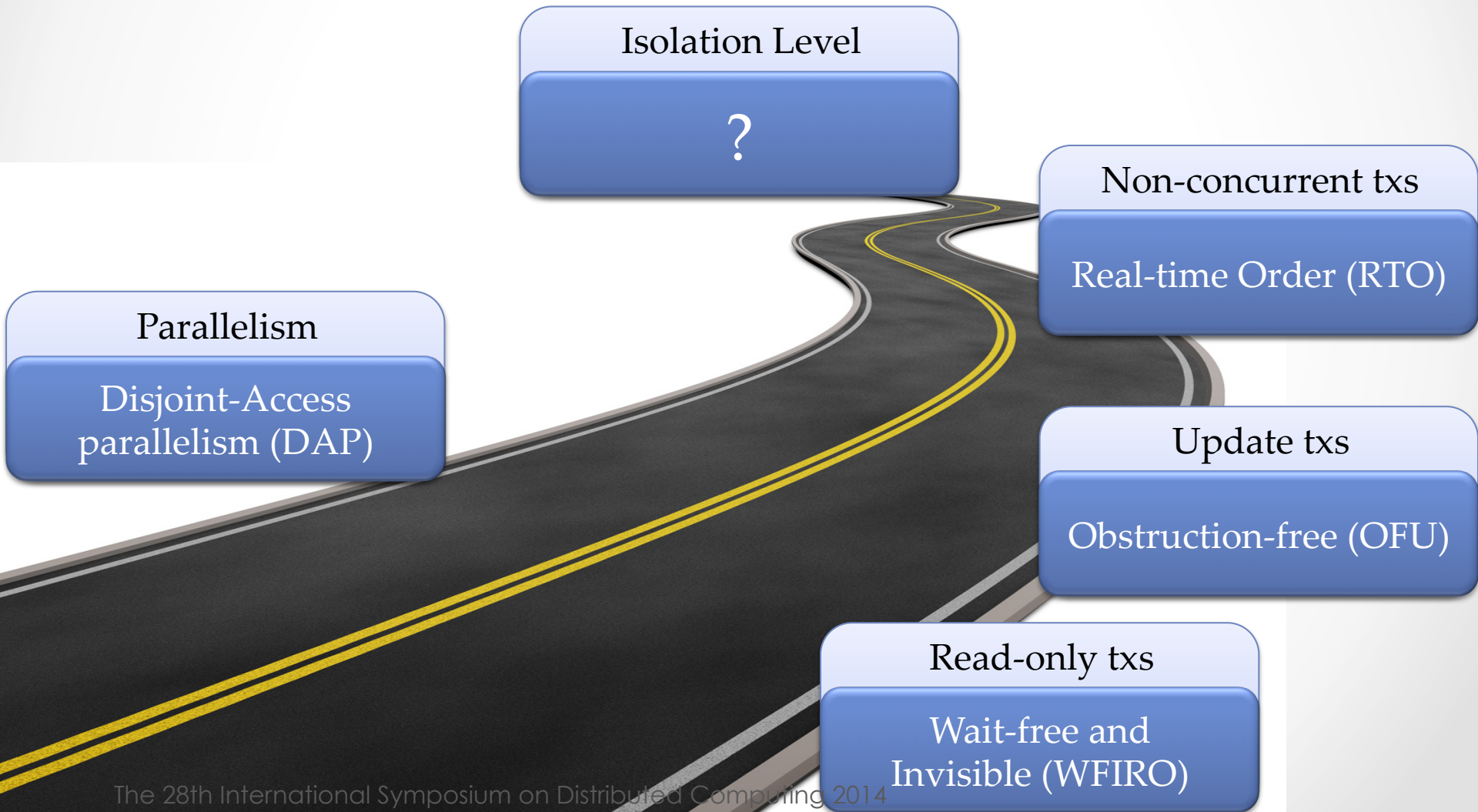
- **Strict Disjoint-Access Parallelism (S-DAP)**: Two transactions do not contend on a common base object if they do not access any common transactional object.
- **Weak Disjoint-Access Parallelism (W-DAP)**: Two transactions do not concurrently contend on a common base object if there is no path between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

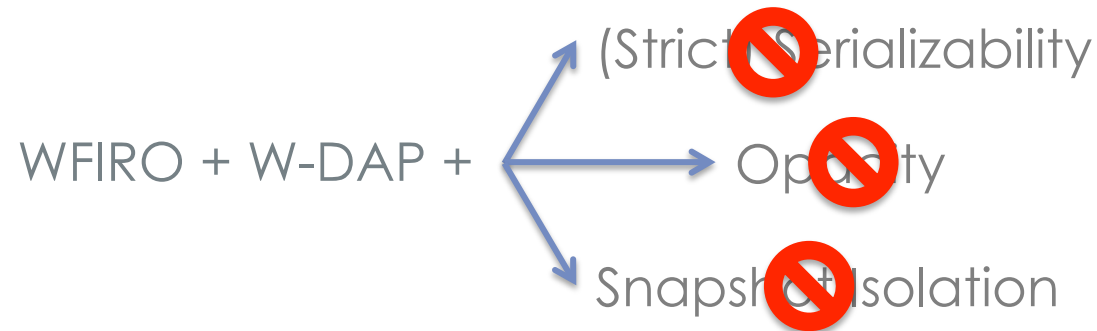
Desirable properties in TM



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]

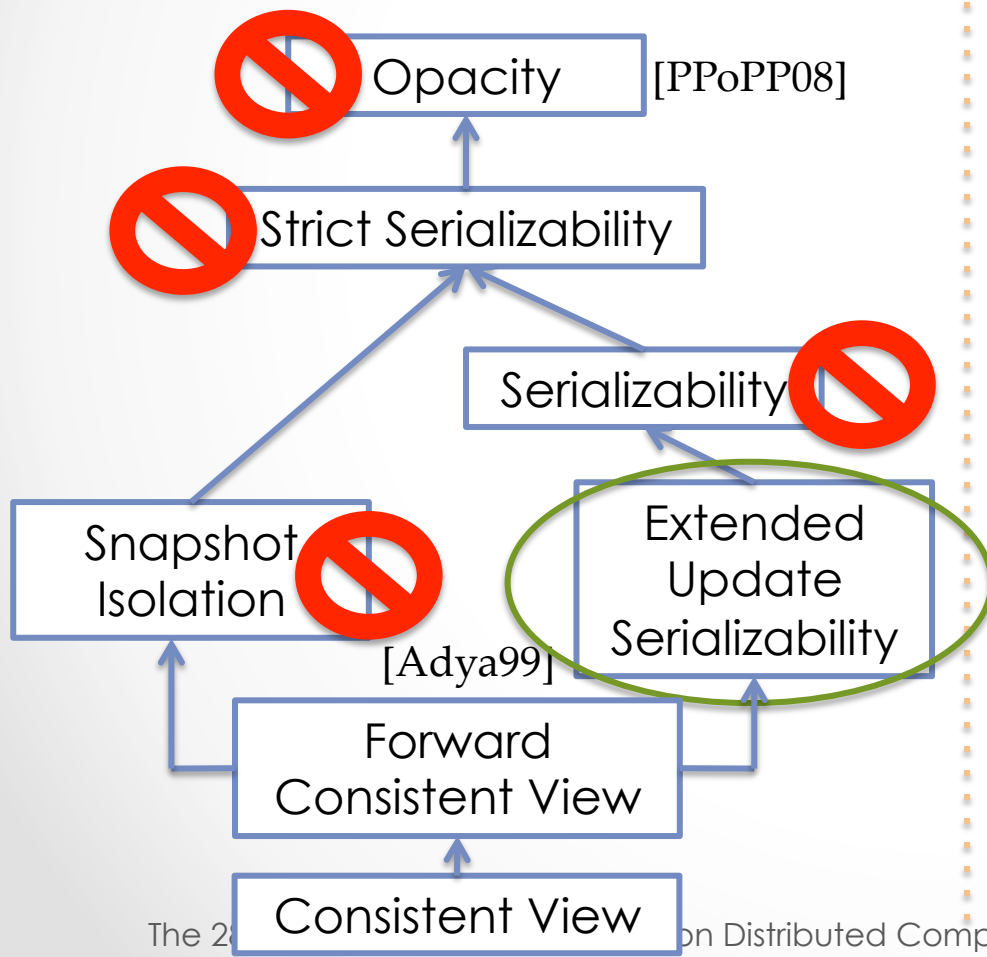
OFU + S-DAP +  Serializability

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

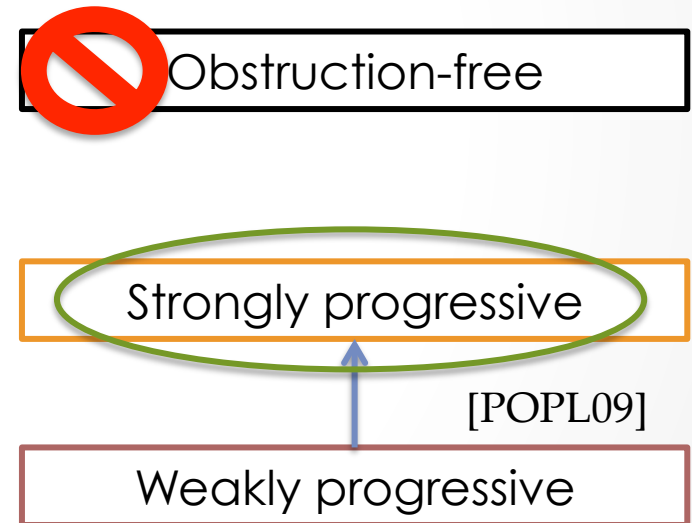
Seeking a Sweet Spot in Disjoint-Access Parallel TM

- Objective: S-DAP + WFIRO

Isolation Levels

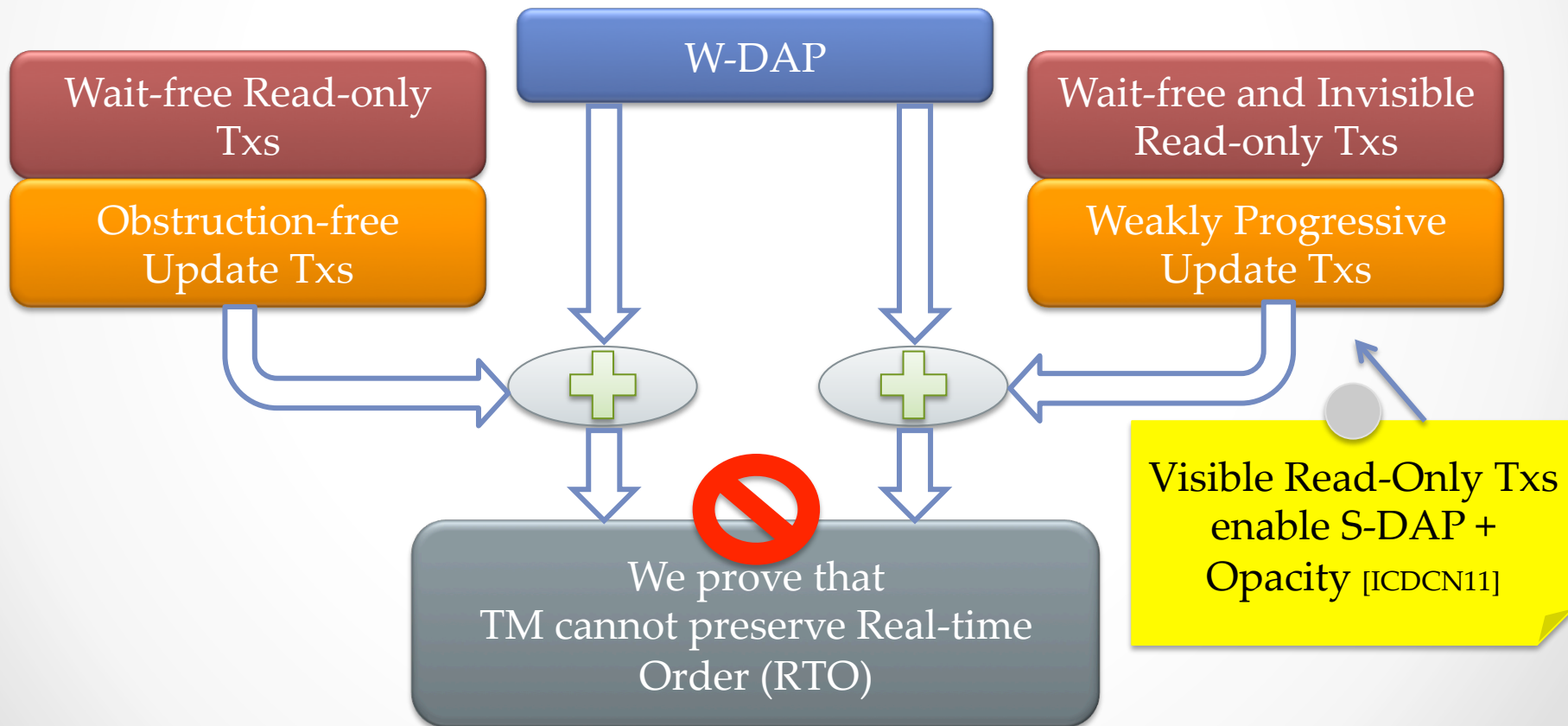


Progress Guarantees for Update Txs



What about Preserving Real-Time Order?

- Real-time order (RTO) relation: T_1 precedes T_2 if the commit of T_1 precedes the begin of T_2 .



A Strictly DAP™

PARALLELISM

READ-ONLY

UPDATES

ISOLATION

 S-DAP

W-DAP

NO DAP

 Wait-free
Invisible


Wait-free

Visible and
No Wait-free

 Obstruction-
free

 Strongly
Progressive

Weakly
Progressive

 Opacity
SER



 EUS
SI

Weak
Adaptive

Witnessable Real-Time Order (WRTO)

Real-Time Order preserved only among directly conflicting transactions

- Properties of EUS:

- Consistent View guaranteed for all transactions (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 

Thanks for the attention

peluso@vt.edu

References

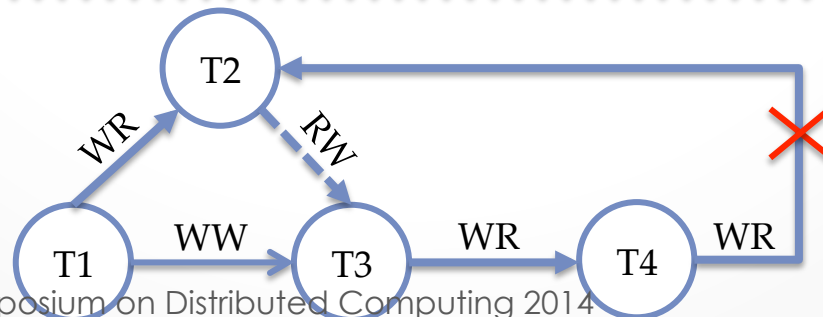
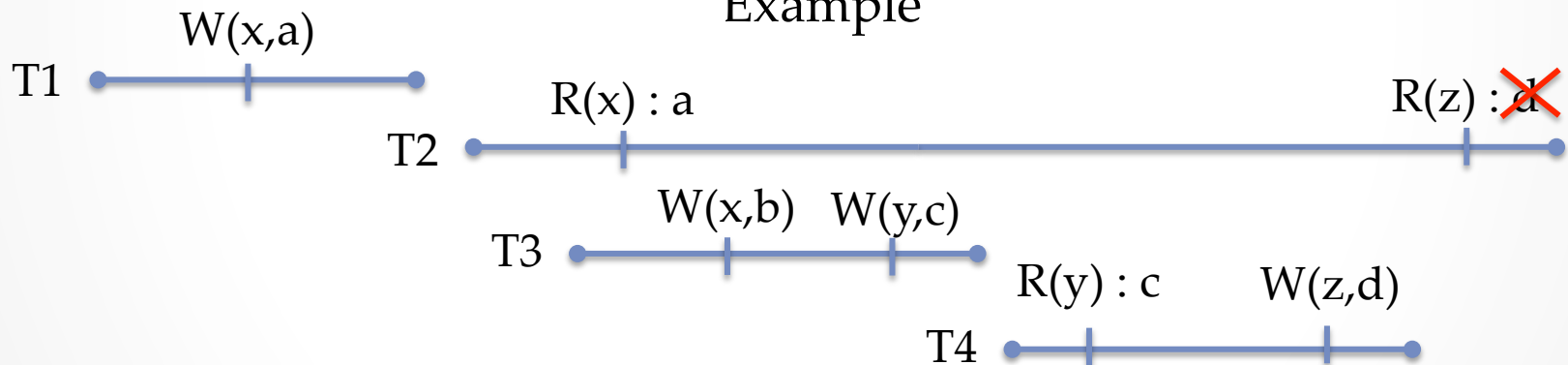
- [Adya99] A. Adya. “Weak Consistency: A Generalized Theory and Optimistic Implementations for Distributed Transactions”. PhD thesis, 1999. AAI0800775.
- [ICDCN11] H. Attiya and E. Hillel. “Single version STMs can be multi-version permissive”. In proc. of the 12th International Conference on Distributed Computing and Networking, ICDCN, 2011.
- [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 Dziuza, 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):698–719, 2011.

Examples

Consistent View

- **Consistent View**: all transactions are provided with a consistent view of the transactional state.
- **Detecting a Non-Consistent Read**: the read creates an oriented cycle with exactly one anti-dependence edge in the conflict graph [Adya99].

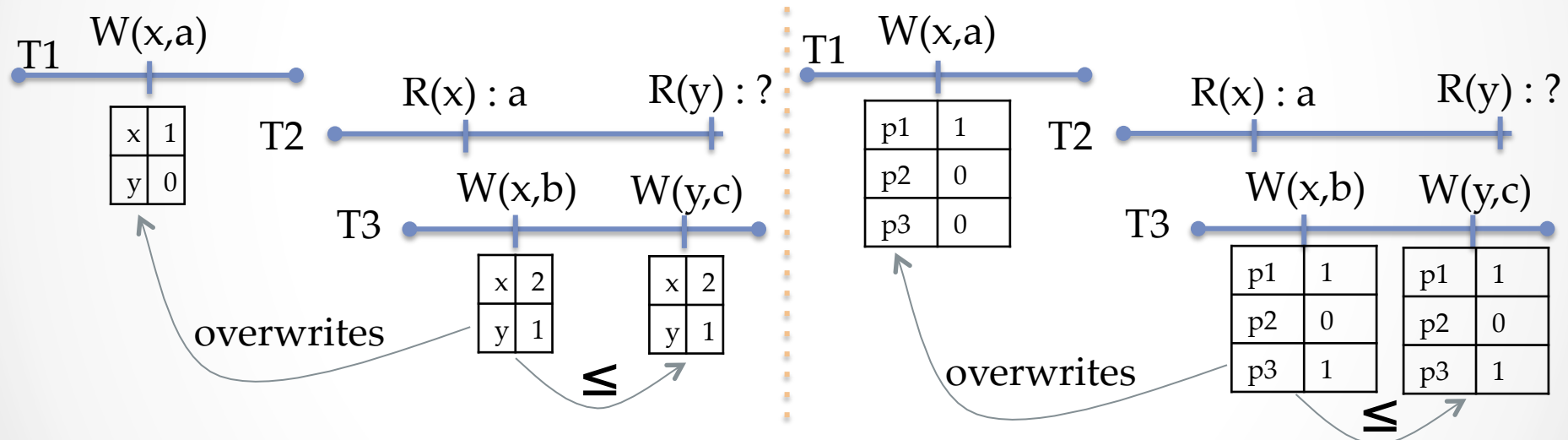
Example



The Costs of Ensuring Consistent View

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

Example of S-DAP implementations



Theorem. Given a S-DAP and Weakly Progressive STM that guarantees WFIRO and ensures consistent view, the space complexity for each version of an object is $\Omega(\min(N_o, N_p))$, where:

- N_o = number of objects
- N_p = number of processes