

RISC-V 内存一致性模型的同地址顺序一致性定理证明^{*}

徐学政, 杨德亨, 王璐, 王涛, 黄安文, 李琼

(军事科学院 国防科技创新研究院, 北京 100071)

通信作者: 李琼, E-mail: qiong.li@nanhulab.ac.cn



摘要: 内存一致性模型定义了并行程序在多核系统中的访存序约束, 是软硬件共同遵守的架构规范。同地址顺序一致性是内存一致性模型的经典公理之一, 它规定了多核系统中对于相同地址的所有访存操作遵循顺序一致性, 被广泛应用于 X86/TSO、Power、ARM 等经典架构的内存一致性模型中, 在芯片内存一致性验证及系统软件和并行程序开发中发挥着重要作用。RISC-V 作为开源的架构规范, 其内存模型由全局访存序、保留程序序以及 3 条公理 (加载值公理、原子性公理和进度保证公理) 定义, 并未将同地址顺序一致性直接作为公理, 这给已有的内存模型验证工具和系统软件开发带来了挑战。面向 RISC-V 内存模型, 基于已定义的公理和规则, 将同地址顺序一致性作为定理, 通过将任意同地址访存序列的构建抽象为确定有限状态自动机进行归纳证明。该研究是对 RISC-V 内存一致性相关形式化方法的一个理论补充。

关键词: RISC-V; 内存一致性; 定理证明

中图法分类号: TP302

中文引用格式: 徐学政, 杨德亨, 王璐, 王涛, 黄安文, 李琼. RISC-V 内存一致性模型的同地址顺序一致性定理证明. 软件学报, 2025, 36(9): 3919–3936. <http://www.jos.org.cn/1000-9825/7292.htm>

英文引用格式: Xu XZ, Yang DH, Wang L, Wang T, Huang AW, Li Q. Sequential Consistency Per Location Theorem Proving in RISC-V Memory Consistency Model. Ruan Jian Xue Bao/Journal of Software, 2025, 36(9): 3919–3936 (in Chinese). <http://www.jos.org.cn/1000-9825/7292.htm>

Sequential Consistency Per Location Theorem Proving in RISC-V Memory Consistency Model

XU Xue-Zheng, YANG De-Heng, WANG Lu, WANG Tao, HUANG An-Wen, LI Qiong

(National Innovation Institute of Defense Technology, Academy of Military Sciences, Beijing 100071, China)

Abstract: The memory consistency model defines constraints on memory access orders for parallel programs in multi-core systems and is an important architectural specification jointly followed by software and hardware. Sequential consistency (SC) per location is a classic axiom of memory consistency models, which specifies that all memory access operations with the same address in a multi-core system follow SC. Meanwhile, it has been widely employed in the memory consistency models of classic architectures such as X86/TSO, Power, and ARM, and plays an important role in chip memory consistency verification, system software, and parallel program development. RISV is an open-source architectural specification, and its memory model is defined by global memory orders, preserved program orders, and three axioms (the load value axiom, atomicity axiom, and progress axiom). Additionally, it does not directly include SC per location as an axiom, which poses challenges to existing memory model verification tools and system software development. This study formalizes the SC per location as a theorem based on the defined axioms and rules in the RISC-V memory model. The proof process abstracts the construction of memory access sequences with the same arbitrary address into deterministic finite automata for inductive proof. This study is a theoretical supplement to the formal methods of RISC-V memory consistency.

Key words: RISC-V; memory consistency; theorem proving

^{*} 基金项目: 国家自然科学基金 (62102439, 62402515)

收稿时间: 2023-10-17; 修改时间: 2024-02-21, 2024-08-15; 采用时间: 2024-09-19; jos 在线出版时间: 2024-12-10

CNKI 网络首发时间: 2025-05-15

内存一致性模型^[1,2], 简称内存模型, 定义了并行程序在多核系统中的访存序约束, 是软硬件共同遵守的架构规范, 经典的内存一致性模型包括 SC (sequential consistency)^[1]、X86/TSO^[3,4]、Power^[5]、ARM^[6]等. 作为沟通多核系统软硬件设计的规范, 学术界和工业界使用形式化方法^[5,7]对内存模型进行描述, 以消除自然语言描述带来的歧义. 公理模型^[7,8]是最为经典的内存模型形式化描述方法之一, 其使用数学语言定义内存模型, 进而辅助软硬件的形式化验证^[9].

同地址顺序一致性 (sequential consistency per location, SCPL)^[7,10]是内存一致性公理模型的经典公理之一, 它规定了多核系统中对于相同地址的所有访存操作遵循顺序一致性^[1], X86/TSO、Power、ARM 等经典架构的内存一致性模型均支持该公理^[7,8]. 此外, Linux Kernel 的内存模型^[11]中也有类似的公理 (SC per variable). SCPL 直观地描述了对于同地址访存操作的顺序约束, 简化了形式化方法在内存模型中的应用, 在芯片的内存一致性验证, 系统软件和并行程序开发和验证等领域^[7,12]发挥着重要作用.

RISC-V 是由加州大学伯克利分校提出的开源指令集规范^[13], 具有精简、开放、模块化、高可定制的特点, 近年来受到了工业界和教育界的广泛关注, 已在物联网、云计算、人工智能、高性能计算等领域广泛应用^[14]. RISC-V 内存模型 (RISC-V weak memory ordering, RVWMO) 是一种基于 Release Consistency (RC)^[15]的宽松内存模型, 旨在为硬件设计提供灵活性的同时保证软件的易开发性. 在 RISC-V 架构规范^[13]中, RVWMO 建立在访存原语 (primitive memory operation) 基础上, 由全局访存序 (global memory order, *gmo*)、保留程序序 (preserved program order, *ppo*) 以及模型公理构成 (见图 1). 其中, 模型公理包括加载值公理 (load value axiom)、原子性公理和进度保证公理.

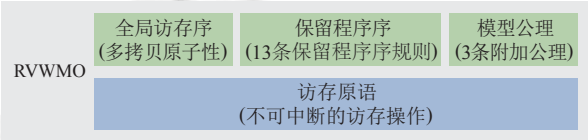


图 1 RVWMO 在 RISC-V 架构规范中的定义构成

RVWMO 并未将 SCPL 直接作为公理加入模型规范, 这对已有的内存模型验证工具和系统软件的开发带来了挑战. 例如, RISC-V 架构规范^[13]的附录 B.2 给出两种 RVWMO 的公理模型, 分别是基于 RVWMO 定义 (全局访存序、13 条保留程序序规则和 3 条附加公理) 并不包含 SCPL 的公理模型以及另一种包含 SCPL 的公理模型. 规范中指出, 后者是等价且更为高效的模型. 我们使用内存模型验证工具 Herd^[7]分别加载两个模型, 对 RISC-V 的内存模型测试套件^[16]进行测试, 以验证架构规范中的结论. 实验表明, 后者对前者有超过 400% 的加速效果 (见图 2) 且内存开销更小.

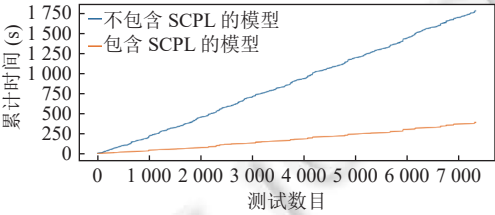


图 2 Herd 基于两种 RVWMO 公理模型的验证效率对比

对于软件开发者, SCPL 是更为直观的公理, 将大大降低软件开发和验证的难度. 图 3 给出了 RVWMO 禁止发生的访存序模式. 其中 x 为共享变量, r_1 和 r_2 为寄存器, 操作 a 和 b 为对 x 的读操作, c 和 d 为对 x 的写操作. 通过运行结果可以看出, 读操作 a 和 b 的值分别来自写操作 d 和 c , 即两次读操作先读到新值, 再读到旧值, 这是反直觉的. 该模式违背了 SCPL, 可直接判定其非法性. 如果不利用 SCPL 公理, 形式化验证工具和软件开发者需要利用全局访存序、加载值公理以及保留程序序进行综合判断, 难度大大增加. 图 3 非法性的证明见命题 4.

本文面向 RVWMO, 基于指令集规范^[13]中定义的公理和规则, 将 SCPL 作为定理进行了证明. 证明过程将任

意同地址访存序列的构建抽象为确定有限状态自动机, 每个状态对应一种访存序列的模式, 通过对各个模式满足 SCPL 以及模式转换正确性的证明, 从而实现对任意访存序列满足 SCPL 的证明. 证明过程使用 Coq 进行了验证. 本研究是对 RVWMO 相关形式化方法的一个理论补充.

初始条件: $x=0$	
线程 0	线程 1
$a: r_1=x;$ $b: r_2=x;$	$c: x=1;$ $d: x=2;$
最终结果: $x=2; r_1=2; r_2=1$	

图3 被 RVWMO 禁止的访存序模式 (x 为共享变量, r_1 和 r_2 为寄存器)

本文涉及的 Coq 脚本与实验数据已开源 (https://gitee.com/twohaha/scpl_proof).

本文第 1 节详细介绍 RVWMO 及 Litmus 测试. 第 2 节介绍 RVWMO 公理模型的相关形式化定义. 第 3 节介绍 SCPL 及其形式化描述. 第 4 节介绍基于确定有限状态自动机的定理证明过程. 第 5 节应用 SCPL 进行案例分析. 最后总结全文.

1 背景知识

1.1 RVWMO 概述

RVWMO 是一种宽松的内存一致性模型, 属于 RC 模型^[15]的变种. RVWMO 规定并程序序执行生成的所有访存原语遵循全局访存序, 顺序与保留程序序一致, 且符合加载值公理, 原子性公理和进度保证公理. 其中, 访存原语指的是不可中断的读写操作, 是访存操作的基本单元, 在后文中统称为访存事件或操作. 全局访存序指的是所有硬件线程的访存事件遵循一个共同的观测顺序, 这是由 RVWMO 满足多拷贝原子性^[17]决定的. 相比之下, Power 内存模型^[5]不具备多拷贝原子性, 各个线程无法形成统一的全局访存序. 程序序 (program order, po) 指的是单个线程执行的指令序列构成的顺序, 而保留程序序指的是程序序中必须维护保留的部分, 即不可乱序的部分. 例如, 具有依赖关系的两条指令在执行时不可乱序. RVWMO 共定义了 13 条保留程序序规则. 这里给出前两条规则的非形式化定义 (形式化定义见第 2.1 节). (1) 具有程序序的两个对同地址的访存操作, 后者是写操作. (2) 具有程序序的两个对同地址的读操作, 二者的值来自不同的写操作, 且程序序在二者之间不存在对同地址的写操作.

RVWMO 定义了 3 条公理对模型进行补充. 加载值公理规定, 对于任意读操作 r , 其目标地址 x 的值来自在以下所有操作中位于全局访存序末位的操作: (1) 全局访存序位于 r 之前的所有对 x 的写操作; (2) 程序序位于 r 之前的所有对 x 的写操作. 图 4 给出了加载值公理的示意图, 其中所有操作的目标地址均相同, r 和 w 分别表示读操作和写操作. 其中, r 的值可能来自全局访存序在其之前的写操作 (w_1 和 w_2) 以及程序序在其之前的写操作 (w_3), 将以上 3 个写操作放在全局访存序中比较, r 的值最终来自排在末位的 w_3 . 值得注意的是, w_3 的全局访存序在 r 之后, 这是因为 RVWMO 允许线程内的写和读乱序. X86/TSO 模型同样放松了该约束^[2,3], 主要由微架构中的写缓冲机制导致的. 原子性公理规定了成对出现的读保留 (load-reserved) 和条件写 (store-conditional) 的相关访存约束. 进度保证公理规定了访存操作在有限时间内全局可见, 这要求我们在判定访存序合法性时考虑每一个访存操作. 本文的证明主要利用了加载值公理.

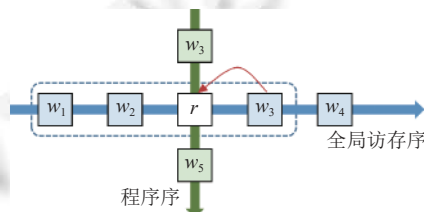


图4 加载值公理的示意图 (r 表示读操作, w 表示写操作)

对 RVWMO 的形式化建模方法主要分为操作模型 (operational model) 和公理模型 (axiomatic model)^[2]. 其中, 操作模型是通过微架构 (乱序发射、写缓冲、流水线、存储层次、Cache 一致性协议等) 的抽象实现, 模拟出访存行为从而得到合法的机器状态, 代表性工作是使用 Sail 语言^[17]的 RMEM^[18]. 公理模型直接以数学公理的形式约束访存行为之间的顺序关系 (例如允许 Store-Load 乱序), 从而计算出合法的机器状态, 代表性工作有基于 Alloy^[19]和 Herd^[7]的建模. 基于操作模型和公理模型, 可自动化地推理出并程序的合法运行结果. 相比操作模型, 公理模型并不对微架构建模, 方法更抽象, 描述更简单. 但操作模型更贴近微架构实现, 可根据微架构设计直接建模, 有利于追踪微架构设计中的缺陷. 本文是基于公理模型在规范层面对 RVWMO 开展研究, 并不涉及微架构的实现.

1.2 Litmus 测试

本文将使用 Litmus 测试^[12]对内存模型进行示例解释. Litmus 测试是一小段并程序, 通常是某种访存序模式的抽象, 用于测试或演示内存模型的某个性质. Litmus 测试通常包含各个线程执行的静态代码以及程序的初始状态和最终状态. 图 3 中的例子本质上就是 Litmus 测试.

图 5(a) 和图 5(b) 分别给出了图 3 对应的 Litmus 标准语法描述和事件关系图示. 其中, 图 5(b) 是根据程序的运行结果绘制的. 每个节点代表一个访存事件, 例如, $c: Wx=1$ 表示事件 c 向地址 x 写入 1, $a: Rx=2$ 表示事件 a 从地址 x 读取 2. 每条边代表事件之间的关系 (详细定义见第 2.1 节), 例如 co 表示写事件的顺序, 由于 x 对应的最终值为 2, 所以事件 d 发生在事件 c 之后; rf 为读取关系, 由于事件 a 读取的值为 2, 因此 a 的值读取自事件 d . 基于公理模型的形式化验证工具 (如 Herd) 会基于模型公理对访存事件及其关系进行合法性检查, 进而判定程序的执行结果是否满足对应的内存模型. 对于该例非法性的证明见命题 4.

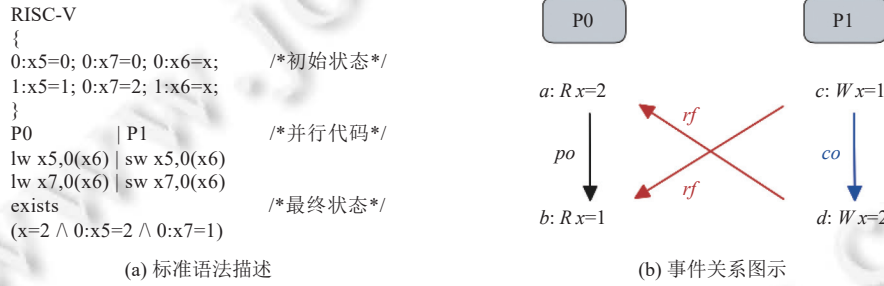


图 5 Litmus 测试标准语法描述与事件关系图示

2 RVWMO 公理模型

2.1 RVWMO 基础定义

本节定义了内存一致性公理模型^[7,10]的基本关系及 RVWMO 的相关公理. 公理模型是内存模型的一种形式化描述方法, 通过将内存模型表示成公理, 可对某个访存序的合法性进行判定. RISC-V 架构规范^[13]的附录 A 中给出了 RVWMO 的几种公理模型, 本节将介绍其中涉及 SCPL 证明的部分.

公理模型研究的基本元素是访存事件. 受分支选择以及多线程的执行顺序影响, 不同的程序执行可能产生不同的访存事件. 公理模型是通过检查访存事件构成的二元关系是否符合所有公理从而对访存序的合法性进行判定的. 在本文中, 使用符号 $\equiv$ 表示等价, 使用符号 $\triangleq$ 解释定义.

定义 1 (访存事件). 访存事件由程序的执行产生, 定义为五元组 $(id, proc, poidx, addr, kind)$. 其中, id 为该事件在全局访存序中的索引, 不同的事件 id 不同; $proc$ 为对应的线程号; $poidx$ 为程序索引, 同一线程的事件 $poidx$ 不同; $addr$ 为访存地址; $kind$ 为事件类型, 仅考虑读 (R) 和写 (W). 将访存事件 a 的 id 域表示为 $id(a)$, $proc$ 域表示为 $proc(a)$, 以此类推; 将读 (写) 事件 a 表示为 $a \in R$ ($a \in W$), 即 $kind(a) = R$ ($kind(a) = W$).

定义 2. int (internal) 是一个二元关系集合, 表示两个事件位于同一线程. 若事件 a 和 b 满足 int 关系, 记为

$(a,b) \in \text{int}$ 或 $\text{int}(a,b)$, 满足: $\text{int}(a,b) \triangleq \text{proc}(a) = \text{proc}(b)$.

定义 3. ext (external) 是一个二元关系集合, 表示两个事件位于不同线程. 若事件 a 和 b 满足 ext 关系, 记为 $(a,b) \in \text{ext}$ 或 $\text{ext}(a,b)$, 满足: $\text{ext}(a,b) \triangleq \text{proc}(a) \neq \text{proc}(b)$.

由 int 和 ext 的定义可得: $\forall a,b \in E, \text{int}(a,b) \vee \text{ext}(a,b)$.

定义 4. loc (same location) 是一个二元关系集合, 表示两个事件的访存地址相同. 若事件 a 和 b 满足 loc 关系, 可知 $\text{addr}(a) = \text{addr}(b)$, 记为 $(a,b) \in \text{loc}$ 或 $\text{loc}(a,b)$. 显然, loc 具有传递性和对称性.

定义 5. gmo 是一个二元关系集合, 表示两个事件在全局访存序中的顺序关系. 若事件 a 和 b 满足 gmo 关系, 记为 $(a,b) \in \text{gmo}$ 或 $\text{gmo}(a,b)$, 满足: $\text{gmo}(a,b) \triangleq \text{id}(a) < \text{id}(b)$. 显然, gmo 是非对称的. 所有访存事件在 gmo 中构成全序, 即: $\forall a,b \in E, a \neq b \rightarrow \text{gmo}(a,b) \vee \text{gmo}(b,a)$.

定义 6. po 是一个二元关系集合, 表示同线程内两个事件在程序执行中的先后顺序. 若事件 a 和 b 满足 po 关系, 记为 $(a,b) \in \text{po}$ 或 $\text{po}(a,b)$, 满足: $\text{po}(a,b) \triangleq \text{int}(a,b) \wedge \text{poidx}(a) < \text{poidx}(b)$. 显然, po 是传递的、非对称的. 同线程的访存事件在 po 上构成全序, 即: $\forall a,b \in E, a \neq b \wedge \text{int}(a,b) \rightarrow \text{po}(a,b) \vee \text{po}(b,a)$.

定义 7. rf (read from) 是一个二元关系集合, 表示读事件的值来自某个写事件. 若事件 a 和 b 满足 rf 关系, 可知: $\text{loc}(a,b) \wedge a \in W \wedge b \in R$, 记为 $(a,b) \in \text{rf}$ 或 $\text{rf}(a,b)$. 注意, 满足 $\text{loc}(a,b) \wedge a \in W \wedge b \in R$ 的事件 a 和 b 并不一定满足 rf 关系.

定义 8. co (coherence order) 是一个二元关系集合, 表示对同地址的两个写事件的顺序关系. 若事件 a 和 b 满足 co 关系, 记为 $(a,b) \in \text{co}$ 或 $\text{co}(a,b)$, 满足: $\text{co}(a,b) \triangleq \text{loc}(a,b) \wedge a \in W \wedge b \in W \wedge \text{gmo}(a,b)$. 显然, co 是传递的、非对称的. 由 gmo 的定义可知, 所有同地址的写事件在 co 上构成全序, 即: $\forall a,b \in W, a \neq b \wedge \text{loc}(a,b) \rightarrow \text{co}(a,b) \vee \text{co}(b,a)$.

定义 9. fr (from read) 是一个二元关系集合, 属于 rf 和 co 的派生关系. 若事件 a 和 b 满足 fr 关系, 记为 $(a,b) \in \text{fr}$ 或 $\text{fr}(a,b)$, 满足: $\text{fr}(a,b) \triangleq a \in R \wedge b \in W \wedge (\exists x \in E, \text{rf}(x,a) \wedge \text{co}(x,b))$. 图 6 给出了 fr 关系的示意图.

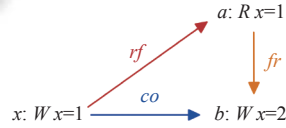


图 6 派生关系 fr 的示意图

定义 10. poloc (program order with the same location) 是一个二元关系集合, 属于 po 和 loc 的派生关系. 若事件 a 和 b 满足 poloc 关系, 记为 $(a,b) \in \text{poloc}$ 或 $\text{poloc}(a,b)$, 满足: $\text{poloc}(a,b) \triangleq \text{po}(a,b) \wedge \text{loc}(a,b)$. 显然, poloc 关系是传递的和非对称的.

定义 11. ppo1 是一个二元关系集合, 表示符合保留程序规则 1 的事件关系. 若事件 a 和 b 满足 ppo1 关系, 记为 $(a,b) \in \text{ppo1}$ 或 $\text{ppo1}(a,b)$, 满足: $\text{ppo1}(a,b) \triangleq \text{poloc}(a,b) \wedge b \in W$.

定义 12. ppo2 是一个二元关系集合, 表示符合保留程序规则 2 的事件关系. 若事件 a 和 b 满足 ppo2 关系, 记为 $(a,b) \in \text{ppo2}$ 或 $\text{ppo2}(a,b)$, 满足:

$$\text{ppo2}(a,b) \triangleq \text{poloc}(a,b) \wedge a \in R \wedge b \in R \wedge (\nexists x \in W, \text{poloc}(a,x) \wedge \text{poloc}(x,b)) \wedge (\nexists x \in W, \text{rf}(x,a) \wedge \text{rf}(x,b)).$$

定义 13. ppo 是一个二元关系集合, 表示保留程序序, 是程序序中不允许乱序的部分. RVWMO 定义了 13 条保留程序序规则, 本文使用前两条, 记为 ppo1 和 ppo2 . 其余 11 条规则与本文的定理证明无关. 若事件 a 和 b 满足 ppo 关系, 记为 $(a,b) \in \text{ppo}$ 或 $\text{ppo}(a,b)$, 满足: $\text{ppo}(a,b) \triangleq \text{ppo1}(a,b) \vee \text{ppo2}(a,b)$.

定义 14. 部分关系根据是否位于同一线程划分为两个子关系:

$$\begin{aligned} \text{rfi}(a,b) &\triangleq \text{rf}(a,b) \wedge \text{int}(a,b), & \text{rfe}(a,b) &\triangleq \text{rf}(a,b) \wedge \text{ext}(a,b); \\ \text{coi}(a,b) &\triangleq \text{co}(a,b) \wedge \text{int}(a,b), & \text{coe}(a,b) &\triangleq \text{co}(a,b) \wedge \text{ext}(a,b); \\ \text{fri}(a,b) &\triangleq \text{fr}(a,b) \wedge \text{int}(a,b), & \text{fre}(a,b) &\triangleq \text{fr}(a,b) \wedge \text{ext}(a,b). \end{aligned}$$

进一步可得:

$$rf(a,b) \equiv rfi(a,b) \vee rfe(a,b), co(a,b) \equiv coi(a,b) \vee coe(a,b), fr(a,b) \equiv fri(a,b) \vee fre(a,b).$$

定义 15. 二元关系 r 的传递闭包记为 r^+ , 满足:

$$\forall a,b \in E, (a,b) \in r \rightarrow (a,b) \in r^+ \wedge \forall a,b,c \in E, (a,b) \in r^+ \wedge (b,c) \in r^+ \rightarrow (a,c) \in r^+.$$

定义 16. 二元关系 r 无环记为 $acyclic(r)$, 满足: $acyclic(r) \triangleq \nexists x \in E, (x,x) \in r^+$.

定义 17. 二元关系 $perloc$ 满足: $perloc \triangleq poloc \cup rf \cup fr \cup co$. 若在长度为 n 的访存事件序列 $s = e_0, e_1, e_2, \dots, e_{n-1}$ 中, 相邻事件满足 $perloc$ 关系, 则记为 $perloc(s)$. 例如, 当 $s = a, b, c$ 时, $perloc(s) \triangleq perloc(a,b) \wedge perloc(b,c)$. 为方便叙述, 我们在不引起歧义的上下文中, 也将 $perloc(s)$ 直接写为 $perloc(a,b,c)$.

定义 18 (执行实例). 一段 (并行) 程序的执行实例定义为四元组 (E, po, rf, co) . 其中, E 为访存事件集合 (见定义 1), po 为程序序关系集合 (见定义 6), rf 为读取关系集合 (见定义 7), co 为写事件的顺序关系集合 (见定义 8). 注意, 执行实例对应的四元组是由程序执行确定的, 无须推导和证明 (另见第 1.2 节).

2.2 RVWMO 公理

本节定义了 4 条用于证明 SCPL 的公理, 均总结自 RISC-V 架构手册第 14.1 节^[13].

公理 1 (read from). 每一个读事件的值必然来自某个写事件:

$$\forall a \in R, \exists b \in W, rf(b,a).$$

若在全局访存序中, 某个读事件先于任何同地址的写事件, 表明该读事件的值来自内存初始值. 在模型检验时, 此时会额外创建一个对应该内存初始值的写事件^[7,10], 该公理依然适用.

公理 2 (单拷贝原子性). 一个读事件的值仅来自一个写事件:

$$\forall a,b \in E, rf(a,b) \rightarrow \nexists x \in E, x \neq a \wedge rf(x,b).$$

显然, 我们可以得到 $\forall a,b,c \in E, rf(a,c) \wedge rf(b,c) \rightarrow a=b$.

公理 3 (保留程序序). 两个事件的全局访存序与 ppo 关系顺序一致: $\forall a,b \in E, ppo(a,b) \rightarrow gmo(a,b)$.

公理 4 (加载值). 将第 1.1 节描述的加载值公理定义如下:

$$\forall a,b \in E, rf(a,b) \rightarrow (po(a,b) \vee gmo(a,b)) \wedge (\nexists x \in W, loc(x,b) \wedge gmo(a,x) \wedge (po(x,b) \vee gmo(x,b))).$$

3 同地址顺序一致性定理

同地址顺序一致性^[7,10]描述的是: 对于某个内存模型下的任意执行实例产生的访存事件集合, 其中存在的 rf , fr 和 co 关系与 $poloc$ 的顺序是一致的, 即不能成环, 表示为 $acyclic(poloc \cup rf \cup fr \cup co)$ 或 $acyclic(perloc)$ (见定义 17). SCPL 的另一个等价表述为: 对于任意的以 x 开始, 以 y 结束的访存事件序列, 若相邻的事件之间均存在 $perloc$ 关系, 则 $perloc(y,x)$ 必然不成立, 即不能成环 (见图 7). 本文中, 将 SCPL 作为主定理进行证明.

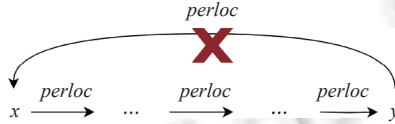


图 7 基于访存事件序列的 SCPL 定理描述示意

定义 19 (SCPL). 假设 s 为某个长度为 n ($n > 0$) 的访存事件序列, 即 $s = e_0, e_1, e_2, \dots, e_{n-1}$, 若序列满足 SCPL, 记为 $SCPL(s)$ 且 $SCPL(s) \triangleq perloc(s) \rightarrow \neg perloc(e_{n-1}, e_0)$.

主定理 (RVWMO 满足 SCPL). 假设 E 为在 RVWMO 下任意执行实例构成的访存事件集合, 由 E 包含的访存事件构成的任意序列 s , 满足 $SCPL(s)$. 第 4 节给出了定理的证明思路和步骤.

图 8 给出了违背 SCPL 的 5 种访存模式的 Litmus 示意图^[6,7]. 5 种模式中所有访存操作的目标地址均为 x . 例如, $coWW$ 模式描述的是同一线程存在先后两个写操作 a 和 b , x 的最终值来自前者 a 而不是 b . $coRW1$ 模式描述的是同一线程中的读操作 a 的值来自其之后的写操作 b . 这些例子都是反直觉的. 直观来看, 5 种模式中存在的 $perloc$ 关系均成环, 利用 SCPL 定理可以轻易证明它们的非法性. 关于 SCPL 定理的应用另见第 5 节.

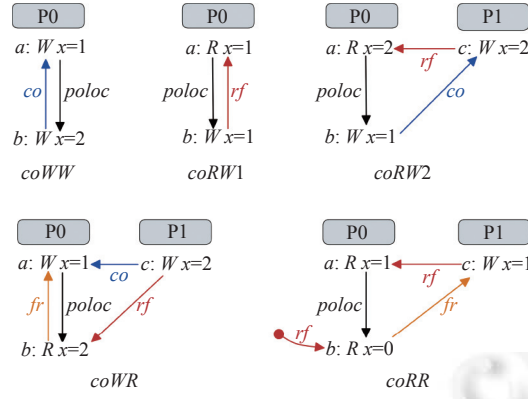


图8 SCPL 禁止的 5 种访存模式的 Litmus 示意图

4 基于确定有限状态自动机的 SCPL 定理证明

4.1 基于确定有限状态自动机的定理证明方法

为了证明 RVWMO 下的任意访存序列满足 SCPL, 本文引入确定有限状态自动机 (deterministic finite automata, DFA)^[20] 进行归纳证明. 该自动机可接收任意访存序列, 从而将问题分解为多个子问题逐一证明. 具体证明思路如下.

- (1) 将证明目标由任意访存事件的序列满足 SCPL 转为任意序列模式满足 SCPL.
- (2) 引入确定有限状态自动机 M , 任意序列模式可被 M 接受.
- (3) 证明自动机 M 中所有状态对应的序列模式均满足 SCPL.
- (4) 证明某个序列模式满足 SCPL, 可由其在自动机 M 中对应的序列模式满足 SCPL 间接证明.

第 4.1.1–4.1.4 节将分别解释以上 4 个步骤.

4.1.1 序列模式

定义 20 (序列模式). 将访存事件序列中每个事件对应的类型构成的序列定义为该序列的模式. 对于任意长度为 n ($n > 0$) 的访存事件序列 s ($s = e_0, e_1, e_2, \dots, e_{n-1}$), 其序列模式 $K(s) = k_0 k_1 k_2 \dots k_{n-1}$, 满足 $k_i = \text{kind}(e_i)$, $0 \leq i < n$. 例如, $s = e_0, e_1, e_2$ 且 $e_0 \in R, e_1 \in W, e_2 \in R$, 则 $K(s) = RWR$. 将定义 17 中 *perloc* 作用的访存事件序列扩展至序列模式. 对于任意长度为 n ($n > 0$) 的序列模式 $X = k_0 k_1 k_2 \dots k_{n-1}$, $\text{perloc}(X) \triangleq \forall e_0 \in k_0, e_1 \in k_1, \dots, e_{n-1} \in k_{n-1}, \text{perloc}(e_0, e_1, e_2, \dots, e_{n-1})$. 类似地, 将定义 19 中 SCPL 的判定对象由访存序列扩展至序列模式. 对于任意长度为 n ($n > 0$) 的序列模式 $X = k_0 k_1 k_2 \dots k_{n-1}$, $\text{SCPL}(X) \triangleq \forall e_0 \in k_0, e_1 \in k_1, \dots, e_{n-1} \in k_{n-1}, \text{perloc}(e_0, e_1, e_2, \dots, e_{n-1}) \rightarrow \neg \text{perloc}(e_{n-1}, e_0)$.

通过引入序列模式, 将证明目标由任意的访存事件序列满足 SCPL 转为其对应的序列模式满足 SCPL. 显然, 对于任意访存序列 s , $\text{SCPL}(K(s)) \rightarrow \text{SCPL}(s)$.

4.1.2 确定有限状态自动机

定义 21 (DFA). 任意序列模式的构造被抽象为图 9 所示的确定有限状态自动机 $M^{[20]}$, 由五元组 $(Q, \Sigma, \delta, q_0, F)$ 表示, 其中,

- Q 为有限状态集合, 共有 12 个, 记为 $q_0, q_1, \dots, q_{11}$. 每个状态对应一种序列模式, 例如, q_2 代表序列模式 RR .
- Σ 为输入集合, 包含两种事件类型 R 和 W .
- δ 是状态转移函数, 类型为 $Q \times \Sigma^* \rightarrow Q$, 以当前状态和 Σ 中的某个符号或符号序列作为输入, 输出 Q 中的某个状态. 例如, $\delta(q_0, R) = q_1, \delta(q_1, \varepsilon) = q_1, \delta(q_0, RRW) = q_3, \delta(q_2, RRR) = q_2$. 由于 M 是确定有限状态自动机, $\forall X \in \Sigma^*, \forall x \in \Sigma, \forall q \in Q, \delta(q, Xx) = \delta(\delta(q, X), x)$.

- q_0 为初始状态.
- F 为终止状态的集合, $F=Q$.

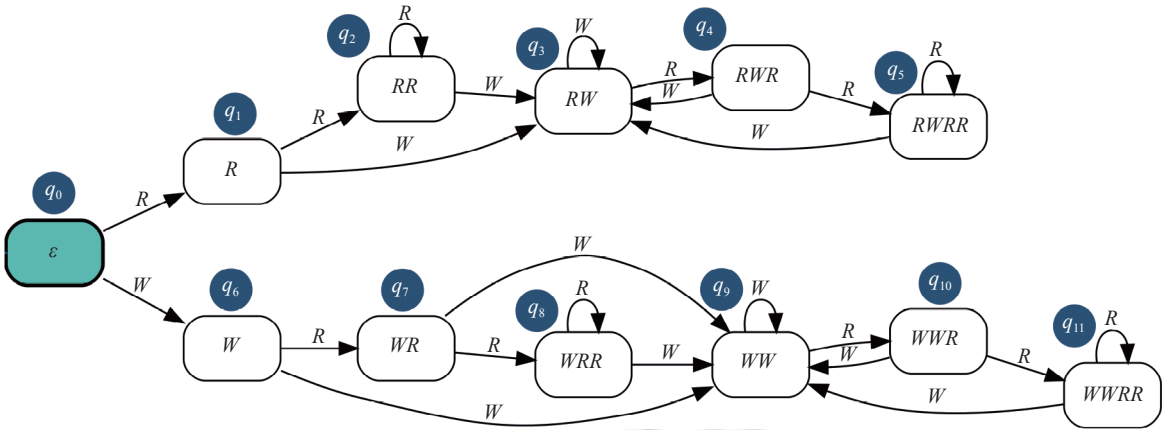


图 9 访问事件序列构造对应的 DFA

所有被 M 接受的序列模式定义为 $L(M)=\{X|X\in\Sigma^* \text{ 且 } \delta(q_0,X)\in F\}$. 由于每个状态均包含对应输入为 R 和 W 的两条出边, 任意访问序列对应的模式均可被 M 接受.

图 9 所示的确定有限状态自动机的每个状态对应一种基本序列模式 (第 4.1.3 节), 第 4.2 节证明其满足 SCPL; 每个状态转移函数对应一种模式转换 (第 4.1.4 节), 第 4.3 节证明其正确性.

4.1.3 基本序列模式

我们将所有序列模式归纳成 12 种基本模式, 对应 Q 中的每个状态 (图 9 中 q_0 – q_{11}), 逐一证明其满足 SCPL. 表 1 给出了状态、序列模式以及证明的对应关系. 详细的证明过程见第 4.2 节.

表 1 状态与序列模式和证明的对应关系

状态	序列模式	证明	状态	序列模式	证明
q_0	ε	定理1	q_6	W	定理7
q_1	R	定理2	q_7	WR	推论2
q_2	RR	定理3	q_8	WRR	推论3
q_3	RW	定理4	q_9	WW	定理8
q_4	RWR	定理5	q_{10}	WWR	定理9
q_5	$RWRR$	定理6	q_{11}	$WWRR$	推论5

4.1.4 模式转换

图 9 中的箭头表示自动机 M 的状态转移, 标注的 R 或 W 是作为输入添加至序列模式尾部的事件类型. 其中, 部分状态转移是自然而直接的, 无须证明, 例如 $\delta(q_0,R)=q_1$, $\delta(q_7,R)=q_8$ 等. 部分状态转移包含了模式间的转换. 例如, $\delta(q_2,R)=q_2$ 表示: $\text{SCPL}(RRR)$ 可以通过 $\text{SCPL}(RR)$ 证明. 引理 1 证明了模式间转换的正确性.

引理 1 (模式转换). 对于任意两个具有相同首元素和尾元素的序列模式 X 和 Y , 记为 $X=aX'b$, $Y=aY'b$, 满足 $(\text{perloc}(X)\rightarrow\text{perloc}(Y))\rightarrow\text{SCPL}(Y)\rightarrow\text{SCPL}(X)$.

证明: 根据定义 19, $\text{SCPL}(X)\triangleq\text{perloc}(X)\rightarrow\neg\text{perloc}(b,a)$, $\text{SCPL}(Y)\triangleq\text{perloc}(Y)\rightarrow\neg\text{perloc}(b,a)$, 证明目标可替换为: $(\text{perloc}(X)\rightarrow\text{perloc}(Y))\rightarrow(\text{perloc}(Y)\rightarrow\neg\text{perloc}(b,a))\rightarrow(\text{perloc}(X)\rightarrow\neg\text{perloc}(b,a))$, 由蕴含的传递性得证.

在引理 1 中, Y 为已证满足 SCPL 的序列模式, X 为待证的序列模式, 想要根据 $\text{SCPL}(Y)$ 证明 $\text{SCPL}(X)$, 只需证明 $\text{perloc}(X)\rightarrow\text{perloc}(Y)$. 图 10 给出了利用引理 1 证明 $\text{SCPL}(RRR)$ 的思路示意图. 即先证明 $\text{SCPL}(RR)$, 再证明 $\text{perloc}(RRR)\rightarrow\text{perloc}(RR)$. 注意, 模式之间的转换必须包含序列模式的首尾两个元素. 实际上, 序列模式的转换过程

是利用了 *perloc* 关系的部分传递性. 因此, 证明序列模式转换的正确性实际是证明在某个序列模式下的 *perloc* 关系的传递性. 关于 *perloc* 关系传递性的补充讨论见第 4.4 节.

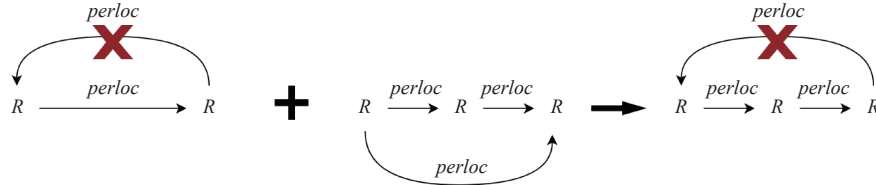


图 10 基于访存事件序列等价转换的证明思路示意

表 2 给出了自动机 M 中需特别证明的模式转换, 即 $perloc(X) \rightarrow perloc(Y)$. 为了方便阅读, 我们在介绍定理和推论时用大写字母表示序列模式转换前后保留的元素, 用小写字母表示转换后消除的元素. 例如, RwR 表示序列模式 RWR 转换为 RR .

表 2 需要证明的序列模式转换

状态转移	转换前模式 X	转换后模式 Y	助记符	证明
$\delta(q_2, R)=q_2$	RRR	RR	RrR	定理10
$\delta(q_3, W)=q_3$	RWW	RW	RwW	定理11
$\delta(q_2, W)=q_3$	RRW	RW	RrW	定理12
$\delta(q_7, W)=q_9$	WRW	WW	WrW	定理13
$\delta(q_9, W)=q_9$	WWW	WW	WwW	定理14
$\delta(q_4, W)=q_3$	$RWRW$	RW	$RwrW$	推论6
$\delta(q_5, R)=q_5$	$RWRRR$	$RWRR$	$RWRrR$	推论7
$\delta(q_8, W)=q_9$	$WRRW$	WW	$WrrW$	推论8
$\delta(q_5, W)=q_3$	$RWRRW$	RW	$RwrrW$	推论9
$\delta(q_8, R)=q_8$	$WRRR$	WRR	$WRrR$	推论10
$\delta(q_{10}, W)=q_9$	$WWRW$	WW	$WwrW$	推论11
$\delta(q_{11}, R)=q_{10}$	$WWRRR$	$WWRR$	$WWRrR$	推论12
$\delta(q_{11}, W)=q_9$	$WWRRW$	WW	$WwrrW$	推论13

虽然访存序列中相邻的两个事件满足 *perloc* 关系, 但根据事件类型不同可以得到更为具体的关系. 例如, 两个满足 *perloc* 关系的读事件只可能存在 *poloc* 关系, 其他关系 (*rf*, *fr* 或 *co*) 根据定义至少需要一个写事件. 表 3 对此类结论进行了总结. 证明是显然的.

表 3 根据事件类型可推断的具体 *perloc* 关系

a 类型	b 类型	若 $perloc(a,b)$, a 和 b 可能满足的关系
R	R	$poloc(a,b)$
W	W	$poloc(a,b) \vee co(a,b)$
R	W	$poloc(a,b) \vee fr(a,b)$
W	R	$poloc(a,b) \vee rf(a,b)$

4.2 基本序列模式满足 SCPL 的证明

定理 1 (ϵ 模式). 空访存事件序列满足 SCPL.

证明: 空序列对应的二元关系为空集, 证明是显然的.

定理 2 (R 模式). 仅包含一个读事件的访存事件序列满足 SCPL.

证明: 由于 *perloc* 包含的二元关系是非自反的, 证明是显然的.

引理 2. $\forall a, b \in E, poloc(a, b) \rightarrow \neg poloc(b, a)$.

证明: 由 $poloc$ 的非对称性可证.

定理 3 (RR 模式). $\forall a, b \in R, perloc(a, b) \rightarrow \neg perloc(b, a)$.

证明: 根据表 3, 将目标替换为已知 $poloc(a, b)$, 求证 $\neg poloc(b, a)$. 由引理 2 可证.

引理 3. $\forall a, b \in E, fr(a, b) \rightarrow \neg poloc(b, a)$.

证明: 已知 $fr(a, b)$ 证明 $\neg poloc(b, a)$.

(1) 反证法. 假设 $poloc(b, a)$ 成立.

(2) 由已知条件 $fr(a, b)$ 以及 fr 的定义可得: $\exists x \in E, rf(x, a) \wedge co(x, b)$. 其中, $b \in W \wedge x \in W$.

(3) 由公理 4 可知: $rf(x, a) \rightarrow \exists y \in W, loc(y, a) \wedge gmo(x, y) \wedge (po(y, a) \vee gmo(y, a))$.

(4) 由 co 的定义可知: $co(x, b) \rightarrow gmo(x, b)$.

(5) 由 (1) 中假设条件 $poloc(b, a)$ 以及 $poloc$ 定义可知: $po(b, a) \wedge loc(b, a)$.

(6) 由 (4) 和 (5) 可知: $loc(b, a) \wedge gmo(x, b) \wedge po(b, a)$. 结合 (2)、(3) 和 (4) 推出矛盾. 得证.

引理 4. $\forall a, b \in E, rf(a, b) \rightarrow \neg fr(b, a)$.

证明: 已知 $rf(a, b)$ 证明 $\neg fr(b, a)$.

(1) 反证法. 假设 $fr(b, a)$ 成立.

(2) 由 (1) 以及 fr 的定义可得: $\exists x \in E, rf(x, b) \wedge co(x, a)$.

(3) 由公理 2 以及 $rf(a, b)$ 可知: $\exists x \in E, x \neq a \wedge rf(x, b)$. 与 (2) 推出矛盾. 得证.

引理 5. $\forall a, b \in E, rf(a, b) \rightarrow \neg poloc(b, a)$.

证明: 已知 $rf(a, b)$ 证明 $\neg poloc(b, a)$.

(1) 反证法. 假设 $poloc(b, a)$ 成立.

(2) 由 $rf(a, b)$ 可知: $a \in W, b \in R$. 由 (1) 中假设 $poloc(b, a)$ 可知: $ppo1(b, a)$.

(3) 由 (2), ppo 的定义以及公理 2 可得: $gmo(b, a)$, 又由 gmo 的非对称性可知: $\neg gmo(a, b)$.

(4) 由 $rf(a, b)$ 以及公理 4 可知: $po(a, b) \vee gmo(a, b)$.

(5) 由 (1) 中假设 $poloc(b, a)$ 可知: $po(b, a)$. 由 po 关系的非对称性可知: $\neg po(a, b)$.

(6) 由 (3)、(4) 和 (5) 推出矛盾.

定理 4 (RW 模式). $\forall a \in R, \forall b \in W, perloc(a, b) \rightarrow \neg perloc(b, a)$.

证明: 根据表 3, 将目标替换为: 已知 $poloc(a, b) \vee fr(a, b)$, 求证 $\neg poloc(b, a) \wedge \neg rf(b, a)$.

(1) 由引理 2 和引理 5 的逆否命题可知: $poloc(a, b) \rightarrow \neg poloc(b, a) \wedge \neg rf(b, a)$.

(2) 由引理 3 和引理 4 的逆否命题可知: $fr(a, b) \rightarrow \neg poloc(b, a) \wedge \neg rf(b, a)$.

(3) 由 (1) 和 (2) 可知: $poloc(a, b) \vee fr(a, b) \rightarrow \neg poloc(b, a) \wedge \neg rf(b, a)$. 得证.

引理 6. $\forall a, b \in E, fri(a, b) \rightarrow poloc(a, b)$.

证明: 已知 $fri(a, b)$ 证明 $poloc(a, b)$. 由 fri 的定义可知: $int(a, b) \wedge loc(a, b)$, 这里只证明 $po(a, b)$. 又因为 $int(a, b)$ 以及 po 定义的推论, 将证明目标转为 $\neg po(b, a)$.

(1) 反证法. 假设 $po(b, a)$ 成立.

(2) 由 (1)、 $loc(a, b)$ 以及 loc 的对称性可得: $poloc(b, a)$.

(3) 由引理 3: $fr(a, b) \rightarrow \neg poloc(b, a)$. 由 fri 的定义以及 (2) 推出矛盾.

引理 7. $\forall a, b \in E, rfi(a, b) \rightarrow poloc(a, b)$.

证明: 已知 $rfi(a, b)$ 证明 $poloc(a, b)$. 由 rfi 的定义可知: $int(a, b) \wedge loc(a, b)$, 这里只证明 $po(a, b)$. 又因为 $int(a, b)$ 以及 po 定义的推论, 将证明目标转为 $\neg po(b, a)$.

(1) 反证法. 假设 $po(b, a)$ 成立.

(2) 由 (1) 以及 loc 的对称性可得: $poloc(b, a)$.

(3) 由 (2)、 $ppo1$ 的定义以及公理 2 可得: $gmo(b, a)$.

(4) 由 $rfi(a, b)$ 以及公理 4 可得: $po(a, b) \vee gmo(a, b)$.

(5) 由 (1)、(3) 以及 po 和 gmo 的非对称性, 与 (4) 推出矛盾.

引理 8. $\forall a, b \in E, rfe(a, b) \rightarrow gmo(a, b)$.

证明: 已知 $rfe(a, b)$ 证明 $gmo(a, b)$.

(1) 由 $rfe(a, b)$ 的定义以及公理 4 可得: $po(a, b) \vee gmo(a, b)$.

(2) 由 $rfe(a, b)$ 可得: $ext(a, b)$, 又由 po 的定义可得: $\neg po(a, b)$.

(3) 由 (1) 和 (2) 可证 $gmo(a, b)$.

引理 9. $\forall a, b \in E, fr(a, b) \rightarrow gmo(a, b)$.

证明: 由定义 14 中的推论可知 $fr(a, b) \triangleq fri(a, b) \vee fre(a, b)$, 将条件分两种情况证明.

(1) 若 $fri(a, b)$ 成立, 由引理 6 可得: $poloc(a, b)$, 又由 $ppo1$ 的定义以及公理 2 可证 $gmo(a, b)$.

(2) 若 $fre(a, b)$ 成立:

(2.1) 由 fre 的定义以及 loc 的对称性可得: $a \neq b \wedge loc(b, a)$.

(2.2) 反证法. 假设 $\neg gmo(a, b)$ 成立. 由 (2.1) 和 gmo 定义, 将 $\neg gmo(a, b)$ 替换为 $gmo(b, a)$.

(2.3) 由 fre 的定义可得: $\exists x \in E, rfx(a, x) \wedge co(x, b)$.

(2.4) 由 (2.3) 以及 co 的定义可得: $gmo(x, b)$.

(2.5) 由 (2.3) 以及公理 4 可得: $\exists y \in W, loc(y, a) \wedge gmo(x, y) \wedge (po(y, a) \vee gmo(y, a))$.

(2.6) 由 (2.1)、(2.2) 和 (2.4) 可得: $loc(b, a) \wedge gmo(x, b) \wedge gmo(b, a)$, 与 (2.5) 推出矛盾.

引理 10. $\forall a, b, c \in E, fr(a, b) \wedge rfb(b, c) \rightarrow \neg poloc(c, a)$.

证明: 已知 $fr(a, b), rfb(b, c)$ 证明 $\neg poloc(c, a)$.

(1) 反证法. 假设 $poloc(c, a)$ 成立.

由定义 14 中的推论可知: $fr(a, b) \triangleq fri(a, b) \vee fre(a, b), rfb(b, c) \triangleq rfi(b, c) \vee rfe(b, c)$, 将条件分 4 种情况证明.

(2) 若 $fri(a, b) \wedge rfi(b, c)$ 成立:

(2.1) 由 $fri(a, b)$ 和引理 6 可得: $poloc(a, b)$.

(2.2) 由 $rfi(b, c)$ 和引理 7 可得: $poloc(b, c)$.

(2.3) 由 (2.1)、(2.2) 以及 $poloc$ 的传递性可得: $poloc(a, c)$.

(2.4) 由 $poloc$ 的非对称性推出矛盾.

(3) 若 $fri(a, b) \wedge rfe(b, c)$ 成立:

(3.1) 由定义 14 可得: $int(a, b) \wedge ext(b, c)$.

(3.2) 由 (3.1) 以及 int 和 ext 定义可得: $ext(a, c)$.

(3.3) 由 (3.2) 以及 po 的定义与 (1) 推出矛盾.

(4) 若 $fre(a, b) \wedge rfi(b, c)$ 成立, 证明方法与 (3) 类似.

(5) 若 $fre(a, b) \wedge rfe(b, c)$ 成立:

(5.1) 由引理 8 和引理 9 可得: $gmo(a, b) \wedge gmo(b, c)$.

下面求证 $ppo2(c, a)$. 由 $ppo2$ 的定义, 我们需要证明 5 个条件. 其中, $poloc(c, a), c \in R$ 和 $a \in R$ 是已知的, 需要证明 $\exists x \in W, poloc(c, x) \wedge poloc(x, a)$ 以及 $\exists x \in W, rfx(x, c) \wedge rfx(x, a)$.

(5.2) 证明 $\exists x \in W, poloc(c, x) \wedge poloc(x, a)$. 反证法. 假设 $\exists x \in W, poloc(c, x) \wedge poloc(x, a)$.

(5.2.1) 由 $poloc(c, x), ppo1$ 的定义以及公理 2 可得: $gmo(c, x)$.

(5.2.2) 由 $fre(a, b)$ 以及 fr 的定义可得: $\exists y \in E, rfy(y, a) \wedge co(y, b)$.

(5.2.3) 由 (5.2.2) 以及 co 的定义可得: $gmo(y, b)$.

(5.2.4) 由 (5.2.2) 以及公理 4 可得: $\exists z \in W, loc(z, a) \wedge gmo(y, z) \wedge (po(z, a) \vee gmo(z, a))$.

(5.2.5) 由 $poloc(x, a)$ 以及 $poloc$ 的定义可得: $po(x, a) \wedge loc(x, a)$.

(5.2.6) 由 (5.2.4) 和 (5.2.5) 可得: $\neg gmo(y, x)$. 又由 gmo 的定义可得: $gmo(x, y)$.

(5.2.7) 由 (5.2.1)、(5.2.3)、(5.2.6) 以及 gmo 的传递性可得: $gmo(c, b)$.

(5.2.8) 由 (5.2.7)、(5.1) 以及 gmo 的非对称性推出矛盾.

(5.3) 证明 $\exists x \in W, rf(x, c) \wedge rf(x, a)$. 反证法. 假设 $\exists x \in W, rf(x, c) \wedge rf(x, a)$.

(5.3.1) 由 (5.3) 中假设 $rf(x, c)$ 、已知条件 $rfe(b, c)$ 以及公理 2 可得: $b=x$.

(5.3.2) 由 (5.3.1) 以及 $rf(x, a)$ 可得: $rf(b, a)$.

(5.3.3) 由 (5.3.2)、 $fir(b, a)$ 以及 fir 定义可得: $\exists y \in E, rf(y, a) \wedge co(y, b)$.

(5.3.4) 由 (5.3.2)、(5.3.3) 以及公理 2 可得: $b=y$. 进而得到 $co(b, b)$, 矛盾.

(5.4) 由 (5.2) 和 (5.3) 中的结论可得: $ppo2(c, a)$. 进而通过 ppo 的定义和公理 2 可得: $gmo(c, a)$.

(5.5) 由 (5.1) 和 gmo 的传递性可得: $gmo(a, c)$. 由 gmo 的非对称性, 与 (5.4) 推出矛盾.

定理 5 (RWR 模式). $\forall a, c \in R, \forall b \in W, perloc(a, b, c) \rightarrow \neg perloc(c, a)$.

证明: 根据表 3, 将目标替换为: $poloc(a, b) \vee fir(a, b), poloc(b, c) \vee rf(b, c)$ 证明 $\neg poloc(c, a)$.

(1) 反证法. 假设 $poloc(c, a)$ 成立.

下面将已知条件中的析取范式展开为 4 种情况分别证明.

(2) 若 $poloc(a, b) \wedge poloc(b, c)$, 由 $poloc$ 的传递性可知 $poloc(a, c)$, 又由 (1) 以及引理 5 推出矛盾.

(3) 若 $fir(a, b) \wedge poloc(b, c)$, 由 (1) 以及 $poloc$ 的传递性可知 $poloc(b, a)$, 又由 $fir(a, b)$ 以及引理 7 推出矛盾.

(4) 若 $poloc(a, b) \wedge rf(b, c)$, 由 (1) 以及 $poloc$ 的传递性可知 $poloc(c, b)$, 又由 $rf(b, c)$ 以及引理 9 推出矛盾.

(5) 若 $fir(a, b) \wedge rf(b, c)$, 由 (1) 以及引理 10 推出矛盾.

推论 1 (RRW 模式). $\forall a \in R, b \in R, c \in W, perloc(a, b, c) \rightarrow \neg perloc(c, a)$.

证明: 将 a, b, c 分别带入定理 5 中的 c, a, b 可证.

定理 6 (RWRR 模式). $\forall a, c, d \in R, b \in W, perloc(a, b, c, d) \rightarrow \neg perloc(d, a)$.

证明: 根据表 3, 将目标替换为: 已知 $perloc(a, b), perloc(b, c), poloc(c, d)$, 求证 $\neg poloc(d, a)$.

(1) 反证法. 假设 $poloc(d, a)$ 成立.

(2) 由 $poloc(c, d), poloc(d, a)$ 以及 $poloc$ 的传递性可得: $poloc(c, a)$.

(3) 由 $poloc(c, a), perloc(a, b)$ 以及推论 1 可得: $\neg perloc(b, c)$. 与条件中的 $perloc(b, c)$ 推出矛盾.

定理 7 (W 模式). 仅包含一个写事件的访存事件序列满足 SCPL.

证明: 同定理 2, 证明是显然的.

推论 2 (WR 模式). $\forall a \in W, b \in R, perloc(a, b) \rightarrow \neg perloc(b, a)$.

证明: 由定理 4 以及逆否命题的推理定律可证.

引理 11. $\forall a, b \in W, perloc(a, b) \rightarrow co(a, b)$.

证明: 根据表 3, 将目标替换为: 已知 $a \in W, b \in W, poloc(a, b) \vee co(a, b)$, 求证 $co(a, b)$.

我们将目标替换为: 已知 $a \in W, b \in W, poloc(a, b)$, 求证 $co(a, b)$.

(1) 由 $b \in W \wedge poloc(a, b)$, $ppo1$ 的定义以及公理 2 可得: $gmo(a, b)$.

(2) 由 (1)、 $a \in W, b \in W$ 以及 co 的定义可得: $co(a, b)$.

引理 12. $\forall a, b \in W, co(a, b) \rightarrow \neg perloc(b, a)$.

证明: 由引理 11 以及 co 的非对称性可证.

定理 8 (WW 模式). $\forall a, b \in W, perloc(a, b) \rightarrow \neg perloc(b, a)$.

证明: 由表 3 和引理 11, 将目标替换为: 已知 $co(a, b)$, 求证 $\neg co(b, a)$. 由 co 的非对称性可证.

推论 3 (WRR 模式). $\forall a \in W, \forall b, c \in R, perloc(a, b, c) \rightarrow \neg perloc(c, a)$.

证明: 由定理 5 以及否定引入定律可证.

定理 9 (WWR 模式). $\forall a, b \in W, \forall c \in R, perloc(a, b, c) \rightarrow \neg perloc(c, a)$.

证明: 根据表 3 和引理 11, 将目标替换为: 已知 $co(a, b), poloc(b, c) \vee rf(b, c)$, 求证 $\neg (poloc(c, a) \vee fir(c, a))$.

(1) 反证法. 假设 $poloc(c, a) \vee fir(c, a)$ 成立.

将条件分以下 4 种情况证明.

(2) 若 $co(a,b) \wedge poloc(b,c) \wedge poloc(c,a)$ 成立:

(2.1) 由 $poloc(b,c) \wedge poloc(c,a)$ 以及 $poloc$ 的传递性可得: $poloc(b,a)$.

(2.2) 由 $co(a,b)$ 、(2.1) 以及引理 12 可推得矛盾.

(3) 若 $co(a,b) \wedge rf(b,c) \wedge poloc(c,a)$ 成立:

(3.1) 由 $rf(b,c)$ 以及公理 4 可得: $po(b,c) \vee gmo(b,c)$.

(3.2) 由 $poloc(c,a)$ 、 $ppo1$ 的定义以及公理 2 可得: $gmo(c,a)$.

(3.3) 由 co 的定义可得: $gmo(a,b)$.

(3.4) 由 (3.2)、(3.3) 以及 gmo 的传递性可得: $gmo(c,b)$. 又因 gmo 的非对称性可得: $\neg gmo(b,c)$.

目前, 想要由 (3.1) 和 (3.4) 推出矛盾, 我们将求证目标变为 $\neg po(b,c)$.

(3.5) 反证法. 假设 $po(b,c)$ 成立. 又因 $rf(b,c)$ 可知 $loc(b,c)$, 可得: $poloc(b,c)$.

(3.6) 由 (3.5)、 $poloc(c,a)$ 以及 $poloc$ 的传递性可得: $poloc(b,a)$.

(3.7) 由 $co(a,b)$ 以及引理 12 的逆否命题可与 (3.6) 推出矛盾.

(4) 若 $co(a,b) \wedge poloc(b,c) \wedge fr(c,a)$ 成立:

(4.1) 由 $fr(c,a)$ 以及 fr 的定义可得: $\exists x \in E, rf(x,c) \wedge co(x,a)$.

(4.2) 由 $rf(x,c)$ 以及公理 4 可得: $\exists y \in W, loc(y,c) \wedge gmo(x,y) \wedge (po(y,c) \vee gmo(y,c))$.

将 b 代入 (4.2) 的 y , 将证明目标变为 $loc(b,c) \wedge gmo(x,b) \wedge po(b,c)$, 其中, $loc(b,c)$ 是显然的.

(4.3) 由 $co(a,b)$ 、(4.1) 中的 $co(x,a)$ 、 co 的定义以及 gmo 的传递性可得: $gmo(x,b)$.

(4.4) 由 $poloc(b,c)$ 以及 $poloc$ 的定义可得: $po(b,c)$. 得证.

(5) 若 $co(a,b) \wedge rf(b,c) \wedge fr(c,a)$ 成立:

(5.1) 由 $fr(c,a)$ 以及 fr 的定义可得: $\exists x \in E, rf(x,c) \wedge co(x,a)$.

(5.2) 由 (5.1)、 $rf(b,c)$ 以及公理 2 可得: $x=b$.

(5.3) 由 (5.1) 以及 (5.2) 可得: $co(b,a)$. 由 co 的非对称性与 $co(a,b)$ 推出矛盾.

推论 4 (WRW 模式). $\forall a, c \in W, \forall b \in R, perloc(a, b, c) \rightarrow \neg perloc(c, a)$.

证明: 将 a 、 b 、 c 分别带入定理 9 中的 b 、 c 、 a 可证.

推论 5 (WWR 模式). $\forall a \in W, b \in W, c \in R, d \in R, perloc(a, b, c, d) \rightarrow \neg perloc(d, a)$.

证明: 已知 $a \in W, b \in W, c \in R, d \in R, perloc(a, b, c, d)$, 求证 $\neg perloc(d, a)$.

(1) 反证法. 假设 $perloc(d, a)$ 成立.

(2) 将 d 、 a 、 b 分别代入定理 11 的 a 、 b 、 c , 可得: $perloc(d, b)$.

(3) 将 d 、 b 、 c 分别代入定理 5 的 a 、 b 、 c , 可得: $\neg perloc(c, d)$. 与条件中的 $perloc(c, d)$ 矛盾.

4.3 序列模式转换正确性的证明

序列模式的转换共有 13 种情况需要证明 (见表 2). 其中 5 种情况处理的是长度为 3 的序列模式 (RRR 、 RWW 、 RRW 、 WRW 和 WWW), 将被作为基础定理进行证明. 其余 8 种情况可由上述 5 个定理推导证明.

图 11 给出了序列模式转换的 5 个定理和 8 个推论的示意, 其中, 不同颜色的虚线箭头代表不同序列模式下产生的 $perloc$ 传递性, 黑色/灰色的 R 或 W 代表模式转换后保留/消除的部分. 图 11(b) 的推论可由一次或多次应用图 11(a) 的定理进行证明.

定理 10 (RrR 转换). $\forall a, b, c \in R, perloc(a, b, c) \rightarrow perloc(a, c)$.

证明: 根据表 3, 将目标替换为: 已知 $poloc(a, b)$, $poloc(b, c)$, 求证 $poloc(a, c)$. 由 $poloc$ 的传递性可证.

引理 13. $\forall a, b, c, d \in E, poloc(a, b) \wedge rf(c, a) \wedge rf(d, b) \rightarrow c=d \vee co(c, d)$.

证明: 已知 $poloc(a, b)$, $rf(c, a)$, $rf(d, b)$, 求证 $c=d \vee co(c, d)$. 由排中律将 $c=d \vee c \neq d$ 作为假设并分情况讨论. 若 $c=d$ 成立, 证明是显然的. 这里只给出 $c \neq d$ 的证明过程. 由 $c \neq d$ 以及 co 的定义可知: $co(c, d) \vee co(d, c)$. 我们将证明目标替换为 $\neg co(d, c)$.

- (1) 反证法. 假设 $co(d,c)$ 成立.
- (2) 由 $rf(d,b)$ 、 $co(d,c)$ 以及 fr 的定义可得: $fr(b,c)$.
- (3) 由 (2)、 $rf(c,a)$ 以及定理 5 可得: $\neg perloc(b,a)$. 与 $poloc(a,b)$ 推出矛盾.

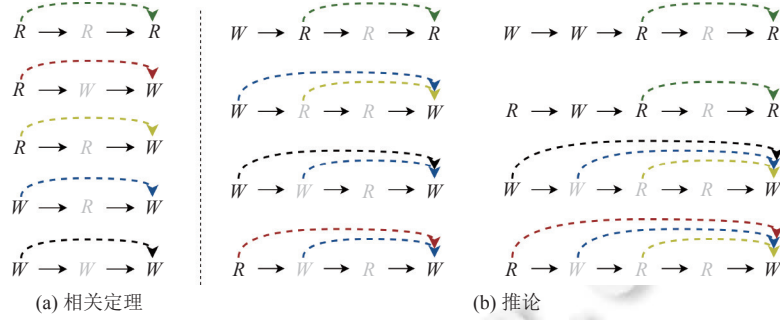


图 11 序列模式转换的相关定理和推论示意

引理 14. $\forall a,b \in E, coi(a,b) \rightarrow poloc(a,b)$.

证明: 由 coi 的定义, 将目标替换为: 已知 $co(a,b)$, $int(a,b)$, 求证 $po(a,b) \wedge loc(a,b)$. 显然, $loc(a,b)$ 是成立的. 由 $int(a,b)$ 以及 po 的定义可得: $po(a,b) \vee po(b,a)$. 最终证明 $\neg po(b,a)$ 即可.

- (1) 反证法. 假设 $po(b,a)$ 成立.
- (2) 由 $po(b,a)$ 、 $loc(a,b)$ 、 $ppo1$ 的定义以及公理 2 可得: $gmo(b,a)$.
- (3) 由 $co(a,b)$ 以及 co 的定义可得: $gmo(a,b)$. 由 gmo 的非对称性与 (2) 推出矛盾.

定理 11 (RwW 转换). $\forall a \in R, \forall b,c \in W, perloc(a,b,c) \rightarrow perloc(a,c)$.

证明: 由表 3 和引理 11, 将目标替换为: 已知 $poloc(a,b) \vee fr(a,b)$, $co(b,c)$, 求证 $perloc(a,c)$. 分两种情况证明.

- (1) 若 $poloc(a,b)$ 成立:
 - (1.1) 由 int 和 ext 的定义可得: $int(b,c) \vee ext(b,c)$. 分开讨论.
 - (1.2) 若 $int(b,c)$ 成立:
 - (1.2.1) 由 $int(b,c)$ 、 $co(b,c)$ 以及引理 14 可得: $poloc(b,c)$.
 - (1.2.2) 由 $poloc(a,b)$ 、 $poloc(b,c)$ 以及 $poloc$ 的传递性可得: $poloc(a,c)$. $perloc(a,c)$ 得证.
 - (1.3) 若 $ext(b,c)$ 成立:
 - (1.3.1) 由 $poloc(a,b)$ 、 $ppo1$ 的定义以及公理 2 可得: $gmo(a,b)$.
 - (1.3.2) 由假设中 $co(b,c)$ 以及 co 定义可得: $gmo(b,c)$.
 - (1.3.3) 由公理 1 可知: $\exists x \in W, rf(x,a)$.
 - (1.3.4) 证明 $x \neq c$.
 - (1.3.4.1) 反证法. 假设 $x=c$. 由 (1.3.2) 可得: $rf(c,a)$.
 - (1.3.4.2) 由 $poloc(a,b)$ 可知 $int(a,b)$, 又因 $ext(b,c)$ 可得: $ext(a,c)$, 进而得到 $rfe(c,a)$.
 - (1.3.4.3) 由 (1.3.4.2) 以及引理 8 可得: $gmo(c,a)$.
 - (1.3.4.4) 由 (1.3.1)、(1.3.2) 以及 gmo 的定义可得: $gmo(a,c)$.
 - (1.3.4.5) 由 (1.3.4.3)、(1.3.4.4) 以及 gmo 的非对称性推出矛盾.
 - (1.3.5) 证明 $gmo(x,c)$. 由 (1.3.4) 以及 gmo 的定义可得: $gmo(x,c) \vee gmo(c,x)$. 只需证 $\neg gmo(c,x)$.
 - (1.3.5.1) 反证法. 假设 $gmo(c,x)$ 成立.
 - (1.3.5.2) 由 (1.3.3) 中 $rf(x,a)$ 以及定义 14 可得: $rft(x,a) \vee rfe(x,a)$. 分开讨论.
 - (1.3.5.3) 若 $rft(x,a)$ 成立:
 - (1.3.5.3.1) 由引理 7 可得: $poloc(x,a)$, 又因 $poloc(a,b)$ 以及 $poloc$ 传递性可得: $poloc(x,b)$.

(1.3.5.3.2) 由 (1.3.5.3.1) 以及引理 11 可得: $co(x,b)$. 由 co 定义进而得到: $gmo(x,b)$.

(1.3.5.3.3) 由 (1.3.5.1)、(1.3.5.3.2) 以及 gmo 的传递性可得: $gmo(c,b)$.

(1.3.5.3.4) 因 (1.3.2) 以及 gmo 的非对称性与 (1.3.5.3.3) 推出矛盾.

(1.3.5.4) 若 $rfe(x,a)$ 成立:

(1.3.5.4.1) 由 $rfe(x,a)$ 以及引理 8 可得: $gmo(x,a)$.

(1.3.5.4.2) 根据 (1.3.5.4.1)、(1.3.1)、(1.3.2) 以及 gmo 的传递性可得: $gmo(x,c)$.

(1.3.5.4.3) 根据 (1.3.5.4.2) 以及 gmo 的非对称性与 (1.3.5.1) 推出矛盾.

(1.3.6) 由 (1.3.3)、(1.3.5) 以及 co 定义可得: $rf(x,a) \wedge co(x,c)$. 又由 fr 定义可得: $fr(a,c)$. $perloc(a,c)$ 得证.

(2) 若 $fr(a,b)$ 成立:

(2.1) 由 fr 定义可得: $\exists x \in E, rf(x,a) \wedge co(x,b)$.

(2.2) 由假设中 $co(b,c)$ 、(2.1) 中 $co(x,b)$ 以及 co 的传递性可得: $co(x,c)$.

(2.3) 由 (2.1) 中 $rf(x,a)$ 、(2.2) 以及 fr 定义可得: $fr(a,c)$. $perloc(a,c)$ 得证.

定理 12 (RrW 转换). $\forall a,b \in R, \forall c \in W, perloc(a,b,c) \rightarrow perloc(a,c)$.

证明: 根据表 3, 将目标替换为: 已知 $poloc(a,b)$, $poloc(b,c) \vee fr(b,c)$, 求证 $perloc(a,c)$. 分两种情况证明.

(1) 若 $poloc(b,c)$ 成立, 由 $poloc$ 的传递性可证.

(2) 若 $fr(b,c)$ 成立:

(2.1) 由 fr 的定义可知: $\exists x \in E, rf(x,b) \wedge co(x,c)$.

(2.2) 由公理 1 可知: $\exists y \in W, rf(y,a)$.

(2.3) 由 $rf(y,a)$ 、 $rf(x,b)$ 、 $poloc(a,b)$ 以及引理 13 可得: $x=y \vee co(y,x)$. 分两种情况证明.

(2.4) 若 $x=y$ 成立, 由 $rf(x,a)$ 、 $co(x,c)$ 以及 fr 的定义可得: $fr(a,c)$. 由 $perloc$ 的定义, $perloc(a,c)$ 得证.

(2.5) 若 $co(y,x)$ 成立, 将 a 、 x 、 c 分别代入定理 11 的 a 、 b 、 c , 可得: $perloc(a,c)$.

定理 13 (WrrW 转换). $\forall a,c \in W, \forall b \in R, perloc(a,b,c) \rightarrow perloc(a,c)$.

证明: 已知 $a \in W, b \in R, c \in W, perloc(a,b)$, $perloc(b,c)$, 求证 $perloc(a,c)$.

(1) 证明 $a \neq c$. 由反证法可证: 若 $a=c$ 成立, 可得: $perloc(a,b)$, $perloc(b,a)$, 由推论 2 可推出矛盾.

(2) 由 (1) 以及 co 定义可得: $co(a,c) \vee co(c,a)$.

为了证明 $perloc(a,c)$, 只需证明 $\neg co(c,a)$. 反证法. 假设 $co(c,a)$ 成立, 由推论 4 推出矛盾.

定理 14 (WwW 转换). $\forall a,b,c \in W, perloc(a,b,c) \rightarrow perloc(a,c)$.

证明: 根据表 3 及引理 11, 将目标替换为: 已知 $co(a,b) \wedge co(b,c)$, 求证 $co(a,c)$. 由 co 的传递性可证.

推论 6 (RwrW 转换). $\forall a,c \in R, \forall b,d \in W, perloc(a,b,c,d) \rightarrow perloc(a,d)$.

证明: 已知 $a \in R, b \in W, c \in R, d \in W, perloc(a,b)$, $perloc(b,c)$, $perloc(c,d)$, 求证 $perloc(a,d)$.

(1) 将 b 、 c 、 d 分别代入定理 13 的 a 、 b 、 c , 可得: $perloc(b,d)$.

(2) 将 a 、 b 、 d 分别代入定理 11 的 a 、 b 、 c , 可得: $perloc(a,d)$.

推论 7 (RWRrR 转换). $\forall a,c,d,e \in R, \forall b \in W, perloc(a,b,c,d,e) \rightarrow perloc(a,b,c,e)$.

证明: 已知 $a \in R, b \in W, c \in R, d \in R, e \in R, perloc(a,b,c,d,e)$, 求证 $perloc(a,b,c,e)$. 只需证明 $perloc(c,e)$. 可由将 c 、 d 、 e 分别代入定理 10 的 a 、 b 、 c 证明.

推论 8 (WrrW 转换). $\forall a,d \in W, \forall b,c \in R, perloc(a,b,c,d) \rightarrow perloc(a,d)$.

证明: 已知 $a \in W, b \in R, c \in R, d \in W, perloc(a,b,c,d)$, 求证 $perloc(a,d)$.

(1) 将 b 、 c 、 d 分别代入定理 12 的 a 、 b 、 c , 可得: $perloc(b,d)$.

(2) 将 a 、 b 、 d 分别代入定理 13 的 a 、 b 、 c , 可得: $perloc(a,d)$.

推论 9 (RwrrW 转换). $\forall a,c,d \in R, \forall b,e \in W, perloc(a,b,c,d,e) \rightarrow perloc(a,e)$.

证明: 已知 $a \in R, b \in W, c \in R, d \in R, e \in W, perloc(a,b,c,d,e)$, 求证 $perloc(a,e)$.

(1) 将 b 、 c 、 d 、 e 分别代入推论 8 的 a 、 b 、 c 、 d , 可得: $perloc(b,e)$.

(2) 将 a 、 b 、 e 分别代入定理 11 的 a 、 b 、 c , 可得: $perloc(a,e)$.

推论 10 (WRRrR 转换). $\forall a \in W, \forall b, c, d \in R, c \in R, d \in R, perloc(a,b,c,d) \rightarrow perloc(a,b,d)$.

证明: 已知 $a \in W, b \in R, c \in R, d \in R, perloc(a,b,c,d)$, 求证 $perloc(a,b,d)$. 只需证明 $perloc(b,d)$. 可由将 b 、 c 、 d 分别代入定理 10 的 a 、 b 、 c 证明.

推论 11 (WwrW 转换). $\forall a, b, d \in W, \forall c \in R, perloc(a,b,c,d) \rightarrow perloc(a,d)$.

证明: 已知 $a \in W, b \in W, c \in R, d \in W, perloc(a,b,c,d)$, 求证 $perloc(a,d)$.

(1) 将 b 、 c 、 d 分别代入定理 13 的 a 、 b 、 c , 可得: $perloc(b,d)$.

(2) 将 a 、 b 、 d 分别代入定理 14 的 a 、 b 、 c , 可得: $perloc(a,d)$.

推论 12 (WWRrR 转换). $\forall a, b \in W, \forall c, d, e \in R, perloc(a,b,c,d,e) \rightarrow perloc(a,b,c,e)$.

证明: 已知 $a \in W, b \in W, c \in R, d \in R, e \in R, perloc(a,b,c,d,e)$, 求证 $perloc(a,b,c,e)$. 只需求证 $perloc(c,e)$. 可由将 c 、 d 、 e 分别代入定理 10 的 a 、 b 、 c 证明.

推论 13 (WwrrW 转换). $\forall a, b, e \in W, \forall c, d \in R, perloc(a,b,c,d,e) \rightarrow perloc(a,e)$.

证明: 已知 $a \in W, b \in W, c \in R, d \in R, e \in W, perloc(a,b,c,d,e)$, 求证 $perloc(a,e)$.

(1) 将 b 、 c 、 d 、 e 分别代入推论 8 的 a 、 b 、 c 、 d , 可得: $perloc(b,e)$.

(2) 将 a 、 b 、 e 分别代入定理 14 的 a 、 b 、 c , 可得: $perloc(a,e)$.

4.4 不满足 $perloc$ 传递性的序列模式举例

第 4.3 节证明了 5 种满足 $perloc$ 传递性的序列模式作为模式转换的定理基础. 实际上, 并非所有长度为 3 的序列模式都具有 $perloc$ 传递性, 这也是引入 DFA 进行证明的原因. 本节使用反例证明了其余 3 种序列模式 (RWR 、 WRR 和 WWR) 不满足 $perloc$ 传递性.

命题 1 (RWR 模式). $\exists a, c \in R, \exists b \in W, perloc(a,b,c) \rightarrow \neg perloc(a,c)$.

证明: 根据表 3, 将目标替换为: 存在同时满足 $a \in R, b \in W, c \in R, perloc(a,b,c), \neg poloc(a,c)$ 的情况.

通过构造反例证明. 根据 $poloc$ 的定义, 只需构造满足 $ext(a,c)$ 的序列即可 (见图 12).

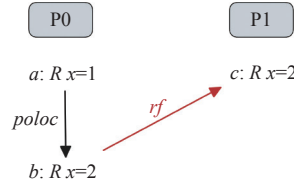


图 12 证明 RWR 模式不存在传递性的 Litmus 示例

命题 2 (WRR 模式). $\exists a \in W, \exists b, c \in R, perloc(a,b,c) \rightarrow \neg perloc(a,c)$.

证明: 根据表 3, 将目标替换为: 存在同时满足 $a \in W, b \in R, c \in R, perloc(a,b,c), \neg(rf(a,c) \vee poloc(a,c))$.

通过构造反例证明. 根据 $poloc$ 的定义, 只需构造满足 $ext(a,c)$ 的序列即可满足 $\neg poloc(a,c)$. 根据 rf 定义, 只需构造 $d \in W$ 并满足 $d \neq a \wedge rf(d,c)$ 即可满足 $\neg rf(a,c)$. 通过构造 $co(a,d)$ 满足 $d \neq a$. 对应的 Litmus 示例见图 13.

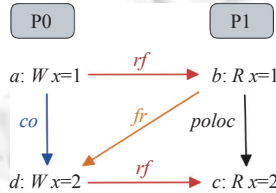
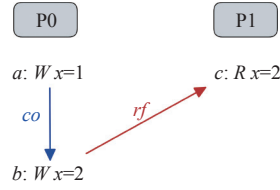


图 13 证明 WRR 模式不存在传递性的 Litmus 示例

命题 3 (WWR 模式). $\exists a, b \in W, \exists c \in R, perloc(a,b,c) \rightarrow \neg perloc(a,c)$.

证明: 与命题 2 中的反例类似, 区别在于可直接通过构造 $d=b$ 满足 $d \neq a$. 对应的 Litmus 示例见图 14.

图 14 证明 WWR 模式不存在传递性的 Litmus 示例

5 案例分析: 应用 SCPL 进行 Litmus 测试合法性判定

应用内存一致性的公理模型可形式化地对 Litmus 测试的合法性进行判定, 从而指导软件开发和芯片验证. 例如, 借助模型检验^[21]可自动对 Litmus 测试进行判定. 其中, 借助 SCPL 将大大简化公理模型对部分 Litmus 测试的形式化验证. 以第 1.2 节中的程序为例, 利用 SCPL 容易判定其非法性.

命题 4. $\forall a \in R, b \in R, c \in W, d \in W, \text{poloc}(a, b) \wedge \text{poloc}(c, d) \wedge \text{co}(c, d) \wedge \text{rfe}(d, a) \wedge \text{rfe}(c, b) \rightarrow \text{False}$.

证明: 根据 $\text{co}(c, d)$ 、 $\text{rfe}(c, b)$ 以及 fr 的定义可得 $\text{fr}(b, d)$, 又由 $\text{rfe}(d, a)$ 、 $\text{poloc}(a, b)$ 以及 SCPL 定理推出矛盾. 得证. 如果不利用 SCPL 定理, 可通过与引理 10 类似的方法进行证明, 相比之下, 基于 SCPL 的证明更为直接.

6 总结

本文证明了 RVWMO 满足 SCPL. 证明过程将序列模式的构建抽象为确定有限状态自动机, 通过对各个模式满足 SCPL 以及模式转换正确性的证明, 从而实现对任意访存序列满足 SCPL 的证明. 通过实验 (图 2) 以及案例分析 (第 5 节) 可见, SCPL 定理将大大简化 RVWMO 模型的相关形式化方法.

本研究是对 RVWMO 相关形式化方法的一个理论补充.

References:

- [1] Lamport L. How to make a correct multiprocess program execute correctly on a multiprocessor. IEEE Trans. on Computers, 1997, 46(7): 779–782. [doi: 10.1109/12.599898]
- [2] Sorin DJ, Hill MD, Wood DA. A Primer on Memory Consistency and Cache Coherence. Cham: Springer, 2011. [doi: 10.1007/978-3-031-01733-9]
- [3] Sewell P, Sarkar S, Owens S, Nardelli FZ, Myreen MO. x86-TSO: A rigorous and usable programmer's model for x86 multiprocessors. Communications of the ACM, 2010, 53(7): 89–97. [doi: 10.1145/1785414.1785443]
- [4] Wang C, Lü Y, Wu P, Jia QW. Decidability of bounded linearizability on TSO memory model. Ruan Jian Xue Bao/Journal of Software, 2022, 33(8): 2896–2917 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/6604.htm> [doi: 10.13328/j.cnki.jos.006604]
- [5] Sarkar S, Sewell P, Alglave J, Maranget L, Williams D. Understanding POWER multiprocessors. In: Proc. of the 32nd ACM SIGPLAN Conf. on Programming Language Design and Implementation. San Jose: ACM, 2011. 175–186. [doi: 10.1145/1993498.1993520]
- [6] Alglave J, Deacon W, Grisenthwaite R, Hacquard A, Maranget L. Armed cats: Formal concurrency modelling at ARM. ACM Trans. on Programming Languages and Systems (TOPLAS), 2021, 43(2): 8. [doi: 10.1145/3458926]
- [7] Alglave J, Maranget L, Tautschnig M. Herding cats: Modelling, simulation, testing, and data mining for weak memory. ACM Trans. on Programming Languages and Systems (TOPLAS), 2014, 36(2): 7. [doi: 10.1145/2627752]
- [8] Alglave J, Cousot P. Ogre and Pythia: An invariance proof method for weak consistency models. In: Proc. of the 44th ACM SIGPLAN Symp. on Principles of Programming Languages. Paris: ACM, 2017. 3–18. [doi: 10.1145/3009837.3009883]
- [9] Wang J, Zhan NJ, Feng XY, Liu ZM. Overview of formal methods. Ruan Jian Xue Bao/Journal of Software, 2019, 30(1): 33–61 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/5652.htm> [doi: 10.13328/j.cnki.jos.005652]
- [10] Alglave J. A shared memory poetics [Ph.D. Thesis]. Paris: Université Paris, 2010.
- [11] Alglave J, Maranget L, McKenney PE, Parri A, Stern A. Frightening small children and disconcerting grown-ups: Concurrency in the Linux kernel. In: Proc. of the 23rd Int'l Conf. on Architectural Support for Programming Languages and Operating Systems. Williamsburg: ACM, 2018. 405–418. [doi: 10.1145/3173162.3177156]
- [12] Alglave J, Maranget L, Sarkar S, Sewell P. Litmus: Running tests against hardware. In: Proc. of the 17th Int'l Conf. on Tools and

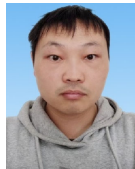
- Algorithms for the Construction and Analysis of Systems. Saarbrücken: Springer, 2011. 41–44. [doi: [10.1007/978-3-642-19835-9_5](https://doi.org/10.1007/978-3-642-19835-9_5)]
- [13] Waterman A, Asanović K, eds. The RISC-V instruction set manual, Vol. I: Unprivileged ISA. Version 20191214-draft, 2019. <https://riscv.github.io/riscv-isa-manual/snapshot/unprivileged/>
- [14] Liu C, Wu YJ, Wu JZ, Zhao C. Survey on RISC-V system architecture research. Ruan Jian Xue Bao/Journal of Software, 2021, 32(12): 3992–4024 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/6490.htm> [doi: [10.13328/j.cnki.jos.006490](https://doi.org/10.13328/j.cnki.jos.006490)]
- [15] Gharachorloo K, Lenoski D, Laudon J, Gibbons P, Gupta A, Hennessy J. Memory consistency and event ordering in scalable shared-memory multiprocessors. In: Sohi GS, ed. 25 Years of the Int'l Symposia on Computer Architecture (Selected Papers). New York: ACM, 1998. 376–387. [doi: [10.1145/285930.285997](https://doi.org/10.1145/285930.285997)]
- [16] Litmus-tests-riscv. 2022. <https://github.com/litmus-tests/litmus-tests-riscv>
- [17] Gray KE, Kernels G, Mulligan D, Pulte C, Sarkar S, Sewell P. An integrated concurrency and core-ISA architectural envelope definition, and test oracle, for IBM POWER multiprocessors. In: Proc. of the 48th Int'l Symp. on Microarchitecture. Waikiki: ACM, 2015. 635–646. [doi: [10.1145/2830772.2830775](https://doi.org/10.1145/2830772.2830775)]
- [18] Armstrong A, Bauereiss T, Campbell B, Reid A, Gray KE, Norton RM, Mundkur P, Wassell M, French J, Pulte C, Flur S, Stark I, Krishnaswami N, Sewell P. ISA Semantics for ARMv8-A, RISC-V, and CHERI-MIPS. In: Proc. of the 2019 ACM on Programming Languages. ACM, 2019. 71. [doi: [10.1145/3290384](https://doi.org/10.1145/3290384)]
- [19] Jackson D. Alloy: A language and tool for exploring software designs. Communications of the ACM, 2019, 62(9): 66–76. [doi: [10.1145/3338843](https://doi.org/10.1145/3338843)]
- [20] Hopcroft JE, Motwani R, Ullman JD. Introduction to Automata Theory, Languages, and Computation. 3rd ed., Boston: Addison-Wesley Longman Publishing Co. Inc., 2006.
- [21] Alglave J, Kroening D, Tautschnig M. Partial orders for efficient bounded model checking of concurrent software. In: Proc. of the 25th Int'l Conf. on Computer Aided Verification. Saint Petersburg: Springer, 2013. 141–157. [doi: [10.1007/978-3-642-39799-8_9](https://doi.org/10.1007/978-3-642-39799-8_9)]

附中文参考文献:

- [4] 王超, 吕毅, 吴鹏, 贾巧雯. TSO 内存模型下限界可线性化的可判定性研究. 软件学报, 2022, 33(8): 2896–2917. <http://www.jos.org.cn/1000-9825/6604.htm> [doi: [10.13328/j.cnki.jos.006604](https://doi.org/10.13328/j.cnki.jos.006604)]
- [9] 王戟, 詹乃军, 冯新宇, 刘志明. 形式化方法概貌. 软件学报, 2019, 30(1): 33–61. <http://www.jos.org.cn/1000-9825/5652.htm> [doi: [10.13328/j.cnki.jos.005652](https://doi.org/10.13328/j.cnki.jos.005652)]
- [14] 刘畅, 武延军, 吴敬征, 赵琛. RISC-V 指令集架构研究综述. 软件学报, 2021, 32(12): 3992–4024. <http://www.jos.org.cn/1000-9825/6490.htm> [doi: [10.13328/j.cnki.jos.006490](https://doi.org/10.13328/j.cnki.jos.006490)]



徐学政(1993—), 男, 博士, 助理研究员, 主要研究领域为程序分析, 系统软件.



王涛(1987—), 男, 博士, 副研究员, 主要研究领域为系统软件, 智能计算机系统.



杨德亨(1996—), 男, 博士, 助理研究员, 主要研究领域为软件工程, 程序分析.



黄安文(1983—), 男, 博士, 副研究员, 主要研究领域为计算机体系结构, 微处理器体系结构.



王璐(1991—), 女, 博士, 助理研究员, 主要研究领域为体系结构, 微处理器设计.



李琼(1967—), 女, 博士, 研究员, 博士生导师, 主要研究领域为高性能计算机系统结构, 智能计算, 信息存储.