

# 面向分布式系统的复制数据类型理论研究概述

## (CCF 2018 第九届优博论坛)

魏恒峰

南京大学软件所

2018 年 08 月 08 日



# 面向分布式系统的复制数据类型理论研究概述

① 研究背景

② 两份工作

# 面向分布式系统的复制数据类型理论研究概述

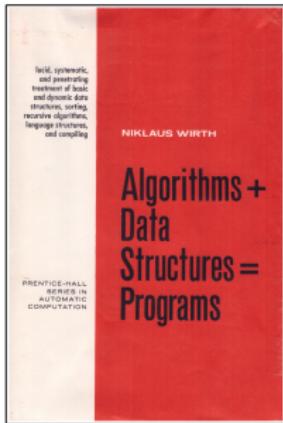
① 研究背景

② 两份工作

# Abstract Data Types (ADT)

[Liskov and Zilles, 1974]

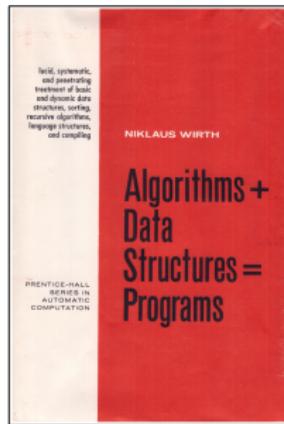
(单线程; 顺序语义)



## Abstract Data Types (ADT)

[Liskov and Zilles, 1974]

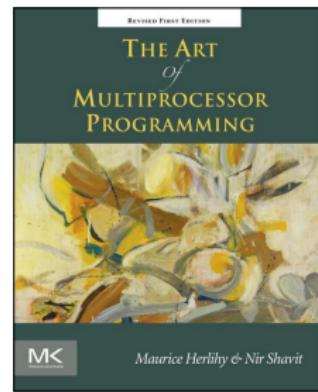
(单线程; 顺序语义)



## Concurrent Data Types

[Herlihy and Wing, 1990]

(多线程; 并发语义)



## Replicated Data Types (RDT; $\approx$ 2010 年)

[Burckhardt et al., 2014]

(多副本; 复制语义)

### Replicated Data Types: Specification, Verification, Optimality

Sebastian Burckhardt

Alexey Gotsman

Hongseok Yang

Marek Zawirski

## Replicated Data Types (RDT; $\approx$ 2010 年)

[Burckhardt et al., 2014]

(多副本; 复制语义)

### Replicated Data Types: Specification, Verification, Optimality

Sebastian Burckhardt

Alexey Gotsman

Hongseok Yang

Marek Zawirski

新平台, 新问题, 新挑战



## 新浪微博社交应用<sup>1</sup>:

- ▶ 日均用户近一亿名
- ▶ 日均消息近一亿条

---

<sup>1</sup>2015 第三季度; 数据来自 China Internet Watch.



## 新浪微博社交应用<sup>1</sup>:

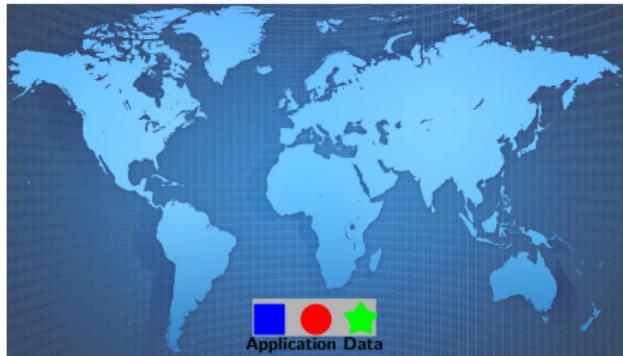
- ▶ 日均用户近一亿名
- ▶ 日均消息近一亿条

## 底层数据服务系统特性需求 ( $H^3L$ ):

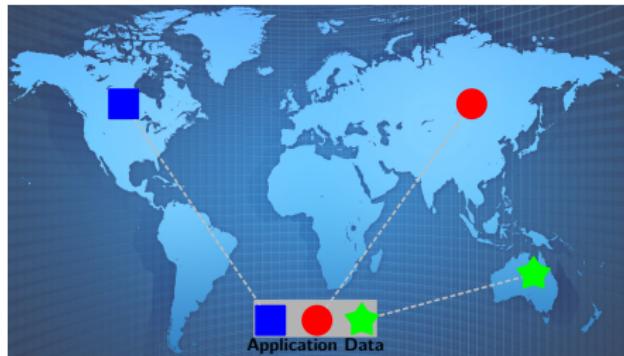
- ▶ 低延迟, 高可用性 (4个9<sup>2</sup>)
- ▶ 高容错性, 高可扩展性

<sup>1</sup>2015第三季度; 数据来自 China Internet Watch.

<sup>2</sup>数据来自 InfoQ.

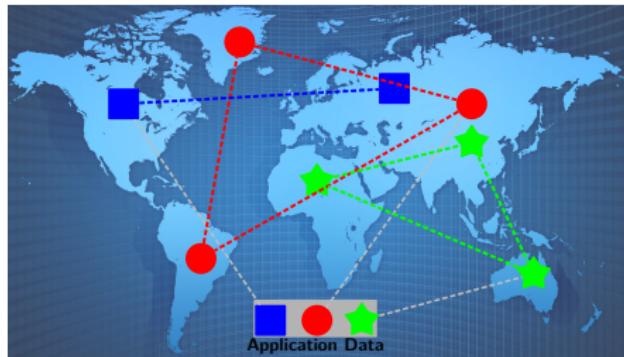


分布数据 (distributed data):



## 分布数据 (distributed data):

1. 分区 (partition): 水平扩展

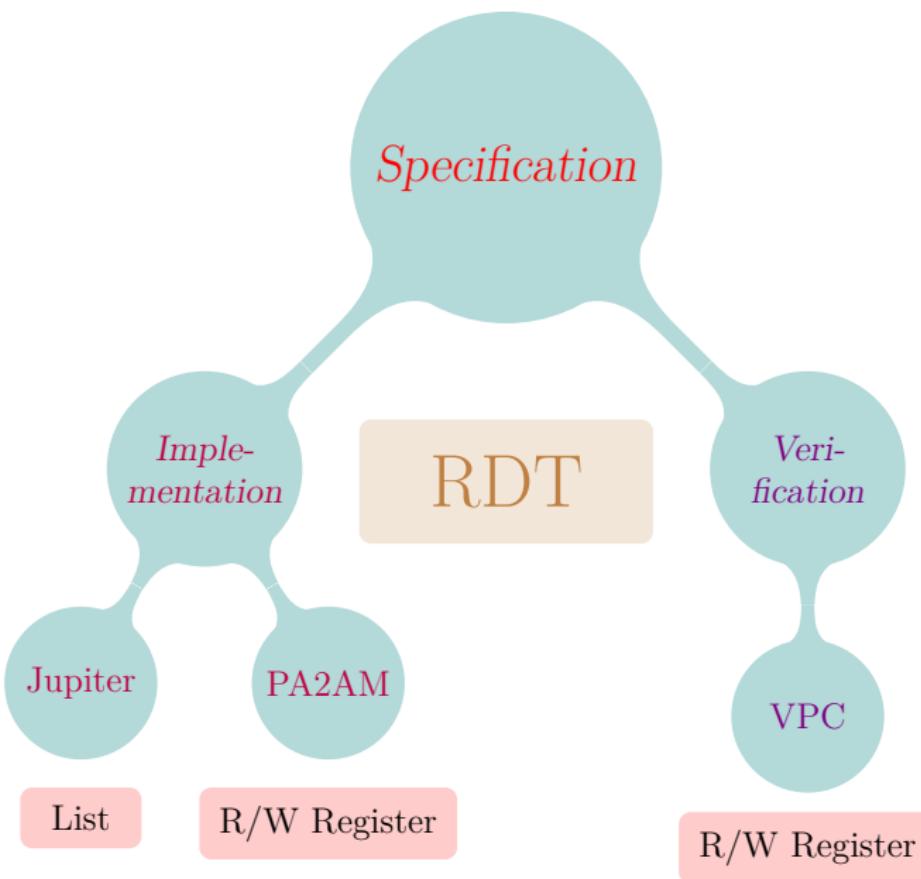


## 分布数据 (distributed data):

1. 分区 (partition): 水平扩展
2. 副本 (replication) : 就近访问, 容灾备份

## 复制数据类型 [Shapiro et al., 2011a]

- ▶ Read/Write Register
- ▶ Counter
- ▶ Set
- ▶ List
- ▶ Hash Map
- ▶ Disjoint Set
- ▶ Graph



# 面向分布式系统的复制数据类型理论研究概述

① 研究背景

② 两份工作

## Brief Announcement @ PODC'2018 <sup>3</sup>

实现复制列表的 Jupiter 协议 [Nichols et al., 1995]<sup>a</sup> 满足  
weak list specification [Attiya et al., 2016]<sup>b</sup>.

---

<sup>a</sup>David A. Nichols et al. (1995). "High-latency, Low-bandwidth Windowing in the Jupiter Collaboration System". In: *Proceedings of the 8th Annual ACM Symposium on User Interface and Software Technology*. UIST '95. ACM, pp. 111–120.

<sup>b</sup>Hagit Attiya et al. (2016). "Specification and complexity of collaborative text editing". In: *Proceedings of the 2016 ACM Symposium on Principles of Distributed Computing*. PODC '16. ACM, pp. 259–268.

## Brief Announcement @ PODC'2018 <sup>3</sup>

实现复制列表的 Jupiter 协议 [Nichols et al., 1995]<sup>a</sup> 满足  
weak list specification [Attiya et al., 2016]<sup>b</sup>.

---

<sup>a</sup>David A. Nichols et al. (1995). "High-latency, Low-bandwidth Windowing in the Jupiter Collaboration System". In: *Proceedings of the 8th Annual ACM Symposium on User Interface and Software Technology*. UIST '95. ACM, pp. 111–120.

<sup>b</sup>Hagit Attiya et al. (2016). "Specification and complexity of collaborative text editing". In: *Proceedings of the 2016 ACM Symposium on Principles of Distributed Computing*. PODC '16. ACM, pp. 259–268.

---

## 3 藏在脚注里的猜想 @ PODC'2016 [Attiya et al., 2016]

# Weak List Specification

# 基于副本的协同文本编辑系统



(a) Google Docs



(b) Apache Wave



(c) Wikipedia



(d) LATEX Editor

## 复制列表对象：建模编辑系统的核心功能

$\text{INS}(a, p)$ ：在  $p$  位置插入元素  $a$

$\text{DEL}(p)$ ：删除  $p$  位置上的元素

$\text{READ}$ ：返回该列表

定义 (最终收敛性 (Eventual Convergence) [Ellis and Gibbs, 1989])

当用户不再提交更新操作时, 每个 *replica* 上的列表是相同的。

定义 (最终收敛性 (Eventual Consistency) [Ellis and Gibbs, 1989])

当用户不再提交更新操作时, 每个 *replica* 上的列表是相同的。

定义 (强最终一致性 (Strong Eventual Consistency) [Shapiro et al., 2011b])

如果两个 *replica* 处理了同一组用户操作, 那么这两个 *replica* 上对列表是相同的。

定义 (最终收敛性 (Eventual Convergence) [Ellis and Gibbs, 1989])

当用户不再提交更新操作时, 每个 *replica* 上的列表是相同的。

定义 (强最终一致性 (Strong Eventual Consistency) [Shapiro et al., 2011b])

如果两个 *replica* 处理了同一组用户操作, 那么这两个 *replica* 上对列表是相同的。

对系统的中间状态缺少足够的约束

## Specification and Complexity of Collaborative Text Editing

Hagit Attiya  
Technion

Sebastian Burckhardt  
Microsoft Research

Alexey Gotsman  
IMDEA Software Institute

Adam Morrison  
Technion

Hongseok Yang  
University of Oxford

Marek Zawirski<sup>\*</sup>  
Inria & Sorbonne Universités,  
UPMC Univ Paris 06, LIP6

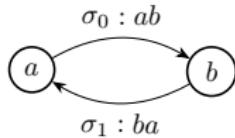
定义 (Weak List Specification  $\mathcal{A}_{\text{weak}}$  [Attiya et al., 2016])

*Informally,  $\mathcal{A}_{\text{weak}}$  requires the ordering between elements that are not deleted to be consistent across the system.*

定义在系统所有列表状态上的全局性质

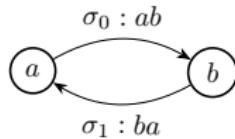
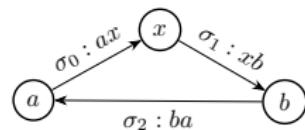
## 定义 (状态对兼容性 (Pairwise State Compatibility Property))

任给两个列表状态  $\sigma_0$ 、 $\sigma_1$ , 若它们含有两个共同元素  $a$ 、 $b$ ,  
则  $a$ 、 $b$  在  $\sigma_0$  与  $\sigma_1$  中的相对顺序保持一致。

$$\boxed{\sigma_0 : ab}$$
$$\boxed{\sigma_1 : ba}$$


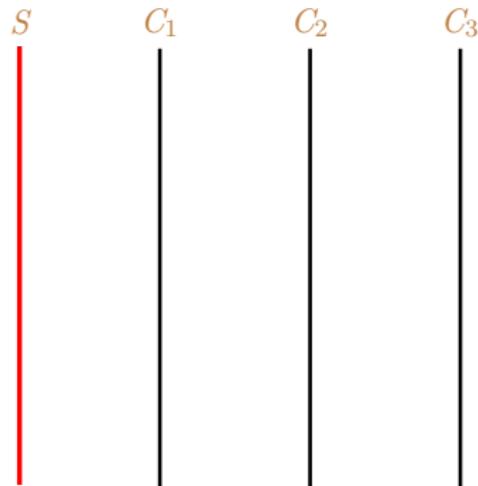
## 定义 (状态对兼容性 (Pairwise State Compatibility Property))

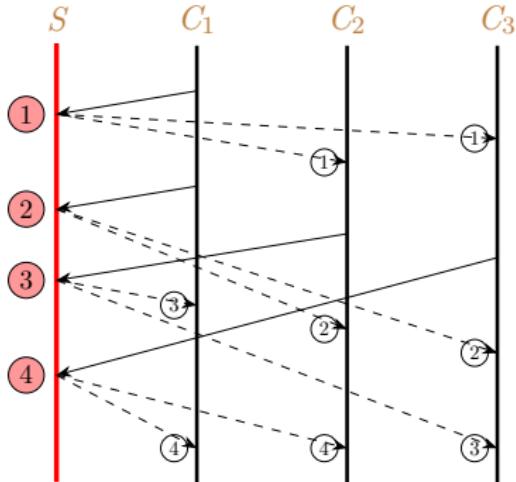
任给两个列表状态  $\sigma_0, \sigma_1$ , 若它们含有两个共同元素  $a, b$ ,  
则  $a, b$  在  $\sigma_0$  与  $\sigma_1$  中的相对顺序保持一致。

 $\sigma_0 : ab$  $\sigma_1 : ba$  $\sigma_0 : ax$  $\sigma_1 : xb$  $\sigma_2 : ba$ 

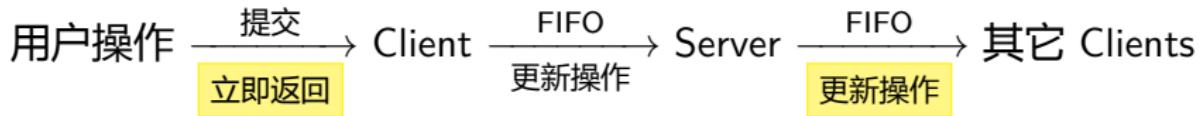
# Jupiter

$(n + 1)$  replica  $\triangleq$   $(n)$  Client +  $(1)$  Server [Nichols et al., 1995]



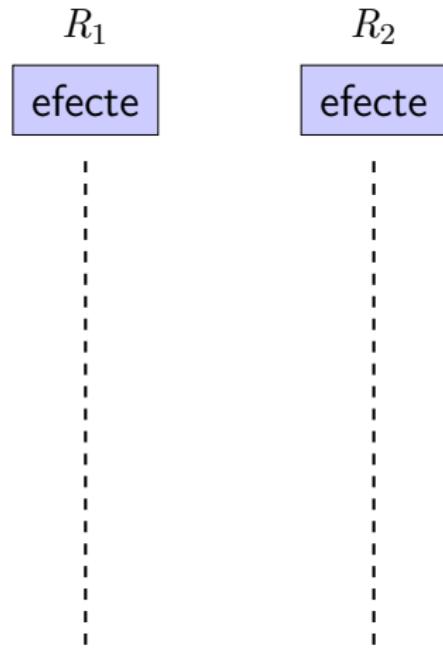
$$(n+1) \text{ replica} \triangleq (n) \text{ Client} + (1) \text{ Server} \text{ [Nichols et al., 1995]}$$


Server 负责将所有操作序列化

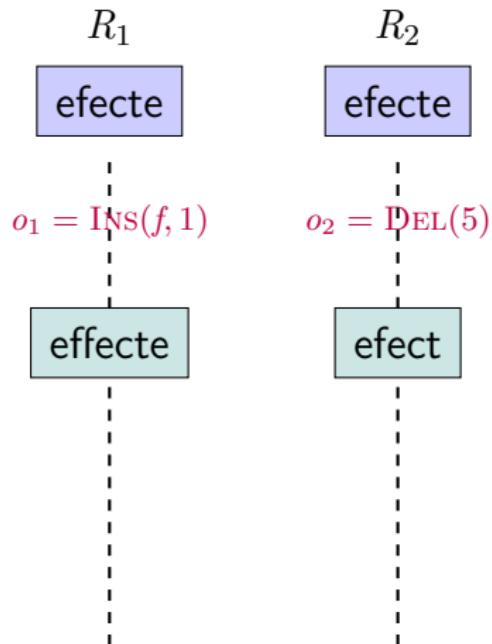


# 操作转换 (Operational Transformation; OT) [Ellis and Gibbs, 1989] 技术

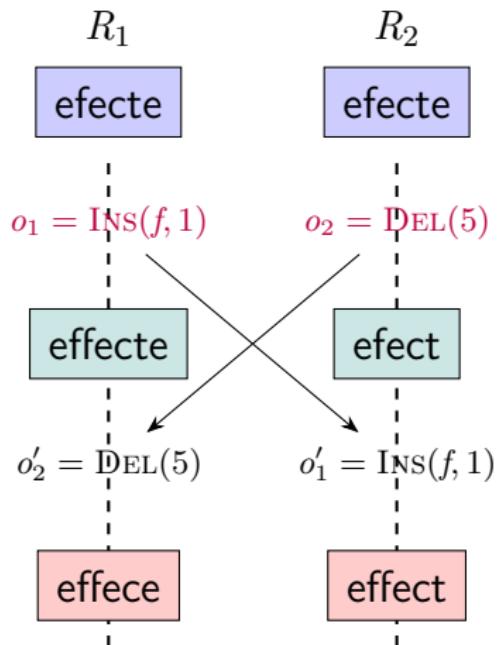
# 操作转换 (Operational Transformation; OT) [Ellis and Gibbs, 1989] 技术



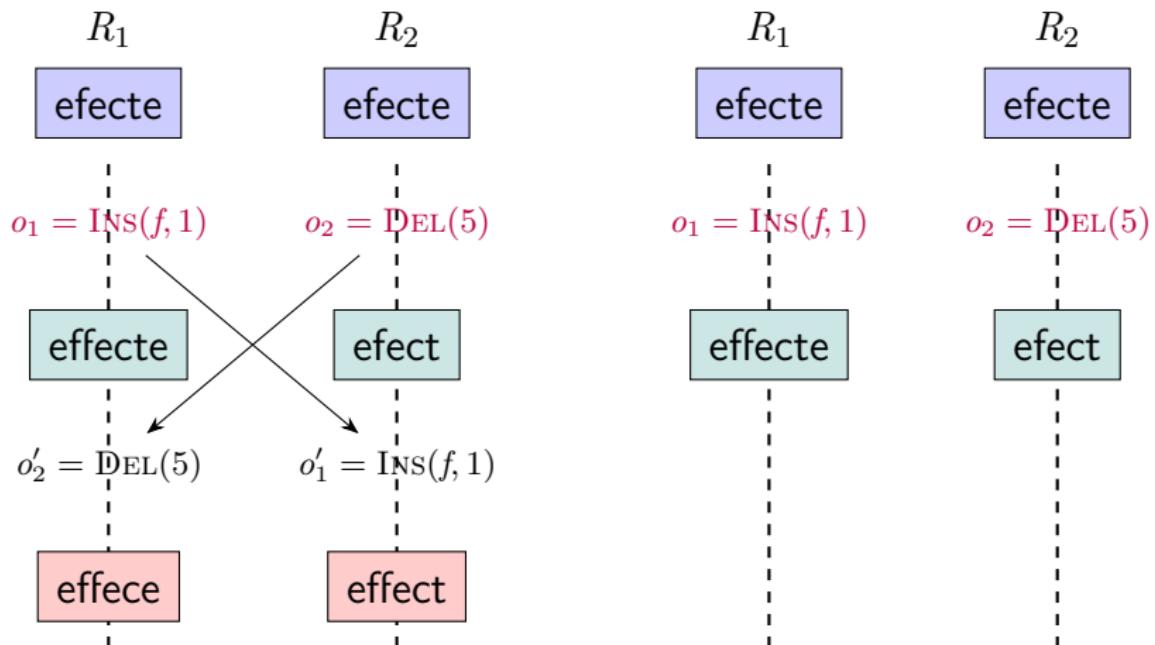
# 操作转换 (Operational Transformation; OT) [Ellis and Gibbs, 1989] 技术



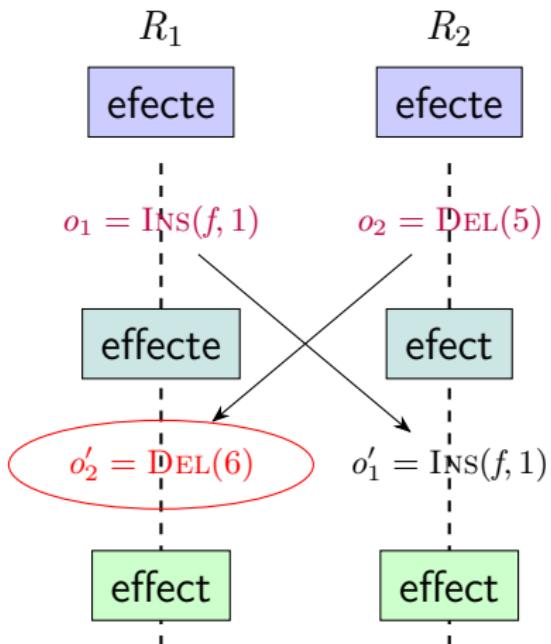
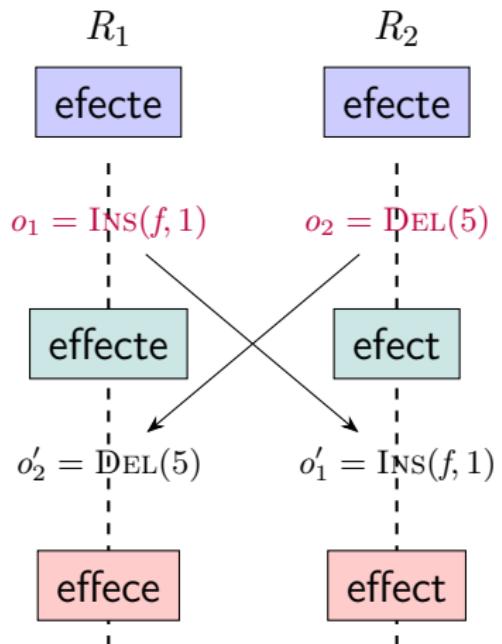
# 操作转换 (Operational Transformation; OT) [Ellis and Gibbs, 1989] 技术

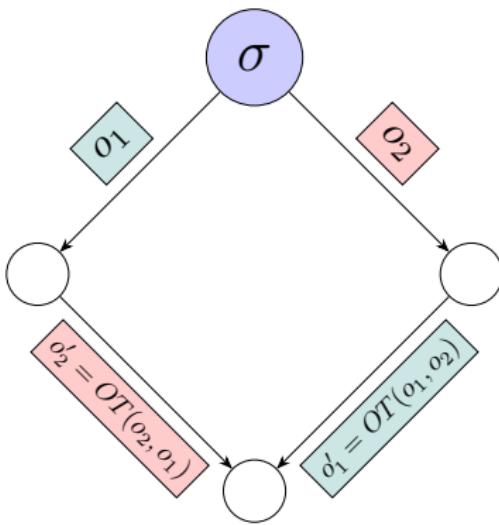


# 操作转换 (Operational Transformation; OT) [Ellis and Gibbs, 1989] 技术



# 操作转换 (Operational Transformation; OT) [Ellis and Gibbs, 1989] 技术





**交换律**  $\sigma; o_1; o_2' \equiv \sigma; o_2; o_1'$

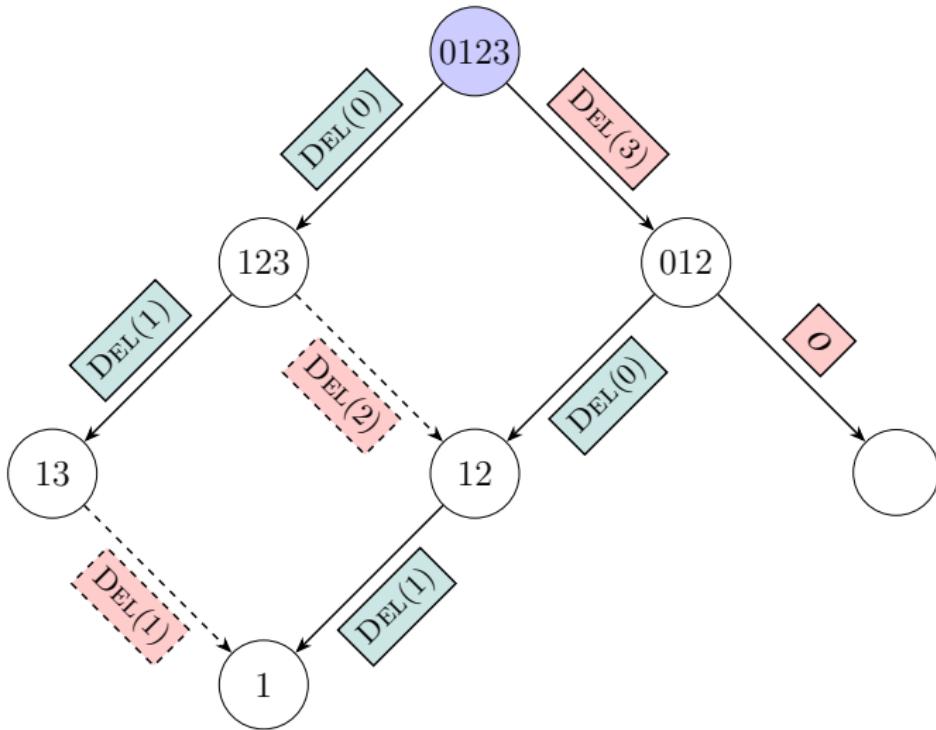
## 针对列表的操作转换函数 [Ellis and Gibbs, 1989]

$$OT\left( \text{INS}(a_1, p_1, pr_1), \text{INS}(a_2, p_2, pr_2) \right) = \begin{cases} \text{INS}(a_1, p_1, pr_1) & p_1 < p_2 \\ \text{INS}(a_1, p_1 + 1, pr_1) & p_1 > p_2 \\ \text{NOP} & p_1 = p_2 \wedge a_1 = a_2 \\ \text{INS}(a_1, p_1 + 1, pr_1) & p_1 = p_2 \wedge a_1 \neq a_2 \wedge pr_1 > pr_2 \\ \text{INS}(a_1, p_1, pr_1) & p_1 = p_2 \wedge a_1 \neq a_2 \wedge pr_1 \leq pr_2 \end{cases}$$

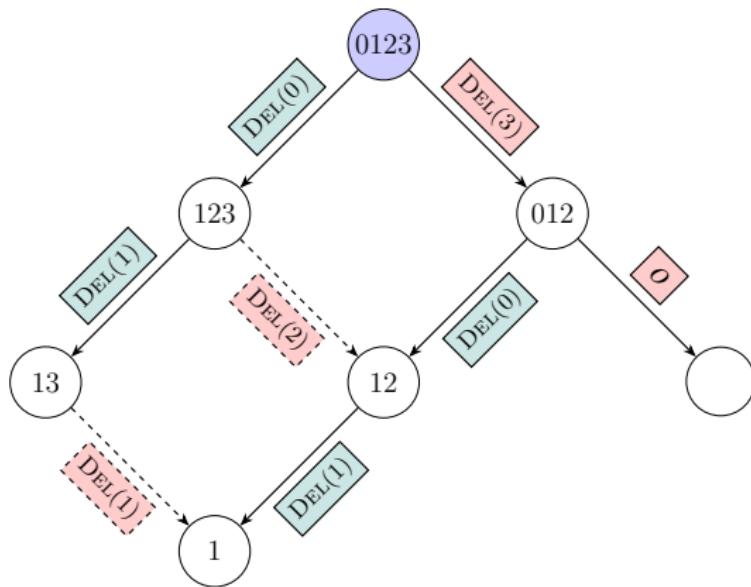
$$OT\left( \text{INS}(a_1, p_1, pr_1), \text{DEL}(\_, p_2, pr_2) \right) = \begin{cases} \text{INS}(a_1, p_1, pr_1) & p_1 \leq p_2 \\ \text{INS}(a_1, p_1 - 1, pr_1) & p_1 > p_2 \end{cases}$$

$$OT\left( \text{DEL}(\_, p_1, pr_1), \text{INS}(a_2, p_2, pr_2) \right) = \begin{cases} \text{DEL}(\_, p_1, pr_1) & p_1 < p_2 \\ \text{DEL}(\_, p_1 + 1, pr_1) & p_1 \geq p_2 \end{cases}$$

$$OT\left( \text{DEL}(\_, p_1, pr_1), \text{DEL}(\_, p_2, pr_2) \right) = \begin{cases} \text{DEL}(\_, p_1, pr_1) & p_1 < p_2 \\ \text{DEL}(\_, p_1 - 1, pr_1) & p_1 > p_2 \\ \text{NOP} & p_1 = p_2 \end{cases}$$

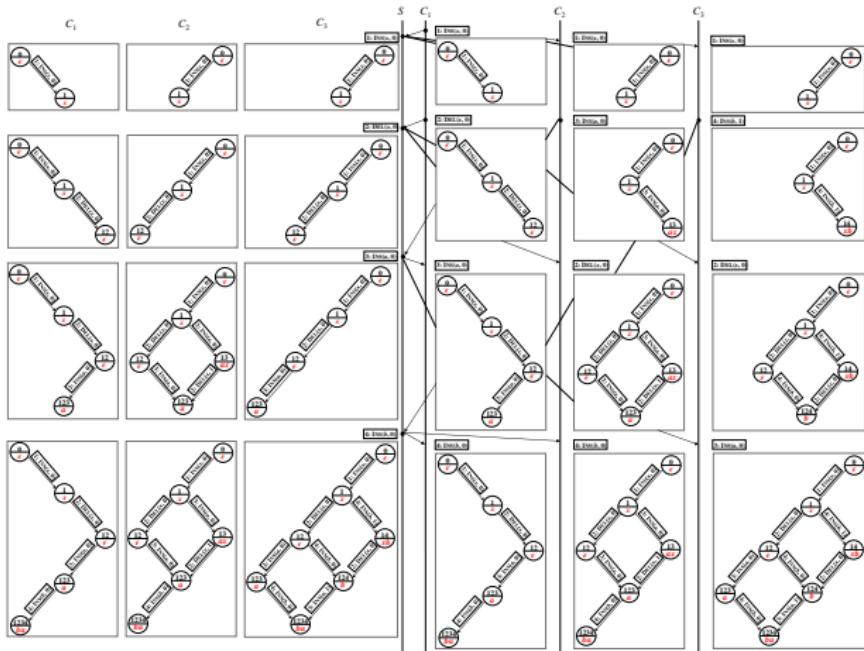


# 利用数据结构 2D 状态空间 [Xu, Sun, and Li, 2014] 控制何时以及如何执行“操作转换”



2D: LOCAL vs. GLOBAL

## 每个 Client 维护一个 2D 状态空间



Server 维护  $n$  个 2D 状态空间, 与  $n$  个 Clients 对应

$\mathcal{A}_{\text{weak}}$  所规定的全局性质



Jupiter 协议中，每个 replica 所维护的局部视图

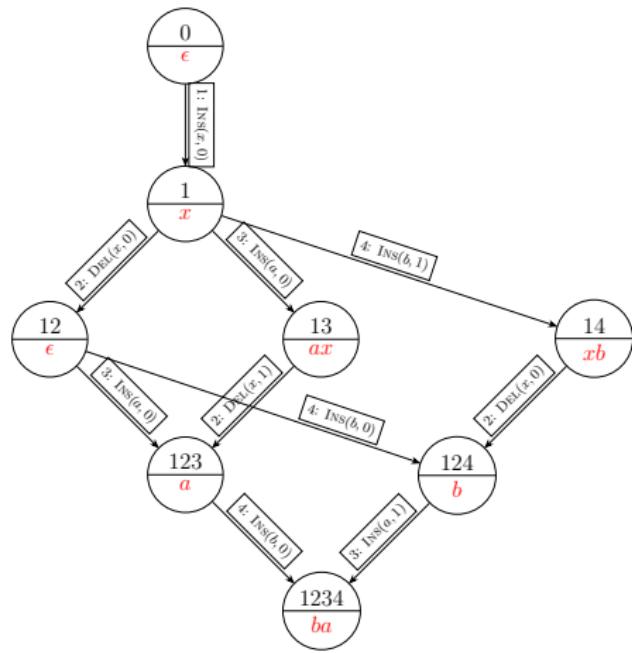
# CJupiter (Compact Jupiter)

# CJupiter (Compact Jupiter)

## Theorem (等价性)

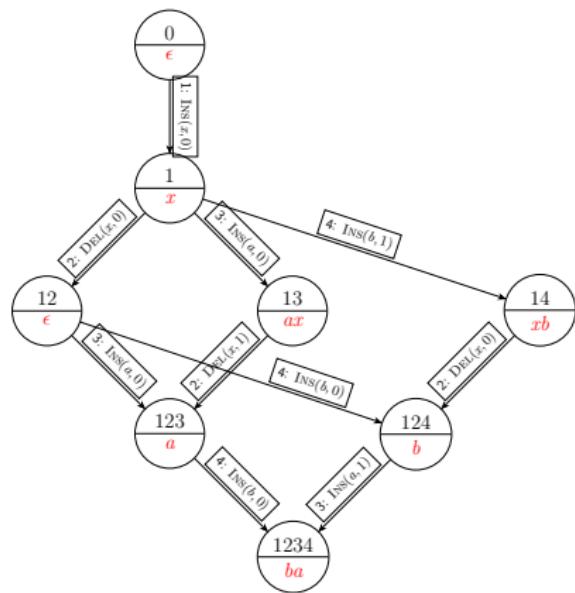
在相同的操作调度下, *CJupiter* 与 *Jupiter* 中的对应 *replica* 的行为 (状态序列) 是相同的。

CJupiter 为每个 replica 维护一个  $n$ -ary 有序状态空间



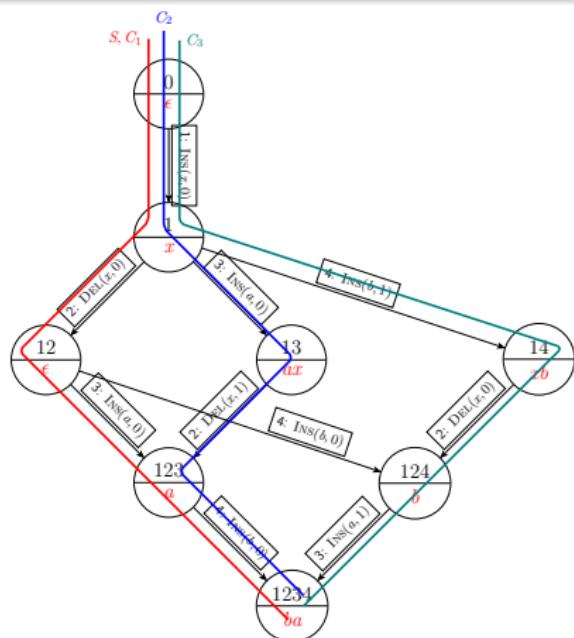
## 命题 (Compactness of CJupiter)

CJupiter 所维护的  $(n + 1)$  个  $n$ -ary 有序状态空间是相同的。



## 命题 (Compactness of C<sub>Jupiter</sub>)

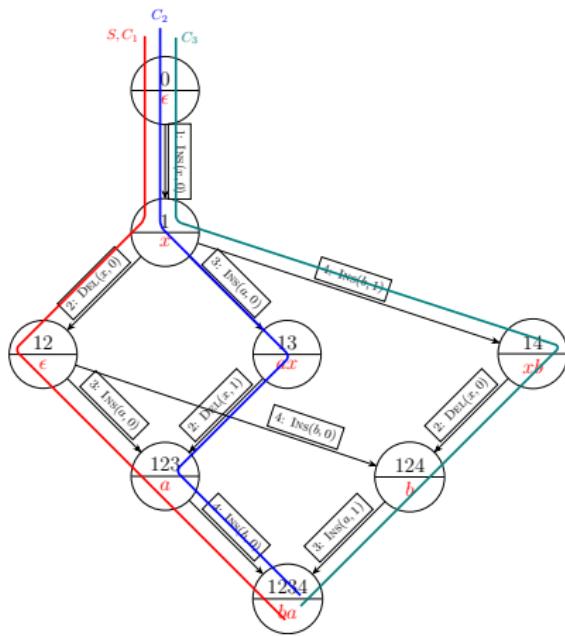
*CJupiter* 所维护的  $(n + 1)$  个  $n$ -ary 有序状态空间是相同的。



每个 replica 的行为对应于该状态空间中的一条路径

# CJupiter 满足 Weak List Specification

# 关注某个 $n$ -ary 有序状态空间, 三步骤证明状态对兼容性

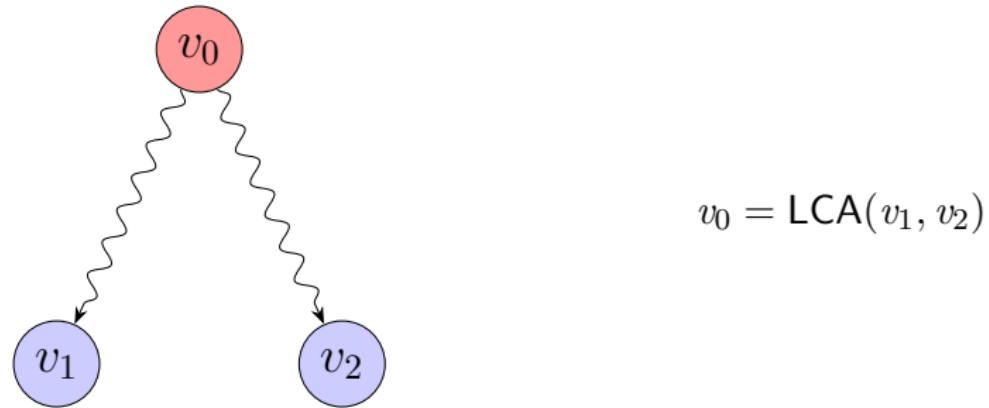


1

任取两个状态节点  $v_1$  和  $v_2$

引理 (LCA (Lowest Common Ancestor))

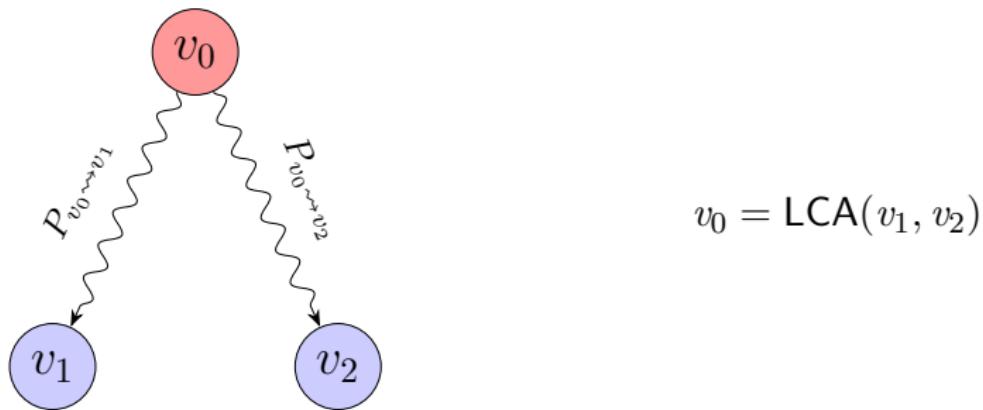
$n$ -ary 有序状态空间中的任意一对状态节点都有唯一的最近公共祖先。



2 考虑从  $v_0 = \text{LCA}(v_1, v_2)$  到  $v_1$  和  $v_2$  的两条路径

### 引理 (Disjoint Paths)

路径  $P_{v_0 \rightsquigarrow v_1}$  上包含的操作集  $O_{v_0 \rightsquigarrow v_1}$  与路径  $P_{v_0 \rightsquigarrow v_2}$  上包含的操作集  $O_{v_0 \rightsquigarrow v_2}$  不相交。

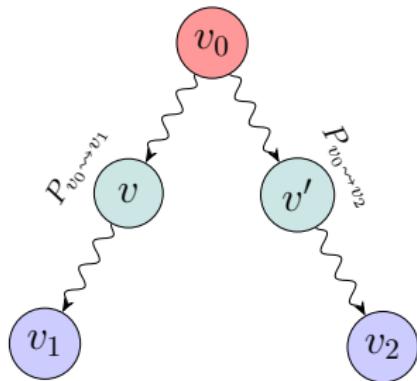


3

考虑两条路径上的状态

### 引理 (Compatible Paths)

$P_{v_0 \rightsquigarrow v_1}$  上的任一状态  $v$  与  $P_{v_0 \rightsquigarrow v_2}$  上的任一状态  $v'$  是兼容的。



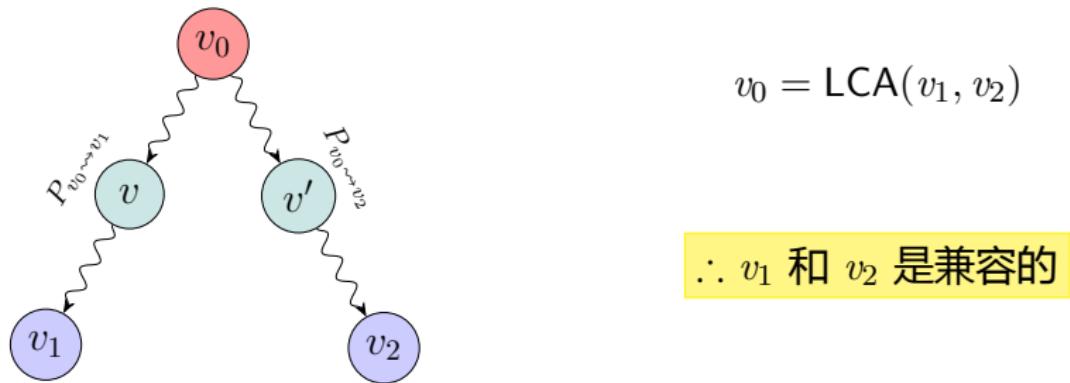
$$v_0 = \text{LCA}(v_1, v_2)$$

3

## 考虑两条路径上的状态

## 引理 (Compatible Paths)

$P_{v_0 \rightsquigarrow v_1}$  上的任一状态  $v$  与  $P_{v_0 \rightsquigarrow v_2}$  上的任一状态  $v'$  是兼容的。



个人感觉：基于 OT 思想的协议晦涩难懂



个人感觉：基于 OT 思想的协议晦涩难懂



- ▶ 协议多种多样
- ▶ 经常不加证明
- ▶ 证明是错误的

个人感觉：基于 OT 思想的协议晦涩难懂

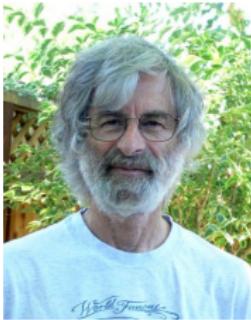


- ▶ 协议多种多样
- ▶ 经常不加证明
- ▶ 证明是错误的
- ▶ **勘误也是错的**

# Model Checking: 使用 TLA+



Refinement



- Attiya, Hagit et al. (2016). "Specification and complexity of collaborative text editing". In: *Proceedings of the 2016 ACM Symposium on Principles of Distributed Computing*. PODC '16. ACM, pp. 259–268.
- Burckhardt, Sebastian et al. (2014). "Replicated Data Types: Specification, Verification, Optimality". In: *Proceedings of the 41st ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*. POPL '14. San Diego, California, USA: ACM, pp. 271–284. ISBN: 978-1-4503-2544-8. DOI: 10.1145/2535838.2535848. URL: <http://doi.acm.org/10.1145/2535838.2535848>.
- Ellis, C. A. and S. J. Gibbs (1989). "Concurrency Control in Groupware Systems". In: *Proceedings of the 1989 ACM SIGMOD International Conference on Management of Data*. SIGMOD '89. ACM, pp. 399–407.
- Herlihy, Maurice P. and Jeannette M. Wing (1990). "Linearizability: A Correctness Condition for Concurrent Objects". In: *ACM Trans. Program. Lang. Syst.* 12.3, pp. 463–492. ISSN: 0164-0925. DOI: 10.1145/78969.78972. URL: <http://doi.acm.org/10.1145/78969.78972>.
- Liskov, Barbara and Stephen Zilles (1974). "Programming with Abstract Data Types". In: *Proceedings of the ACM SIGPLAN Symposium on Very High Level Languages*. Santa Monica, California, USA: ACM, pp. 50–59. DOI: 10.1145/800233.807045. URL: <http://doi.acm.org/10.1145/800233.807045>.
- Nichols, David A. et al. (1995). "High-latency, Low-bandwidth Windowing in the

Thank  
You!

