

数据库管理系统中数据异常体系化定义与分类*

李海翔¹, 李晓燕², 刘畅¹, 杜小勇^{3,4}, 卢卫^{3,4}, 潘安群¹

¹(腾讯科技(北京)有限公司,北京 100080)

²(数学科学学院信息与计算科学系(北京大学),北京 100871)

³(数据工程与知识工程教育部重点实验室(中国人民大学),北京 100872)

⁴(中国人民大学 信息学院,北京 100872)

通讯作者: 李海翔, E-mail: blueseali@tencent.com



摘要: 数据异常尚没有统一的定义,其含义是指可能破坏数据库一致性状态的特定数据操作模式.已知的数据异常有脏写,脏读,不可重复读,幻读,丢失更新,读偏序和写偏序等.为了提高并发控制算法的效率,数据异常也被用于定义隔离级别,采用弱的隔离级别以提高事务处理系统的效率.本文体系化地研究了数据异常以及对应的隔离级别.发现了二十二种未被其它文献报告过的新的数据异常,并对全部数据异常进行分类.基于数据异常的分类提出了新的且不同粒度的隔离级别体系,揭示基于数据异常定义隔离级别的规律,使得对于数据异常和隔离级别等相关概念的认识可更加简明.

关键词: 事务处理;数据异常;隔离级别;并发访问控制算法;数据库

中图法分类号: TP311

中文引用格式: 李海翔,李晓燕,刘畅,杜小勇,卢卫,潘安群. 数据库管理系统中数据异常体系化定义与分类.软件学报. <http://www.jos.org.cn/1000-9825/6442.htm>

英文引用格式: Li HX, Li XY, Liu C, DU XY, Lu W, Pan AQ. Systematic definition and classification of data anomalies in DBMS. Ruan Jian Xue Bao/Journal of Software, 2021 (in Chinese). <http://www.jos.org.cn/1000-9825/6442.htm>

Systematic Definition and Classification of Data Anomalies in DBMS

LI Hai-Xiang¹, LI Xiao-Yan², LIU Chang¹, DU Xiao-Yong^{3,4}, LU Wei^{3,4}, Pan An-Qun¹

¹(Tencent Technology (Beijing) Co., Ltd, Beijing 100080, China)

²(Department of Information Sciences, School of Mathematical Sciences, Peking University, Beijing 100871, China)

³(Key Laboratory of Data Engineering and Knowledge Engineering of the Ministry of Education (Renmin University of China), Beijing 100872, China)

⁴(School of Information, Renmin University of China, Beijing 100872, China)

Abstract: There is no unified definition of Data anomalies, which refers to the specific data operation mode that may destroy the consistency of the database. Known data anomalies include Dirty Write, Dirty Read, Non-repeatable Read, Phantom, Read Skew and Write Skew, etc. In order to improve the efficiency of concurrency control algorithms, data anomalies are also used to define the isolation levels, because the weak isolation level can improve the efficiency of transaction processing systems. This paper systematically studies the data anomalies and the corresponding isolation levels. We report twenty-two new data anomalies that have not been reported by other papers, and all data anomalies are classified miraculously. Based on the classification of data anomalies, two new isolation levels system with different granularity are proposed, which reveals the rule of defining isolation levels based on data anomalies and makes the cognition of data anomalies and isolation levels more concise.

Key words: transaction processing; data anomalies; isolation levels; concurrency control algorithms; database

1 引言

数据库的隔离级别是通过消除特定的数据异常来定义的. ANSI-SQL 标准[1]描述了四种数据异常:脏写(Dirty Write),脏读(Dirty Read),不可重复读(Fuzzy or Non-repeatable Read),幻读(Phantom),相应地,其定义了四种隔离级别:读未提交,读已提交,可重复读,可串行化.例如读已提交可以消除脏写和脏读异常,最高的可串行化级别可以消除所有的数据异常.大多数数据库都支持了这四种隔离级别,如 DB2, Informix, MySQL, PostgreSQL, TDSQL 等.这些数据库在默认配置下,使用较弱的隔离级别,而较弱的隔离级别被认为可以提高数据库事务处理的吞吐量.

James Gray 等人[2]指出 ANSI-SQL 标准对数据异常的描述缺少形式化定义,造成对数据异常的理解存在偏差,因此基于形式化表达对上述四个数据异常进行了重新定义,并提出了四个新的数据异常:丢失更新(Lost Update 和 Cursor Lost Update),读偏序(Read Skew),写偏序(Write Skew). 截止到目前,本文通过对相关文献的详细调研,发现已被公开报告过的数据异常共有十八个

* 收稿时间: 2021-06-30; 修改时间: 2021-07-31; 采用时间: 2021-09-13; jos 在线出版时间: 2021-10-21

[1,2,3,4,5,6,7,8,9,10,11,12],表 1 所示.

表 1 现存数据异常汇总

编号	异常名称(异常名称,文献,年份)	形式化定义
1	Dirty Write,[1] 1992,[2] 1995	$W_1[x] \dots W_2[x] \dots (C_1 \text{ or } A_1) \text{ and } (C_2 \text{ or } A_2) \text{ in any order}$
2	Lost Update,[2] 1995	$R_1[x] \dots W_2[x] \dots W_1[x] \dots C_1$
3	Dirty Read,[1] 1992,[2] 1995	$W_1[x] \dots R_2[x] \dots (A_1 \text{ and } C_2 \text{ in either order})$
4	Aborted Reads,[3] 2000,[4] 2015	$W_1(x1:i) \dots R_2(x1:i) \dots (A_1 \text{ and } C_2 \text{ in any order})$
5	Fuzzy OR Non-Repeatable Read,[1] 1992	$R_1[x] \dots W_2[x] \dots C_2 \dots R_1[x] \dots C_1$
6	Phantom,[1] 1992	$R_1[P] \dots P_2[y \text{ in } P] \dots C_2 \dots R_1[P] \dots C_1$
7	Intermediate Reads,[3] 2000,[4] 2015	$W_1(x1:i) \dots R_2(x1:i) \dots W_1(x1:j) \dots C_2$
8	Read Skew,[2] 1995	$R_1[x] \dots W_2[x] \dots W_2[y] \dots C_2 \dots R_1[y] \dots (C_1 \text{ or } A_1)$
9	未命名的异常,[5] 2000	$R_3[y] R_1[x] W_1[x] R_1[y] W_1[y] C_1 R_2[x] W_2[x] R_2[z] W_2[z] C_2 R_3[z] C_3$
10	fractured reads,[6] 2014,[7] 2017	$R_1[x_0] \dots W_2[x_1] \dots W_2[y_1] \dots C_2 \dots R_1[y_1]$
11	Serial-Concurrent-Phenomenon,[8] 2014	$R_1[x_0] \dots W_2[x_1] \dots W_2[y_1] \dots C_2 \dots R_1[y_1]$
12	Cross-Phenomenon,[8] 2014	$R_1[x_0] \dots R_2[y_0] \dots W_3[x_1] \dots C_3 \dots W_4[y_1] C_4 \dots R_2[x_1] \dots R_1[y_1]$
13	long fork anomaly,[7] 2017	$R_4[x_0] \dots W_1[x_1] \dots R_3[y_0] \dots R_3[x_1] \dots W_2[y_1] \dots R_4[y_1]$
14	causality violation anomaly,[7] 2017	$R_3[x_0] \dots W_1[x_1] \dots C_1 \dots R_2[x_1] \dots W_2[y_1] \dots C_2 \dots R_3[y_1]$
15	A read-only transaction anomaly,[9] 1982,[10] 2004	$R_2(x_0,0) R_2(y_0,0) R_1(y_0,0) W_1(y_1,20) C_1 R_3(x_0,0) R_3(y_1,20) C_3 W_2(x_2,-11) C_2$
16	Write Skew,[2] 1995	$R_1[x] \dots R_2[y] \dots W_1[y] \dots W_2[x] \dots (C_1 \text{ and } C_2 \text{ occur})$
17	Predicate-Based Write Skew,[11] 2005	$R_1(\{x_0 \text{ in } P\}) \dots R_2(\{y_0 \text{ in } P\}) \dots W_1[\{y_1 \text{ in } P\}] \dots C_1 \dots W_2[\{x_1 \text{ in } P\}]$
18	读半已提交数据异常(分布式数据库),[12] 2019	$R_1[x] \dots W_2[x] \dots W_2[y] \dots C_2 \dots R_1[y] \dots C_1$

本文的主要贡献在于:体系化地研究了数据异常,用统一的方法定义了数据异常(并提出二十二种未被报告过的数据异常),并研究了数据异常的特征,对所有数据异常进行分类,基于数据异常的分类方法演示了如何基于数据异常定义不同粒度的隔离级别.本文第 2 节描述相关工作.第 3 节给出描述数据异常的框架,并对数据异常进行分类和形式化定义.第 4 节基于数据异常的分类定义隔离级别.第 5 节总结全文并致谢.

2 相关工作

过往研究,未曾有对数据异常进行系统化地研究.已知的文献研究提及数据异常的,主要有以下几个方面:

不断以案例的方式报告新的数据异常,缺乏体系化研究.数据异常一直没有被体系化地展开过研究.对于数据异常的认识,停留在了案例式的方式,如表 1 所示,2000 年,2005 年,2014 年,2017 年等都有新的数据异常被提出,表明数据异常的研究一直没有间断过.

ANSI-SQL 标准[1]提出的四种数据异常,都是单个变量构成的数据异常:[4,3]提出中间读(Intermediate Reads),回滚读(Abort Read)等异常也是只和单个变量相关的数据异常.[2]提出的读偏序和写偏序等异常涉及了两个变量造成的数据异常.[5]提出三个变量相关的类似读偏序的异常.[13]在 PostgreSQL 中讨论了更多形式的写偏序异常的变种.[9,10]描述了两个变量三个并发事务造成的异常(A read-only transaction anomaly).[9,10,11,8,7,6]提出的三个或四个事务构成的两变量异常,没有进一步研究更多变量是否会造成新数据异常.尽管数据异常看似和变量个数有关,但没有研究结果表明变量个数和异常之间的关系.

[8]提出了两种分布式数据异常(Serial-Concurrent-Phenomenon 和 Cross-Phenomenon).[7]提出与[8]相似的但独立命名的两种异常(Long Fork Anomaly 和 Causality Violation Anomaly).[6]提出与读偏序相似的异常(Fractured Reads).这些数据异常被提出,使得我们产生几个问题:分布式数据异常的本质是什么?其和非分布式数据异常之间是否存在关系?一些看似相似的数据异常为什么看似相似但不同的人给予不同的命名?

[11]描述了基于两个变量并违反约束的谓词写偏序(Predicate-Based Write Skew)等异常.

到目前为止:尚不知道数据异常究竟有多少?数据异常之间有什么本质的关系?

数据异常与可串行化理论.[14]给出可串行化的形式化定义,并发事务只有在满足可串行化的调度下,才能保证不出现数据异常.[15]利用可串行化的概念描绘了数据异常的定义,以作者有限的认知,这是首个也是唯一的对数据异常进行定义的文献.对于可串行化的求解技术,[16]指出基于 DSG(Direct Serialization Graph)图是一个 NP(non-deterministic polynomial-time)问题.因此,可串行化概念只具有验证一个调度是否能确保数据一致性[17,18]的价值而没有实践意义.之后[19,20]讨论了冲突可串行化.冲突可

串行化是可串行化的充分非必要条件.依据冲突可串行化的 Serialization Graph[19]/Precedence Graph[21]/Dependency Graph [2]/DSG[3,11]等技术具有指导事务调度引擎研发的实践价值,但其串行化思想融合在 2PL[14],T/O[23]等算法时因需尽可能确保可串行化而导致假回滚增高.本文依扩充了冲突关系和冲突可串行化图的概念,定义了偏序对和偏序环图,然后根据偏序环图进一步定义了数据异常的概念,从而奠定了本文的理论基础.

数据异常与隔离级别.[1,2,3]等都基于有限的数据异常定义了不同的,且隔离级别层次不同的隔离级别体系.但是,本文通过基于偏序对的有向环图推知数据异常有无数个,如果不能基于所有的异常来定义隔离级别,则会陷于以点代面式的从个例来认识全局的局限中.而本文通过对所有数据异常的进行形式化描述,给出了数据异常的定义和特征,并对数据异常进行分类,基于面向所有异常的数据异常分类体系而定义隔离级别,使得隔离级别的定义更加清晰,简明.

数据异常的量化与分类研究.现有的研究工作,是在已知的,有限的数据异常范围内(只包括表 1 中所示的部分已知数据异常),利用可串行化理论[22]和依赖图技术,展开对数据异常的量化与分类的工作.[23]开发了一套工具和方法,对源自应用的数据异常在快照隔离技术下进行了自动检测.[24,25,26,27]采用在应用和数据库中间,嵌入一层中间件的方式,使用特定场景如 TPC-C 对数据异常进行量化和分类,该方法不能也没有发现新的数据异常.[28]定量地研究了快照隔离技术下的数据异常在不同隔离级别时的完整性被违反的比率,但也没有发现新的数据异常.[3]基于 DSG 和一些数据异常个例,定义了一些类似数据异常的现象(phenomena),但没有进一步体系化地研究数据异常.本文则可以对所有数据异常加以定义和分类,并指出数据异常和隔离级别以及并发访问控制算法的关系等.

本文旨在体系化研究数据异常.本文体系化地研究了数据异常,用统一的方法定义了数据异常,并研究了数据异常的特征,对所有数据异常进行分类(并提出二十二类未被报告过的数据异常),并基于分类后的数据异常种类灵活定义不同形式的隔离级别.

3 数据异常的形式化定义与分类

并发控制机制就是要用正确的方式调度并发操作,使一个用户事务的执行不受其他事务的干扰,从而避免造成数据的不一致,即出现数据异常.

本节将对数据异常进行形式化定义.为了形式化地定义数据异常,我们先约定一些符号.

数据库中的数据对象,称为变量,记为 $D = \{x, y, z, \dots\}$. 一个变量可以是一个元组(tuple)或者记录(record),也可以是一个页面(page)甚至一个表(table)对象等.一个变量被写操作作用后会改变其状态,称为这个变量的一个版本,用下标表示一个变量的不同版本,记为 x_0, x_1, x_2 等.事务集表示为 $T = \{t_1, t_2, \dots, t_n\}$, 其中 $t_i = (op_i, <_s)$, 其中 op_i 表示事务 t_i 的操作和状态集合, $p_i <_s q_j$ 表示在调度 s 中操作 p_i 先于 q_j 发生,或在 q_j 之前处于状态 p_i .

3.1 基础定义

定义 1:操作(Operation)

一个事务在变量上的操作,包括读(Read,R),写(Write,W),用 $R_i[x_m]$ 和 $W_j[y_n]$ 分别表示事务 t_i 读变量 x 获得版本 x_m 和事务 t_j 写变量 y 形成版本 y_n , 其中读写操作符号的下标为该操作所属的事务号.

事务的状态包括提交(Commit,C),回滚(Abort,A)和未完成(Undone,U),用 C_i 表示事务 t_i 为提交状态,用 A_i 表示事务 t_i 为回滚状态,用 U_i 表示事务 t_i 为未完成状态,后面还将会事务 t_i 的操作,通常不会对未完成状态进行特别标记.

提交状态表示事务所读取到的数据项是合法的,所写的数据项也是合法的,提交是对本事务的合法状态进行确认.

回滚状态表示事务放弃所读取到的数据项,并撤销本事务所写的数据项的新值而恢复写操作之前对应的旧值(前像).

在无需区别 R, W, C, A 和 U 的时候,用 p 或 q 来代表一个操作或一种状态,即 $p_j = \{R_j[x_n], W_j[x_n], C_j, A_j \mid x \in D, n \in N, t_j \in T\}$, 其

中 T 为事务集,集合 D 为操作的变量集, n 表示变量的版本号.我们使用 $p_j[x_m]$ 表示事务 t_j 对版本为 m 的变量 x 进行操作,或用 p_j 简化表示事务 t_j 的一次操作或某种状态.

定义 2:调度(Schedule, s)

调度是指由一组事务的多个操作或状态组成的序列,记为 s .

调度 s 中所有的事务组成的集合记为 $T(s)$, 所有出现的变量组成的集合为 $D(s)$, 所有的操作组成的集合记为 $Op(s)$.

3.2 数据异常

定义 3:冲突和冲突关系(Conflicts and Conflicts Relation):

任意调度 s 中, $t_i, t_j \in T(s), t_i \neq t_j$, 在不考虑变量版本的情况下, 如果存在两操作 $p \in t_i$ 和 $q \in t_j$ 作用于同一个变量, 并且操作 p, q 中至少存在一个“写”操作, i.e. $(p = R[x] \wedge q = W[x]) \vee (p = W[x] \wedge q = R[x]) \vee (p = W[x] \wedge q = W[x])$, 称操作 p, q 为调度 s 中的一个冲突(Conflict).

将调度 s 中所有的冲突组成的集合称为调度 s 中的冲突关系, 记为

$$\text{conf}(s) = \{(p, q) \mid p, q \text{ 组成调度 } s \text{ 中的冲突, 并且 } p <_s q\}.$$

【例 1】 调度 $s = W_1[x_0]W_1[x_1]W_2[x_3]R_1[x_3]C_1C_2$ 的冲突关系为 $\text{conf}(s) = \{(W_1W_2[x]), (W_2R_1[x])\}$.

传统冲突关系中只考虑了“读”“写”两种操作组成的冲突, 实际中的冲突关系与事务的“提交”“回滚”也密切相关, 因此, 我们将传统冲突关系的定义中增加这两种操作, 细化调度 s 中的冲突关系.

定义 4: 蕴涵事务状态的冲突和冲突关系(Conflicts and Conflicts Relation with status):

任意调度 s 中, $t_i, t_j \in T(s), t_i \neq t_j$, 如果存在两操作 $p \in t_i$ 和 $q \in t_j$ 可构成冲突, 记为 (p_i, q_j) . 则冲突可分为以下七类:

- (1) $p_i - C_i - q_j - U_j / A_j / C_j$: 操作 p, q 中至少存在一个“写”操作, 并且 $p_i <_s C_i <_s q_j <_s U_j / A_j / C_j$
- (2) $p_i - A_i - q_j - U_j / A_j / C_j$: 操作 p, q 中至少存在一个“写”操作, 并且 $p_i <_s A_i <_s q_j <_s U_j / A_j / C_j$
- (3) $p_i - q_j - C_i - U_j / A_j / C_j$: 操作 p, q 中至少存在一个“写”操作, 并且 $p_i <_s q_j <_s C_i <_s U_j / A_j / C_j$
- (4) $p_i - q_j - A_i - U_j / A_j / C_j$: 操作 p, q 中至少存在一个“写”操作, 并且 $p_i <_s q_j <_s A_i <_s U_j / A_j / C_j$
- (5) $p_i - q_j - C_j - U_i / A_i / C_i$: 操作 p, q 中至少存在一个“写”操作, 并且 $p_i <_s q_j <_s C_j <_s U_i / A_i / C_i$
- (6) $p_i - q_j - A_j - U_i / A_i / C_i$: 操作 p, q 中至少存在一个“写”操作, 并且 $p_i <_s q_j <_s A_j <_s U_i / A_i / C_i$
- (7) $p_i - q_j = \{p_i - U_i - q_j - U_j / A_j / C_j \vee p_i - q_j - U_i - U_j / A_j / C_j \vee p_i - q_j - U_j - U_i / A_i / C_i\}$: 操作 p, q 中至少存在一个“写”操作, 并且两事务均未完成操作;

其中类别(1)-(4)为 t_i 在 t_j 之前提交或回滚, 即 $p_i <_s q_j$; 记为 (p_i, o_i, q_j) 或 (p_i, q_j, o_i) , $o \in \{A, C\}$; 类别(5)-(6)为 t_j 在 t_i 之前提交或回滚, 用 $p_i <_s q_j$ 表示, 记为 (p_i, q_j, o_j) , $o \in \{A, C\}$; 类别(7)表示调度 s 中两个事务均未完成, 即只有 $p_i <_s q_j$, 记为 (p_i, q_j) . 将上述类别称为蕴涵事务状态的冲突类别. 将调度 s 中所有蕴涵事务状态的冲突组成的集合称为调度 s 中蕴涵事务状态的冲突关系, 记为 $\text{conf}_{ac}(s)$, 即

$$\text{conf}_{ac}(s) = \{(p_i, o_i, q_j) \vee (p_i, q_j, o_i) \vee (p_i, q_j, o_j) \vee (p_i, q_j) \mid ((p_i, q_j) \in \text{conf}(s)) \wedge (o \in \{A, C\})\}.$$

定义 5: 蕴涵事务状态的冲突等价(Conflict with status)

两调度 s 和 s' 称为蕴涵事务状态的冲突等价, 记为 $s \approx_{\text{conf}_{ac}} s'$, 如果两调度有相同的操作, 同时在考虑事务状态下的冲突关系也相同. 即若 $s \approx_{\text{conf}_{ac}} s'$, 那么需要满足

- (1) $op(s) = op(s')$
- (2) $\text{conf}_{ac}(s) = \text{conf}_{ac}(s')$.

定义 6: 蕴涵事务状态的冲突图模型(Conflict with Status Graph, CCAG):

调度 s 蕴涵事务状态的冲突图模型 $G(s) = (V, E)$, 其中

- (1) $V = T(s)$,
- (2) $(p_i, q_j) \in E \wedge t_i \neq t_j \wedge (p_i, q_j) \in \text{conf}_{ac}(s)$

【例 2】 $s = R_1[x_0]R_1[y_0]W_2[y_1]W_3[z_1]W_1[z_2]C_2W_3[y_2]$ 的蕴涵事务状态的冲突关系为

$$conf_{ac}(s) = \{(R_1W_2[y]), (W_3W_1[z]), (R_1W_3[y]), (W_2C_2W_3[y])\};$$

该示例的蕴涵事务状态的冲突图如图 1:

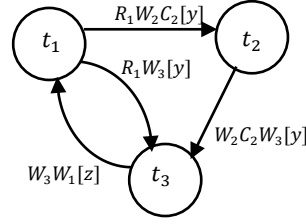


图 1 调度 s 的蕴涵事务状态的冲突图模型

定义 7:偏序对 (Partial order pair, Pop)

在调度 s 中蕴涵事务状态的冲突关系中的操作组合中,第(2)类操作中由于 t_i 的及时回滚,操作 p_i 不会对操作 q_j 产生影响.第(6)类操作组合同理.将 $(p, q) \in \{WW, WR, RW\}$ 三种操作组合代入剩余 5 种情况中可得 15 种情况.

我们将 15 种情形分为 12 类:

- (1) $W_iC_iR_j = \{W_iC_iR_j\}$: t_i 所写的且已经确认有效的版本,被 t_j 读取;
- (2) $W_iC_iW_j = \{W_iC_iW_j\}$: t_i 所写的版本,被 t_j 用更新的版本覆盖,但是合法覆盖不会造成数据状态不一致;
- (3) $R_iC_iW_j = \{R_iC_iW_j\}$: t_i 所读的版本,被 t_j 用更新的版本覆盖,同时相邻的操作组合中 t_i 需要有写操作;
- (4) $W_iW_j = \{W_iW_j, W_iW_jC_j\}$: t_i 所写的版本,被 t_j 用更新的版本覆盖,使得可能存在数据状态不一致;
- (5) $W_iR_j = \{W_iR_j, W_iR_jC_j\}$: t_i 所写的版本,被 t_j 读取;
- (6) $R_iW_j = \{R_iW_j, R_iW_jC_j\}$: t_i 读取的版本,被 t_j 修改后生成新版本,可能影响 t_i 读取或修改同样的变量;
- (7) $W_iR_jA_i = \{W_iR_jA_i\}$: t_j 读取的版本是 t_i 所写,被 t_i 回滚后可能使得 t_j 读取了一个不存在的版本;
- (8) $W_iW_jC_i = \{W_iW_jC_i\}$: t_i 所写的版本,被 t_j 使用更新的版本覆盖,使得 C_i 发生后不能读取到 t_j 所写的数据项的值.
- (9) $W_iW_jA_i = \{W_iW_jA_i\}$: t_j 所写的版本,被 t_i 因回滚而使用更旧的版本覆盖.
- (10) $R_iW_jC_i = \{R_iW_jC_i\}$: t_i 所读的版本,被 t_j 用更新的版本覆盖后 t_i 提交;
- (11) $W_iR_jC_i = \{W_iR_jC_i\}$: t_i 所写的版本,被 t_j 读到后 t_i 提交;
- (12) $R_iW_jA_i = \{R_iW_jA_i\}$: t_i 所读的版本,被 t_j 用更新的版本覆盖后 t_i 提交;

显然,第(7)-(12)类操作组合本身构成环,并且第(10)类事务 t_i 的提交操作并未对事务 t_j 的写操作产生影响,因此语义与 R_iW_j 相同;第(11)类 $W_iR_jC_i$ 同理,事务 t_i 的提交操作即事务 t_j 读操作的版本,因此语义与 W_iR_j 相同;第(12)类中,事务 t_i 的回滚操作也不影响事务 t_j 的写操作,因此语义与 R_iW_j 相同.

但(7)-(9)产生事务 t_j 操作无效的结果(其中 $W_iW_jC_i$ 和 $W_iW_jA_i$ 构成异常附录 1),不失一般性,我们将这三类进行细化:

第(7)类中,我们可以将 $\{W_iR_jA_i\}$ 拆为 $\{W_iR_j[x]\}$ 和 $\{R_jA_i[x]\}$,即第(4)类和 $\{R_jA_i[x]\}$ 在同一变量 x 下的操作组合,因此第(7)类可简记为 R_jA_i ,可表示为图 2.

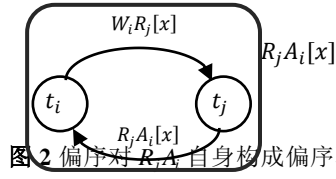


图 2 偏序对 $R_i A_i$ 自身构成偏序环

第(8)类中,我们可以将 $\{W_i W_j C_i[x]\}$ 拆为 $\{W_i W_j[x]\}$ 和 $\{W_j C_i[x]\}$,即第(3)类和 $\{W_j C_i[x]\}$ 在同一变量 x 下的操作组合,因此第(8)类可简记为 $W_j C_i$,可表示为图 3.

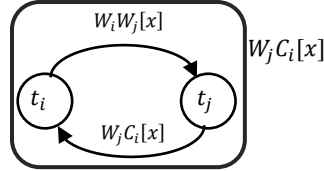


图 3 偏序对 $W_j C_i$ 自身构成偏序环

第(9)类中,我们可以将 $\{W_i W_j A_i\}$ 拆为 $\{W_i W_j[x]\}$ 和 $\{W_j A_i[x]\}$,即第(3)类和 $\{W_j A_i[x]\}$ 在同一变量 x 下的操作组合,因此第(9)类可简记为 $W_j A_i$,可表示为图 4.

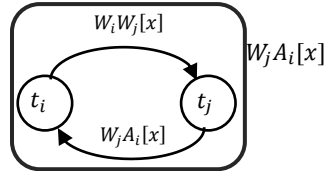


图 4 偏序对 $W_j A_i$ 自身构成偏序环

称(1)-(9)的 9 类操作组合为偏序对,对调度 s 中的变量 $\forall x \in D(s), \forall p_i <_s q_j$,记为 $p_{ij}(s, x)$ 或 $p_{ij}[x]$.

调度 s 中所有偏序对组成的集合表示为 $Pop(s) = \{p_{ij}(s, x) \mid x \in D(s) \wedge t_i <_s t_j\}$.

全部偏序对的集合表示为 $Pop = \{p(s, x) \mid s \text{ 为任意调度} \wedge x \in D(s)\}$.

定义 8:偏序对图(Partial order pair Graph, PG)

根据上述的对应关系将蕴涵事务状态的冲突图模型简化后得到的图模型为**偏序对图模型**,记为 $PG = (V, E)$,其中

- (1) $V = T(s)$;
- (2) $(t_i, t_j) \in E \wedge t_i \neq t_j \wedge \forall x \in D$

$$p_{ij}[x] \in \{W_i C_i R_j, W_i C_i W_j, R_i C_i W_j, W_i W_j, W_i R_j, R_i W_j, W_i R_j A_i, W_i W_j C_i, W_i W_j A_i\}$$

【例 3】对调度 $s = W_1[x]R_2[x]A_1W_2[y]C_2R_3[y]$,可得

$$Pop(s) = \{(W_1 R_2 A_1[x]), (W_2 C_2 R_3[y])\}$$

因此,调度 s 对应的偏序对图为图 5.

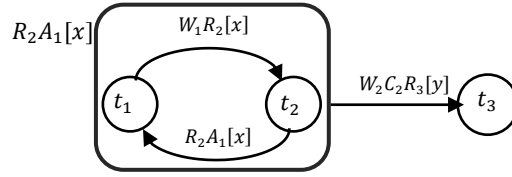


图5 调度 s 的偏序对图

定义 9:偏序对等价

两调度 s 和 s' 称为偏序对等价,记为 $s \approx_{PG} s'$,如果两调度有相同的操作,并且两调度所对应的偏序对图相同.即若 $s \approx_{PG} s'$,需要满足

$$(1) \quad op(s) = op(s'),$$

$$(2) \quad E(s) = E(s').$$

定义 10:数据异常(Data Anomalies, DA)

若调度 s 称为存在数据异常,如果存在调度 s' ,其偏序对图中存在闭环(包含偏序对自身组成的环),并且 $s \approx_{PG} s'$.

偏序对图中的有向闭环简称**偏序环(Partial order pair circle graph, PCG)**,记为 $PCG(s) = (V, E)$.

根据偏序环的定义,偏序环中至少含有两个事务,同时构成有向环,因此至少有两条边.显然,调度 s 中的 $T(s)$ 可以有无数个,使得 PCG 中可以有无数顶点,可存在无数条边.由偏序环图和数据异常的等价关系,调度集中可能出现的数据异常有无数个.因此对无穷多的数据异常进行分类具有重要的意义.偏序环图可根据边的种类进行分类,那么我们就可以将数据异常根据偏序环图的类别进行分类,从而达到了数据异常分类的目的.

在实际调度中,我们将偏序环图中边数最少最先出现的偏序环,作为调度 s 中区分异常的偏序环.若同时存在多个偏序环按事务在调度中的操作顺序排列,取最先进行操作的一环作为该调度中的偏序环.

【例 4】 调度 $s_1 = R_1[x_0]W_2[x_i]W_2[y_j]W_3[z]R_4[z]R_4[x_j]W_4[x_p]$,偏序对组成的集合为

$$Pop(s) = \{(R_1W_2[x]), (R_1W_4[x]), (W_2R_3[x]), (W_2W_4[x]), (W_2W_3[y]), (W_3R_1[z])\}.$$

偏序对图为图 6.

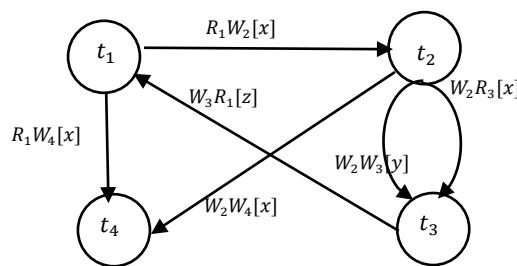


图6 调度 s_1 的偏序对图

调度 s 中选择边数最少且最先出现的偏序环为 $G = (\{t_1, t_2, t_3\}, \{(R_1W_2[x]), (W_2R_3[x]), (W_3R_1[z])\})$.

定理 10.1:若 3 事务单变量构成偏序环,则一定存在 2 事务的偏序子环.

Proof:

不妨设 3 事务单变量构成的偏序环 $G = (\{t_1, t_2, t_3\}, \{(p_1q_2), (p_2q_3), (p_3q_1)\})$.

若偏序关系中存在偏序对 $\{RA, WA, WC\}$,本身构成 2 事务最小环.

由于偏序关系中必然存在写操作,同时调度中的操作顺序也不确定,因此我们做如下讨论.

- (1) 当 $p_1 = W_1$ 时, 无论 p_2 为读或写操作均构成偏序关系, 由于操作顺序不确定性, 偏序关系有 $(p_1 p_2)$ 和 $(p_2 p_1)$ 两种. 若为 $(p_1 p_2)$, 由于存在偏序关系 $(p_2 q_3)$, 则一定有操作顺序 $p_1 <_s p_2 <_s q_3$, 因此构成偏序关系 $(p_1 q_3)$, 与 $(p_3 q_1)$ 构成 2 事务环; 若为 $(p_2 p_1)$, 则与已存在的偏序 $(p_1 q_2)$ 构成 2 事务环.
- (2) 当 $p_1 = R_1$ 时, 由于存在偏序关系 $(p_1 q_2)$, 那么一定有 $q_2 = W_2$, 因此无论 p_3 为读或写操作均构成偏序关系, 由于操作顺序不确定性, 偏序关系有 $(q_2 p_3)$ 和 $(p_3 q_2)$ 两种. 若为 $(q_2 p_3)$, 由于存在偏序关系 $(p_3 q_1)$, 则一定有操作顺序 $q_2 <_s p_3 <_s q_1$, 因此构成偏序关系 $(p_1 q_2)$, 与 $(q_2 q_1)$ 构成 2 事务环; 若为 $(p_3 q_2)$, 则与已存在的偏序 $(p_2 q_3)$ 构成 2 事务环.

【例 5】3 事务偏序环可规约为 2 事务偏序环, 约简过程如图 7.

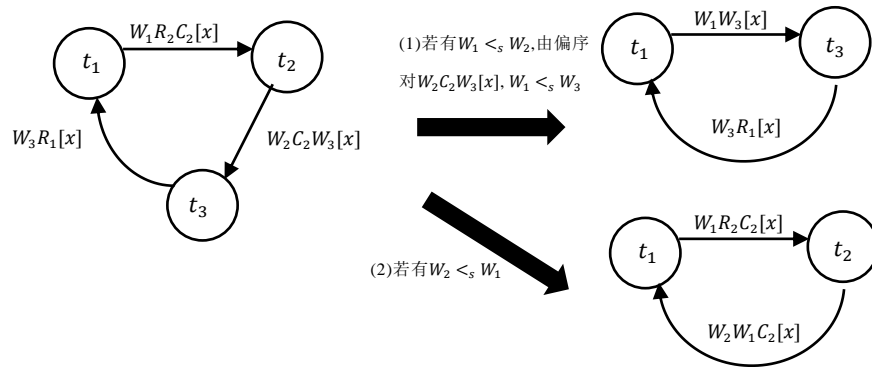


图 7 3 事务单变量偏序环可规约为 2 事务单变量偏序环

定理 10.2: 单变量的多事务(≥ 3)的偏序环可规约为 2 事务偏序环.

Proof:

不妨设单变量多事务偏序环 $G=(V,E)$, 顶点集为 $V=\{t_1, t_2, \dots, t_n\}, n \geq 4, V \in T(s)$. 边集为

$E=\{(p_1 q_2), (p_2 q_3), (p_3 q_4), \dots, (p_{n-1} q_n), (p_n q_1)\}$. 使用数学归纳法证明.

由定理 10.1, 当 $N_i = 3$ 时成立.

假设当 $N_i < n$ 时命题成立. 即当 $E=(p_1 q_2), (p_2 q_3), (p_3 q_4), \dots, (p_{k-1} q_k), (p_k q_1), k < n$ 时, 可规约为 2 事务环 $(p_k q_k)(p_k q_s)$.

当 $N_i = n$ 时, $E(n)=(p_1 q_2), (p_2 q_3), (p_3 q_4), \dots, (p_{n-1} q_n), (p_n q_1)$,

- (1) 当 $p_1 = W_1$ 时, 无论 p_{n-1} 为读或写操作均构成偏序关系, 由于操作顺序不确定性, 偏序关系有 $(p_1 p_{n-1})$ 和 $(p_{n-1} p_1)$ 两种. 若为 $(p_1 p_{n-1})$, 由于存在偏序关系 $(p_{n-1} q_n)$, 则一定有操作顺序 $p_1 <_s p_{n-1} <_s q_n$, 因此构成偏序关系 $(p_1 q_n)$, 与 $(p_n q_1)$ 构成 2 事务环; 若为 $(p_{n-1} p_1)$, 则与已存在的偏序 $(p_1 q_2)$ 构成偏序 $(p_{n-1} q_2)$, 此时与已有偏序关系构成 $N_i = n - 2$ 的偏序环 $E=(p_2 q_3), (p_3 q_4), \dots, (p_{n-1} q_2)$, 由假设, 可规约为 2 事务环.
- (2) 当 $p_1 = R_1$ 时, 由于存在偏序关系 $(p_1 q_2)$, 那么一定有 $q_2 = W_2$, 因此无论 p_n 为读或写操作均构成偏序关系, 由于操作顺序不确定性, 偏序关系有 $(q_2 p_n)$ 和 $(p_n q_2)$ 两种. 若为 $(q_2 p_n)$, 由于存在偏序关系 $(p_n q_1)$, 则一定有操作顺序 $q_2 <_s p_n <_s q_1$, 因此构

成偏序关系 (q_2q_1) , 与 (p_1q_2) 构成 2 事务环; 若为 (p_nq_2) , 此时与已有偏序关系构成 $N_i = n-1$ 的偏序环 $E = (p_2q_3), (p_3q_4), \dots, (p_nq_2)$, 由假设, 可规约为 2 事务环。

【例 6】5 事务偏序环可规约为 2 事务偏序环, 约简过程如图 8。

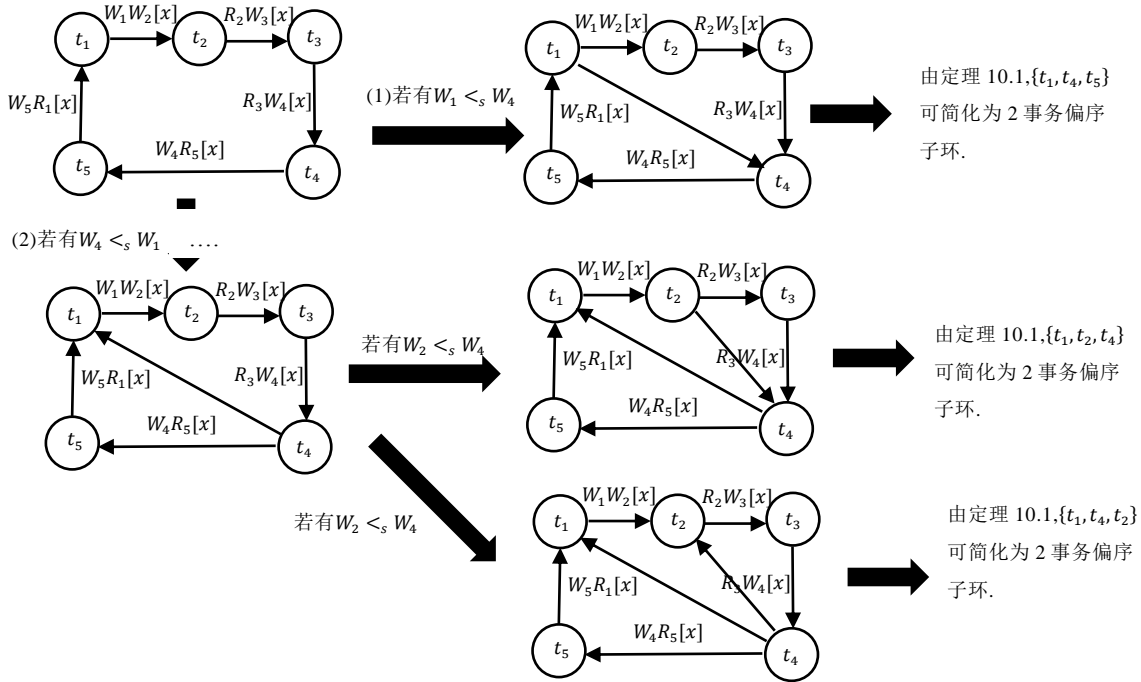


图 8 5 事务单变量偏序环可规约为 2 事务单变量偏序环

表 2 符号及含义

符号	描述
s	调度
$p_i <_s q_j$	调度 s 中操作 p_i 在操作 q_j 之前
$op_i(s)$	调度 s 中事务 t_i 的操作集合
$p_{ij}[x] = (p_i p_j[x])$	调度 s 中事务 t_i 对事务 t_j 同时操作变量 x 时产生的偏序对集合
$W_i[x_m]$	事务 t_i 的写操作 W 写数据项 x 的第 m 个版本
$R_i[x_n]$	事务 t_i 的读操作 R 读数据项 x 的第 n 个版本
$T(s)$	调度 s 中事务的集合
$Pop(s)$	调度 s 中的偏序对组成的集合
$D(s)$	调度 s 中所有变量组成的集合
Pop	全部偏序对组成的集合
PCG	偏序环图
N_D	调度 s 中变量的个数
N_T	调度 s 中事务的个数
$C_{da} = (N_D, N_T)$	多事务异常
WAT	写异常
RAT	读异常
IAT	交叉异常
SDA	单元异常
DDA	二元异常
MDA	多元异常

本文所使用的基本符号, 参见表 2。

表 3 单变量双边偏序环组合

编号	名称	形式化表达	偏序环
(1)	Dirty Write	$W_i[x_m] \dots W_j[x_{m+1}] \dots A_i / C_i$	$W_i W_j - W_j A_i$
			$W_i W_j - W_j C_i$
			$W_i W_j C_i - R_j W_i C_i$
			$W_i W_j C_i - R_j W_i C_i$
			$W_i W_j C_i - W_j W_i C_i$
(2)	Dirty Read	$W_i[x_m] \dots R_j[x_m] \dots A_i$	$W_i R_j - R_j A_i$
			$W_i W_j A_j - R_j W_i A_j$
			$R_i W_j A_j - R_j W_i A_j$
			$W_i W_j C_j - W_j C_j R_i$
			$W_i R_j C_j - W_j C_j R_i$
(3)	Lost Self Update Committed	$W_i[x_m] \dots W_j[x_{m+1}] \dots C_j \dots R_i[x_{m+1}]$	$W_i W_j C_j - W_j C_j R_i$
			$W_i R_j C_j - W_j C_j R_i$
			$W_i W_j C_j - W_j C_j W_i$
			$W_i R_j C_j - W_j C_j W_i$
			$W_i W_j C_j - R_j C_j W_i$
(4)	Full-Write Committed	$W_i[x_m] \dots W_j[x_{m+1}] \dots C_j \dots W_i[x_{m+2}]$	$W_i W_j C_j - W_j C_j W_i$
			$W_i R_j C_j - W_j C_j W_i$
			$W_i W_j C_j - R_j C_j W_i$
			$W_i W_j C_j - R_j C_j W_i$
			$W_i W_j C_j - R_j C_j W_i$
(5)	Non-repeatable Read Committed	$R_i[x_m] \dots W_j[x_{m+1}] \dots C_j \dots R_i[x_{m+1}]$	$R_i W_j C_j - W_j C_j R_i$
			$R_i W_j C_j - W_j C_j W_i$
			$R_i R_j C_j - W_j C_j W_i$
			$R_i W_j C_j - R_j C_j W_i$
			$R_i W_j C_j - R_j C_j W_i$
(6)	Lost Update Committed	$R_i[x_m] \dots W_j[x_{m+1}] \dots C_j \dots W_i[x_{m+2}]$	$R_i W_j C_j - W_j C_j W_i$
			$R_i W_j C_j - R_j C_j W_i$
			$R_i W_j C_j - R_j C_j W_i$
			$W_i W_j - W_j W_i$
			$W_i R_j - W_j W_i$
(7)	Full Write	$W_i[x_m] \dots W_j[x_{m+1}] \dots W_i[x_{m+2}]$	$W_i W_j - R_j W_i$
			$W_i R_j C_i - W_j W_i C_i$
			$W_i R_j C_j - R_j C_j W_i$
			$R_i W_j - W_j W_i$
			$R_i W_j - R_j W_i$
(8)	Lost Update	$R_i[x_m] \dots W_j[x_{m+1}] \dots W_i[x_{m+2}]$	$R_i W_j C_i - W_j W_i C_i$
			$R_i W_j C_i - R_j W_i C_i$
			$R_i W_j C_j - R_j W_i C_j$
			$W_i W_j - W_j R_i$
			$W_i R_j - W_j R_i$
(9)	Lost Self Update	$W_i[x_m] \dots W_j[x_{m+1}] \dots R_i[x_{m+1}]$	$W_i W_j C_j - W_j R_i C_j$
			$W_i R_j C_j - W_j R_i C_j$
			$R_i W_j - W_j R_i$
			$R_i W_j C_j - W_j R_i C_j$
			$R_i W_j C_i - W_j R_i C_i$
(10)	Non-repeatable Read	$R_i[x_m] \dots W_j[x_{m+1}] \dots R_i[x_{m+1}]$	$R_i W_j C_j - W_j R_i C_j$
			$R_i W_j C_i - W_j R_i C_i$
			$W_i R_j - R_j W_i$
			$W_i R_j C_i - R_j W_i C_i$
			$W_i R_j C_i - R_j W_i C_i$
(11)	Intermediate Read	$W_i[x_m] \dots R_j[x_m] \dots W_i[x_{m+1}]$	$W_i R_j - R_j W_i$
			$W_i R_j C_i - R_j W_i C_i$
			$W_i R_j C_i - R_j W_i C_i$
			$W_i R_j C_i - R_j W_i C_i$
			$W_i R_j C_i - R_j W_i C_i$

3.3 数据异常的分类

显然,由于调度中事务数,变量和操作的复杂性,可能发生的数据异常有无数种.弄清无数种数据异常之间的关系和相近特性,有利于认识数据异常的多样性,并挖掘数据异常间内在的规律,进而形成条理清晰的分类体系,并能对每一类数据异常的特征进行科学的描述,以弄清不同类数据异常之间的关联关系.分类后的结果表明,分类除了有利于体系化地认知数据异常,还有助于定义隔离级别(第4节),并对并发访问控制算法设计形成新的认知,基于新的认知,本文提出:第一,基于全部数据异常的分类方法,可完备定义隔离级别;第二,隔离级别可以有多种不同形式的定义方式;第三,基于简化的隔离级别的定义,有助于简化并发访问控制算法的实现(不必区分多种情况).复杂的隔离级别的定义,因有更多的级别而有助于认知数据异常分类之间细微的差异.本质上,隔离级别就是对数据异常的另外一种分类方式.

根据定义10,数据异常和偏序环密切相关.因此,本节我们旨在通过分析偏序环的种类对数据异常进行类别划分.

3.3.1 单变量二事务数据异常的分类

根据定理 10.1 和定理 10.2 可知,单变量的偏序环最终只会归结到两事务环中.换句话说,单变量的数据异常最终会归结到二事务环操作中.因此,我们先对二事务偏序环做类别划分.显然,当调度 s 中所有偏序对组成的集合 $Pop(s)$ 中包含 $\{W_iW_jA_i, W_iW_jC_i, R_iW_jA_i\}$ 三种偏序对时,本身即为偏序环,因此可将这三组偏序对直接对应一个二事务偏序环(见附录 2 异常(1)-(2)).当不包含这三种偏序时,二事务单变量偏序环 $PCG(s) = (\{t_i, t_j\}, \{p_{ij}[x], p_{ji}[x]\})$ 可组成的数据异常形式有 9 种(见附录 2 异常(3)-(11)),将单变量异常形式总结为表 3.

通过对二事务单变量的操作遍历可得表中 11 种偏序环双边的组合情况.显然第(1)(3)(4)(7)(8)(9)这六种数据异常的形式化表达中存在 $W_{i/j}[x_m] \dots W_{j/i}[x_{m+1}]$; 第(2)(10)(11)的数据异常的形式化表达中不包含 $W_{i/j}[x_m] \dots W_{j/i}[x_{m+1}]$ 但是包含先写后读的操作 $W_i[x_m] \dots R_j[x_m]$ 或 $W_j[x_{m+1}] \dots R_i[x_{m+1}]$; 第(5)种数据异常通过 C_j 将 $W_j[x_{m+1}] \dots R_i[x_{m+1}]$ 分隔开来,第(6)种数据异常通过 C_j 将 $W_{i/j}[x_m] \dots W_{j/i}[x_{m+1}]$ 分隔开来.

根据对单变量数据异常的分析,我们可以通过对任意事务数和变量数的偏序环中边的偏序对操作对数据异常进行分类.对偏序环 $PCG = (V, E)$,若该环中的变量集合表示为 $D_{PCG}(s)$

1) **写异常类型(Write Anomaly Type, WAT)**:偏序环形式化表达中包含 $W_{i/j}[x_m] \dots W_{j/i}[x_{m+1}]$.

即 $E_{WAT} = \{PCG(V, E) \mid \text{形式化表达中包含 } W_i[x_m] \dots W_j[x_{m+1}] \text{ 或 } W_j[x_m] \dots W_i[x_{m+1}]\}$.

特别的,当 $D(s) = \{x\}$ 时,数据异常(1)(3)(4)(7)(8)(9) $\in E_{WAT}[x]$; 将其称为写异常下的单变量二事务异常(Single-variable by Double-transactions Anomalies, SDA,简称单元异常)记为 WAT-SDA.

2) **读异常类型(Read Anomaly Type, RAT)**:偏序环形式化表达中不包含 $W_{i/j}[x_m] \dots W_{j/i}[x_{m+1}]$,且包含 $W_{i/j}[x_m] \dots R_{j/i}[x_m]$;

即 $E_{RAT} = \{PCG(V, E) \mid \text{形式化表达中不包含 } W_{i/j}[x_m] \dots W_{j/i}[x_{m+1}] \wedge \text{形式化表达中包含 } W_{i/j}[x_m] \dots R_{j/i}[x_m]\}$.

特别的,当 $D(s) = \{x\}$ 时,数据异常(2)(10)(11) $\in E_{RAT}[x]$; 将其称为读异常下的二事务单变量异常(SDA),记为 RAT-SDA.

3) **交叉异常类型(Intersect Anomaly Type, IAT)**:偏序环形式化表达中上面两种情况均不包含的情况.

即 $E_{IAT} = \{PCG(V, E) \mid \text{形式化表达中不包含 } W_{i/j}[x_m] \dots W_{j/i}[x_{m+1}] \wedge \text{形式化表达中不包含 } W_{i/j}[x_m] \dots R_{j/i}[x_m]\}$.

特别的,当 $D_{PCG}(s) = \{x\}$ 时,数据异常(5)(6) $\in E_{RAT}[x]$; 将其称为交叉异常下的二事务单变量异常(SDA),记为 IAT-SDA.

3.3.2 双变量二事务数据异常的分类

显然对任意变量数我们均可画出偏序环,并简化为形式化表达,获取数据异常的类别.本节将对双变量二事务数据异常按照分类的标准进行划分.

仍以二事务双边环做遍历分析.不妨设二事务双变量偏序环 $PCG(s) = (\{t_i, t_j\}, \{p_{ij}[x], p_{ji}[x]\})$,由 $t_i <_s t_j$,可组成的数据异常形式有 15 种(见附录 2 异常(12)-(26)),将上述二事务双变量偏序环组成的数据异常总结如表 4.

根据对数据异常的一级分类规则,我们可以得出双变量二事务类别:

第(12)(13)(18)-(21)(24)(25)种异常为写异常,即(12)(13)(18)-(21)(24)(25) $\in E_{WAT}$; 将其称为写异常下的二事务双变量异常(Double-variables by Double-transactions Anomalies, DDA,简称双元异常),记为 WAT-DDA.

表 4 双变量双边偏序环组合

编号	名称	形式化表达	偏序环
----	----	-------	-----

(12)	Double-Write Skew2 Committed	$W_i[x_m] \dots W_j[x_{m+1}] \dots W_j[y_n] \dots C_j \dots R_i[y_n]$	$W_i W_j C_j[x] - W_j C_j R_i[y]$
(13)	Full-Write Skew Committed	$W_i[x_m] \dots W_j[x_{m+1}] \dots W_j[y_n] \dots C_j \dots W_i[y_{n+1}]$	$W_i W_j C_j[x] - W_j C_j W_i[y]$
(14)	Write-Read Skew Committed	$W_i[x_m] \dots R_j[x_m] \dots W_j[y_n] \dots C_j \dots R_i[y_n]$	$W_i R_j C_j[x] - W_j C_j R_i[y]$
(15)	Double-Write Skew1 Committed	$W_i[x_m] \dots R_j[x_m] \dots W_j[y_n] \dots C_j \dots W_i[y_{n+1}]$	$W_i R_j C_j[x] - W_j C_j W_i[y]$
(16)	Read Skew Committed	$R_i[x_m] \dots W_j[x_{m+1}] \dots W_j[y_n] \dots C_j \dots R_i[y_n]$	$R_i W_j C_j[x] - W_j C_j R_i[y]$
(17)	Read-Write Skew1 Committed	$R_i[x_m] \dots W_j[x_{m+1}] \dots W_j[y_n] \dots C_j \dots W_i[y_{n+1}]$	$R_i W_j C_j[x] - W_j C_j W_i[y]$
(18)	Full Write Skew	$W_i[x_m] \dots W_j[x_{m+1}] \dots W_j[y_n] \dots W_i[y_{n+1}]$	$W_i W_j[x] - W_j W_i[y]$
(19)	Double-Write Skew1	$W_i[x_m] \dots R_j[x_m] \dots W_j[y_n] \dots W_i[y_{n+1}]$	$W_i R_j[x] - W_j W_i[y]$ $W_i R_j C_i[x] - W_j W_i C_i[y]$
(20)	Read-Write Skew1	$R_i[x_m] \dots W_j[x_{m+1}] \dots W_j[y_n] \dots W_i[y_{n+1}]$	$R_i W_j[x] - W_j W_i[y]$ $R_i W_j C_i[x] - W_j W_i C_i[y]$
(21)	Double-Write Skew2	$W_i[x_m] \dots W_j[x_{m+1}] \dots W_j[y_n] \dots R_i[y_{n+1}]$	$W_i W_j[x] - W_j R_i[y]$ $W_i W_j C_j[x] - W_j R_i C_j[y]$
(22)	Write-Read Skew	$W_i[x_m] \dots R_i[x_m] \dots W_j[y_n] \dots R_i[y_n]$	$W_i R_j[x] - W_j R_i[y]$ $W_i R_j C_i[x] - W_j R_i C_i[y]$ $W_i R_j C_j[x] - W_j R_i C_j[y]$
(23)	Read Skew	$R_i[x_m] \dots W_j[x_{m+1}] \dots W_j[y_n] \dots R_i[y_n]$	$R_i W_j[x] - W_j R_i[y]$ $R_i W_j C_j[x] - W_j R_i C_j[y]$ $R_i W_j C_i[x] - W_j R_i C_i[y]$ $W_i W_j[x] - R_j W_i[y]$
(24)	Read-Write Skew2	$W_i[x_m] \dots W_j[x_{m+1}] \dots R_j[y_n] \dots W_i[y_{n+1}]$	$W_i W_j C_j[x] - R_j C_j W_i[y]$ $W_i W_j C_j[x] - R_j W_j C_j[y]$ $W_i W_j A_j[x] - R_j W_i A_j[y]$
(25)	Read Skew2	$W_i[x_m] \dots R_j[x_m] \dots R_j[y_n] \dots W_i[y_{n+1}]$	$W_i R_j[x] - R_j W_i[y]$ $W_i R_j C_j[x] - R_j W_j C_j[y]$ $R_i W_j[x] - R_j W_i[y]$
(26)	Write Skew	$R_i[x_m] \dots W_j[x_{m+1}] \dots R_j[y_n] \dots W_i[y_{n+1}]$	$R_i W_j C_j[x] - R_j C_j W_i[y]$ $R_i W_j C_j[x] - R_j W_j C_j[y]$ $R_i W_j A_j[x] - R_j W_i A_j[y]$

第(14)(15)(22)(23)种异常为读异常,即(14)(15)(22)(23) $\in E_{RAT}$; 将其称为读异常下的二事务双变量异常,记为 RAT-DDA。 剩余第(16)(17)(26)种异常为交叉异常; 将其称为交叉异常下的二事务双变量异常,记为 IAT-DDA。

表 5 数据异常定义示例

数据异常名	数据异常形式化定义	PCG
Lost Update Committed	$R_1[x_m] \dots W_2[x_{m+1}] \dots C_2 \dots W_1[x_{m+2}]$	图 10-(k) $V = \{t_1, t_2\}$ $E = \{R_1 W_2 C_2[x], W_2 C_2 W_1[x]\}$
Read Skew Committed	$R_1[x_m] \dots W_2[x_{m+1}] \dots W_2[y_n] \dots C_2 \dots R_1[y_n]$	图 11-(m) $V = \{t_1, t_2\}$ $E = \{R_1 W_2 C_2[x], W_2 C_2 R_1[y]\}$

【例 7】两个已知的数据异常 Lost Update 和 Read Skew 的定义如表 5。

下面我们讨论多事务多变量的偏序环,以下简称多事务异常。

多事务异常可以表示 $C_{da} = (N_D, N_T)$, 其中数据异常中变量的个数表示为 $N_D = |D(s)|, N_D \geq 2$, 事务的个数表示为

$N_T = |T(s)|, N_T \geq 3$. 同样的, 对多事务多变量偏序环, 如果其偏序环的形式化定义种满足:

- 1) 偏序环形式化表达中包含 $W_{i/j}[x_m] \dots W_{j/i}[x_{m+1}]$, 称为写异常下的多事务异常 (Multi-variables by Double-transactions Anomalies, MDA, 简称多元异常), 记为 WAT-MDA;
- 2) 偏序环形式化表达中不包含 $W_{i/j}[x_m] \dots W_{j/i}[x_{m+1}]$ 的前提下, 包含 $W_{i/j}[x_m] \dots R_{j/i}[x_m]$, 称为读异常下的多事务异常, 记为 RAT-MDA;
- 3) 偏序环形式化表达中均不包含上面两种情况的情况, 称为交叉异常下多事务异常, 记为 IAT-MDA.

表 6 数据异常的具体定义和分类(粗体, 新报告的异常)

主类	子类/C _{da}	数据异常名称	数据异常形式化定义	编号	PCG 样例
WAT	SDA	Dirty Write	$W_i[x_m] \dots W_j[x_{m+1}] \dots A_i / C_i$	(1)	图 10-(a)
	SDA	Lost Self Update Committed	$W_i[x_m] \dots W_j[x_{m+1}] \dots C_j \dots R_i[x_{m+1}]$	(3)	图 10-(b)
	SDA	Full-Write Committed	$W_i[x_m] \dots W_j[x_{m+1}] \dots C_j \dots W_i[x_{m+2}]$	(4)	图 10-(c)
	SDA	Full-Write	$W_i[x_m] \dots W_j[x_{m+1}] \dots W_i[x_{m+2}]$	(7)	图 10-(d)
	SDA	Lost Update	$R_i[x_m] \dots W_j[x_{m+1}] \dots W_i[x_{m+2}]$	(8)	图 10-(e)
	SDA	Lost Self Update	$W_i[x_m] \dots W_j[x_{m+1}] \dots R_i[x_{m+1}]$	(9)	图 10-(f)
	DDA	Double-Write Skew 2 Committed	$W_i[x_m] \dots W_j[x_{m+1}] \dots W_j[y_n] \dots C_j \dots R_i[y_n]$	(12)	图 11-(a)
	DDA	Full-Write Skew Committed	$W_i[x_m] \dots W_j[x_{m+1}] \dots W_j[y_n] \dots C_j \dots W_i[y_{n+1}]$	(13)	图 11-(b)
	DDA	Full-Write Skew	$W_i[x_m] \dots W_j[x_{m+1}] \dots W_j[y_n] \dots W_i[y_{n+1}]$	(18)	图 11-(c)
	DDA	Double-Write Skew 1	$W_i[x_m] \dots R_j[x_m] \dots W_j[y_n] \dots W_i[y_{n+1}]$	(19)	图 11-(d)
	DDA	Double-Write Skew 2	$W_i[x_m] \dots W_j[x_{m+1}] \dots W_j[y_n] \dots R_i[y_n]$	(21)	图 11-(f)
	DDA	Read-Write Skew 1	$R_i[x_m] \dots W_j[x_{m+1}] \dots W_j[y_n] \dots W_i[y_{n+1}]$	(20)	图 11-(e)
	DDA	Read-Write Skew 2	$W_i[x_m] \dots W_j[x_{m+1}] \dots R_j[y_n] \dots W_i[y_{n+1}]$	(24)	图 11-(g)
	MDA	Step WAT	$\dots W_{i/j}[x_m] \dots W_{j/i}[x_{m+1}] \dots$, and $N_D \geq 3$		图 12-(a)
RAT	SDA	Dirty Read	$W_i[x_m] \dots R_j[x_m] \dots A_i$	(2)	图 10-(g)
	SDA	Non-repeatable Read	$R_i[x_m] \dots W_j[x_{m+1}] \dots R_i[x_{m+1}]$	(10)	图 10-(h)
	SDA	Intermediate Read	$W_i[x_m] \dots R_j[x_m] \dots W_i[x_{m+1}]$	(11)	图 10-(i)
	DDA	Write-Read Skew Committed	$W_i[x_m] \dots R_j[x_m] \dots W_j[y_n] \dots C_j \dots R_i[y_n]$	(14)	图 11-(h)
	DDA	Double-Write Skew 1 Committed	$W_i[x_m] \dots R_j[x_m] \dots W_j[y_n] \dots C_j \dots W_i[y_{n+1}]$	(15)	图 11-(i)
	DDA	Write-Read Skew	$W_i[x_m] \dots R_j[x_m] \dots W_j[y_n] \dots R_i[y_n]$	(22)	图 11-(j)
	DDA	Read Skew	$R_i[x_m] \dots W_j[x_{m+1}] \dots W_j[y_n] \dots R_i[y_n]$	(23)	图 11-(k)
	DDA	Read Skew 2	$W_i[x_m] \dots R_j[x_m] \dots R_j[y_n] \dots W_i[y_{n+1}]$	(25)	图 11-(l)
	MDA	Step RAT	$\dots W_{i/j}[x_m] \dots R_{j/i}[x_m] \dots$, and $N_D \geq 3$ and not include ($\dots W_{i/j}[x_m] \dots W_{j/i}[x_{m+1}] \dots$)		图 12-(b)
	IAT	SDA	Non-repeatable Read Committed	$R_i[x_m] \dots W_j[x_{m+1}] \dots C_j \dots R_i[x_{m+1}]$	(5)
SDA		Lost Update Committed	$R_i[x_m] \dots W_j[x_{m+1}] \dots C_j \dots W_i[x_{m+2}]$	(6)	图 10-(k)
DDA		Read Skew Committed	$R_i[x_m] \dots W_j[x_{m+1}] \dots W_j[y_n] \dots C_j \dots R_i[y_n]$	(16)	图 11-(m)
DDA		Read-Write Skew 1 Committed	$R_i[x_m] \dots W_j[x_{m+1}] \dots W_j[y_n] \dots C_j \dots W_i[y_{n+1}]$	(17)	图 11-(n)
DDA		Write Skew	$R_i[x_m] \dots W_j[x_{m+1}] \dots R_j[y_n] \dots W_i[y_{n+1}]$	(26)	图 11-(o)
MDA		Step IAT	Not include ($\dots W_{i/j}[x_m] \dots R_{j/i}[x_m] \dots$ and $\dots W_{i/j}[x_m] \dots W_{j/i}[x_{m+1}] \dots$) $N_D \geq 3$		图 12-(c)

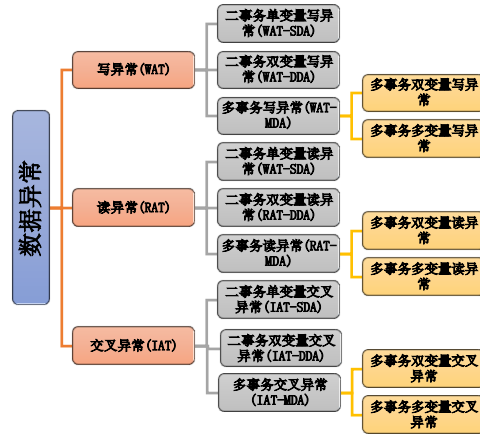


图9 数据异常类别结构图

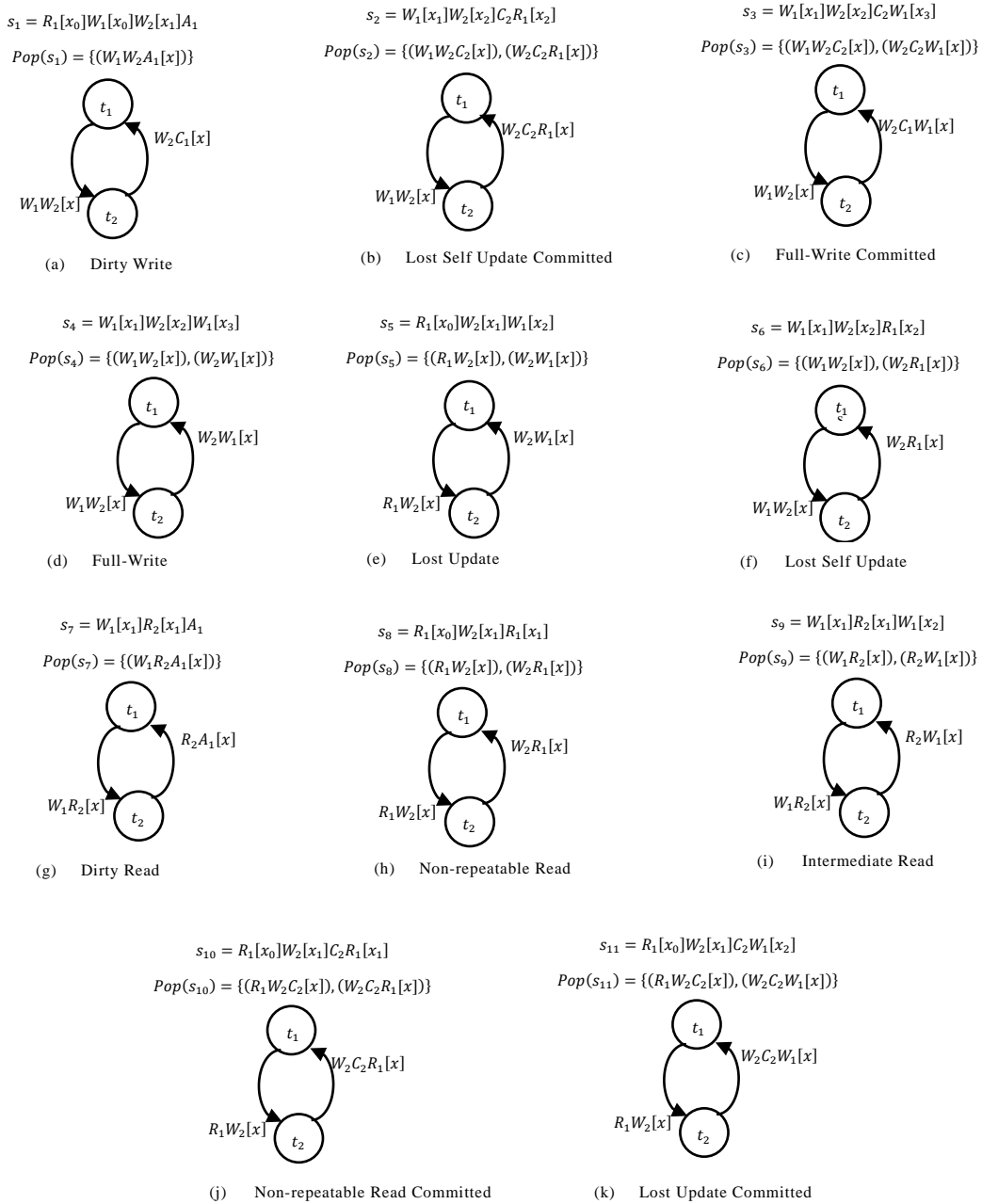


图10 单元数据异常 PCG 图

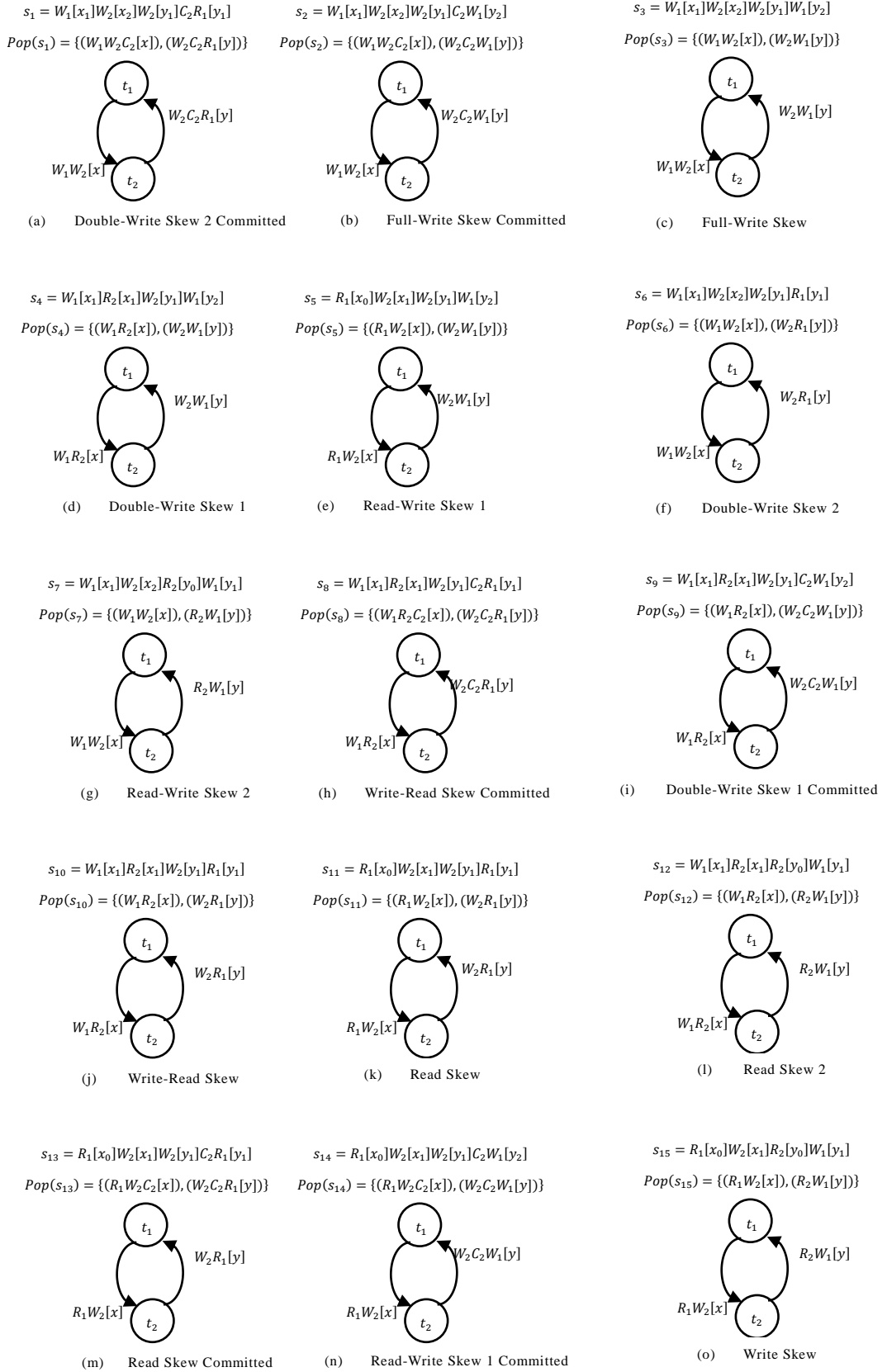


图 11 双元数据异常 PCG 图

数据异常的类别可表示为图 9。

根据前述的数据异常定义和分类方法,每一种数据异常的定义如表 6 所示.该表根据数据异常分类,并给出每种具体的数据异常的名字和其形式化定义。

【例 8】图 9 中各 MDA 类异常可能对应的调度如下所示:

1. 图 12-a, Step WAT, 其调度表达式为: $R_1[x_0]W_2[x_1]W_2[y_1]W_3[y_2]R_3[z_0]W_1[z_1]$. 该调度中包含了 WW 偏序对.
2. 图 12-b, Step RAT, 其调度表达式为: $R_1[x_0]W_2[x_1]W_2[y_1]R_3[y_1]R_3[z_0]W_1[z_1]$. 该调度中没有包含 WW 偏序对但包含了 WR 偏序对.
3. 图 12-c, Step IAT, 其调度表达式为: $R_1[x_0]W_2[x_1]R_2[y_0]W_3[y_1]R_3[z_0]W_1[z_1]$. 该调度中没有包含 WW 和 WR 偏序对但包含了 RW 偏序对. 在 Step IAT 类别中的边的种类为 $\{RW, WCR, WCW\}$, 并且必然包含边 RW .

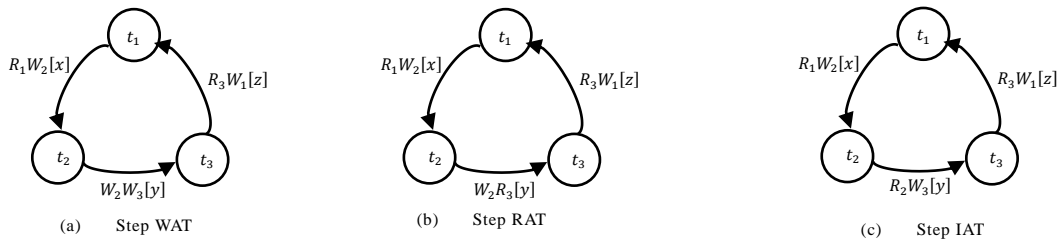


图 12 多元数据异常 PCG 图

4 隔离级别

文献[1,2,3]等分别基于有限的数据异常定义了不同个数的隔离级别,其定义隔离级别的体系和方法复杂难懂.难懂的原因在于,一是缺乏对数据异常的体系化定义(文献[1,2]基于有限个数的数据异常定义了不同级别个数的数据异常,而表 1 和本文 3.2 节表明数据异常有无数个,因此基于有限个数的数据异常定义隔离级别缺乏科学性),二是隔离级别的分类标准不统一(文献[1,2]基于有限个数的数据异常定义隔离级别,文献[3]基于冲突环上边的特征和特定的两个数据异常定义隔离级别),三是隔离级别较多不简单明了(文献[1]定义了四种隔离级别,文献[2]定义了八种隔离级别).

表 7 工程实践用的简化的隔离级别

主类	子类/C _{da}	数据异常名称	NRW	NA
WAT	SDA	Dirty Write	Not Possible	Not Possible
	SDA	Lost Update	Not Possible	Not Possible
	SDA	Lost Self Update	Not Possible	Not Possible
	SDA	Full-Write	Not Possible	Not Possible
	SDA	Full-Write Committed	Not Possible	Not Possible
	DDA	Read-Write Skew 1	Not Possible	Not Possible
	DDA	Read-Write Skew 2	Not Possible	Not Possible
	DDA	Double-Write Skew 1	Not Possible	Not Possible
	DDA	Double-Write Skew 2	Not Possible	Not Possible
	DDA	Double-Write Skew 2 Committed	Not Possible	Not Possible
	DDA	Full-Write Skew	Not Possible	Not Possible
	DDA	Full-Write Skew Committed	Not Possible	Not Possible
	MDA	Step WAT	Not Possible	Not Possible
	RAT	SDA	Dirty Read	Not Possible
SDA		Non-repeatable Read	Not Possible	Not Possible
SDA		Non-repeatable Read Committed	Not Possible	Not Possible
SDA		Intermediate Read	Not Possible	Not Possible
DDA		Read Skew	Not Possible	Not Possible
DDA		Read Skew 2	Not Possible	Not Possible
DDA		Write-Read Skew	Not Possible	Not Possible
DDA		Write-Read Skew Committed	Not Possible	Not Possible
DDA		Double-Write Skew 1 Committed	Not Possible	Not Possible
MDA		Step RAT	Not Possible	Not Possible
IAT	SDA	Lost Update Committed	Possible	Not Possible
	DDA	Read Skew Committed	Possible	Not Possible
	DDA	Read-Write Skew 1 Committed	Possible	Not Possible
	DDA	Write Skew	Possible	Not Possible
	MDA	Step IAT	Possible	Not Possible

文献[1,2,3]用不同的隔离级别消除不同的数据异常,其本质在于对数据异常进行类别划分.因此,隔离级别是对数据异常分类方法的更高层次的分类方式,而传统的隔离级别定义对数据异常的划分方式在不同的隔离级别上由弱到强形成递进关系,但这种方式终究还是一种分类方法.因此,出于不同的目的,我们可以灵活定义不同的隔离级别体系.而灵活定义隔离级别,是为了便于工程实现和细化认知并发访问控制体系.

基于表 6 的数据异常定义和分类,本文给出两种隔离级别的定义方式,以演示如何基于全部数据异常完备且灵活地定义隔离级别。

简化的隔离级别的定义,只包括两种隔离级别(表 7),目的是设定简单规则低代价地避免一部分数据异常,并降低并发访问控制算法的复杂度.该方式可用于实践中构建工程系统:

1. No Read-Write Anomalies(NRW):即不存在 WAT 和 RAT 类数据异常,但不能消除 IAT 类数据异常。
2. No Anomalies(NA):不允许任何数据异常发生.效果等价于 ANSI-SQL 标准[1]定义可串行化隔离级别。
3. NRW 级别较弱,NA 级别最强。

简化的隔离级别的定义,对指导并发访问控制算法有帮助.例如,按照表 7 简化的的隔离级别,可以通过规则(如采用禁止 WW 边出现避免 WAT 异常;采用读已提交,快照读等方式避免 RAT 类型的数据异常),达到 NRW 级别;通过类似可串行化快照(Serializable Snapshot Isolation,SSI)[29]技术,仅需识别连续的 2 个 RW 边而达到 NA 级别;仅定义两种隔离级别,使得隔离级别简化,且因规则的使用而导致并发算法的实现变得简单,并发算法通过规则避免多数种类的数据异常是一种低代价的并发算法实现方式,因此并发算法采用更少的步骤可使的事务处理的吞吐率变高。

另外,前述给出的规则,也适用于其他的并发访问控制算法如 2PL[14],T/O[23],OCC[30],MVCC[19,29,31]等。

非简化的隔离级别的定义,包括四种隔离级别(表 8),分级依据是数据异常可以通过多种简单规则加以避免,但不同的规则适用于不同的级别使得包含多种隔离级别.该方式可用于教学演示系统,以展示数据异常和隔离级别之间的细微关系:

1. No Write Anomalies(NW):即不存在 WAT 类数据异常,但不能消除 RAT 和 IAT 类数据异常。
2. No Read-Write Anomalies(NRW):即不存在 WAT 和 RAT 类数据异常,但不能消除 IAT 类数据异常,也不能消除幻读,读偏序类数据异常。
3. No Predicate Anomalies(NPA):即不存在 WAT 和 RAT 类数据异常,也不存在幻读,读偏序类数据异常,但不能消除 IAT 类数据异常.但是,本文为了表达清晰易懂,简化了模型,没有讨论谓词类数据异常,因此幻读异常未能体现在表 6 的定义中,因此也未能在表 7 和表 8 中。
4. No Anomalies(NA):不允许任何数据异常发生.效果等价于 ANSI-SQL 标准[1]定义可串行化隔离级别。
5. NW 级别较弱,NRW 次之,NPA 级别较强,NA 级别最强。
6. 对于 NW,禁止 WW 冲突即可消除该类数据异常.对于 NRW,采用读已提交规则即可消除该类数据异常.对于 NPA,采用快照即可消除该类数据异常.但是,这些规则不是唯一消除相应数据异常的方式.每一种规则用以消除不同的隔离级别,前三种规则分开使用就是教学演示用的隔离级别;如三种规则同时使用就对应简化的隔离级别,因此可知:隔离级别可以灵活定义。

表 8 教学演示用的细粒度隔离级别

主类	子类/C _{da}	数据异常名称	NW	NRW	NPA	NA
WAT	SDA	Dirty Write	Not Possible	Not Possible	Not Possible	Not Possible
	SDA	Lost Update	Not Possible	Not Possible	Not Possible	Not Possible
	SDA	Lost Self Update	Not Possible	Not Possible	Not Possible	Not Possible
	SDA	Full-Write	Not Possible	Not Possible	Not Possible	Not Possible
	SDA	Full-Write Committed	Not Possible	Not Possible	Not Possible	Not Possible
	DDA	Read-Write Skew 1	Not Possible	Not Possible	Not Possible	Not Possible
	DDA	Read-Write Skew 2	Not Possible	Not Possible	Not Possible	Not Possible
	DDA	Double-Write Skew 1	Not Possible	Not Possible	Not Possible	Not Possible
	DDA	Double-Write Skew 2	Not Possible	Not Possible	Not Possible	Not Possible
	DDA	Double-Write Skew 2 Committed	Not Possible	Not Possible	Not Possible	Not Possible
	DDA	Full-Write Skew	Not Possible	Not Possible	Not Possible	Not Possible
	DDA	Full-Write Skew Committed	Not Possible	Not Possible	Not Possible	Not Possible
	MDA	Step WAT	Not Possible	Not Possible	Not Possible	Not Possible
	RAT	SDA	Dirty Read	Possible	Not Possible	Not Possible
SDA		Non-repeatable Read	Possible	Not Possible	Not Possible	Not Possible
SDA		Non-repeatable Read Committed	Possible	Possible	Not Possible	Not Possible
SDA		Intermediate Read	Possible	Not Possible	Not Possible	Not Possible
DDA		Read Skew	Possible	Possible	Not Possible	Not Possible
DDA		Read Skew 2	Possible	Not Possible	Not Possible	Not Possible
DDA		Write-Read Skew	Possible	Not Possible	Not Possible	Not Possible
DDA		Write-Read Skew Committed	Possible	Not Possible	Not Possible	Not Possible
DDA		Double-Write Skew 1 Committed	Possible	Not Possible	Not Possible	Not Possible
MDA		Step RAT	Possible	Not Possible	Not Possible	Not Possible
IAT	SDA	Lost Update Committed	Possible	Possible	Possible	Not Possible
	DDA	Read Skew Committed	Possible	Possible	Possible	Not Possible
	DDA	Read-Write Skew 1 Committed	Possible	Possible	Possible	Not Possible
	DDA	Write Skew	Possible	Possible	Possible	Not Possible
	MDA	Step IAT	Possible	Possible	Possible	Not Possible

5 结束语

本文体系化地研究了数据异常.利用偏序环图定义了数据异常,对数据异常做了分类,使得可以区分所有的数据异常.之后基于数据异常的分类定义了简明的隔离级别.

未来,我们将就谓词类数据异常,并发访问控制算法等展开研究,以期进一步完善数据异常的体系化研究工作.

本文内容在研究过程中,得到中国人民大学教育部数据工程和知识工程重点实验室和腾讯公司的参与和支持,特别向项目参与人,支持者致谢.

基于本文的内容实现了一些基本的数据异常识别算法,相关代码已经在 [github](https://github.com/tencent/3TS) 开源.

<https://github.com/tencent/3TS>

References:

- [1] ANSI X3.135-1992, American National Standard for Information Systems – Database Language – SQL, Nov 1992.
- [2] Hal Berenson, Philip A. Bernstein, Jim Gray, Jim Melton, Elizabeth J. O’Neil, Patrick E. O’Neil: A Critique of ANSI SQL Isolation Levels. SIGMOD Conference 1995: 1-10
- [3] A. Adya, B. Liskov, and P. O’Neil. Generalized isolation level definitions. In Proceedings of the 16th International Conference on Data Engineering, ICDE ’00, pages 67–78, Washington, DC, USA, 2000. IEEE Computer Society.
- [4] C. Xie, C. Su, C. Littlely, L. Alvisi, M. Kapritsos, and Y. Wang. High-performance acid via modular concurrency control. In Proceedings of the 25th Symposium on Operating Systems Principles, SOSP’15, pages 279{294, New York, NY, USA, 2015. ACM
- [5] Ralf Schenkel, Gerhard Weikum, Norbert Weissenberg, et al. “Federated transaction Management with Snapshot Isolation”. In: transactions and Database Dynamics. Vol. 1773. Springer Berlin / Heidelberg, 2000, pp. 1–25.
- [6] P. Bailis, A. Fekete, J. M. Hellerstein, A. Ghodsi, and I. Stoica, “Scalable atomic visibility with ramp transactions,” in SIGMOD, 2014, pp. 27–38.
- [7] Andrea Cerone, Alexey Gotsman, and Hongseok Yang. Algebraic laws for weak consistency. International Conference on Concurrency theory (CONCUR 2017), LIPICs 85, pages 26:1-26:18, 2017.
- [8] Carsten Binnig, Stefan Hildenbrand, Franz Färber, Donald Kossmann, Juchang Lee, Norman May: Distributed snapshot isolation: global transactions pay globally, local transactions pay locally. VLDB J. 23(6): 987-1011 (2014)
- [9] Lamport L, Shostak R, Pease M. the Byzantine generals problem[J]. ACM transactions on Programming Languages and Systems (TOPLAS), 1982, 4(3): 382-401
- [10] Alan Fekete, Elizabeth O’Neil, and Patrick O’Neil. A read-only transaction anomaly under snapshot isolation. SIGMOD Rec., 33(3):12–14, 2004.
- [11] A. Fekete, D. Liarakis, E. O’Neil, P. O’Neil, and D. Shasha, “Making snapshot isolation serializable,” ACM TODS, vol. 30, no. 2, pp. 492–528, 2005.
- [12] Du XY et al: Big Data Management, Higher Education Press, 2017.
- [13] https://wiki.postgresql.org/wiki/SSI#Read_Only_transactions
- [14] Kapali P. Eswaran, Jim Gray, Raymond A. Lorie, Irving L. traiger: the Notions of Consistency and Predicate Locks in a Database System. Commun. ACM 19(11): 624-633 (1976)
- [15] Andrea Cerone and Alexey Gotsman. Analysing snapshot isolation. Journal of the ACM (JACM 2018), 65(2), 11:1-11:41, 2018.
- [16] Christos H. Papadimitriou: the serializability of concurrent database updates. J. ACM 26(4): 631-653 (1979)
- [17] Jim Gray, Raymond A. Lorie, Gianfranco R. Putzolu, Irving L. traiger: Granularity of Locks and Degrees of Consistency in a Shared Data Base. IFIP Working Conference on Modelling in Data Base Management Systems 1976: 365-394
- [18] Wang S, Sa SX: Introduction to Database System, 5th edition, Higher Education Press, 2014.
- [19] P. Bernstein, V. Hadzilacos, and N. Goodman. Concurrency Control and Recovery in Database Systems. Addison-Wesley, 1987.
- [20] Gerhard Weikum, Gottfried Vossen (2001): transactional Information Systems, Chapter 19, Elsevier, ISBN 1-55860-508-8
- [21] Ramez Elmasri, Shamkant B. Navathe: the Fundamentals of Database Systems, 5th Edition
- [22] P. A. Bernstein, D. Shipman, and W. Wong. Formal aspects of serializability in database concurrency control. IEEE transactions on Software Engineering, 5(3):203–216, 1979.
- [23] Roshan K. thomas, Ravi S. Sandhu: towards a Unified Framework and theory for Reasoning about Security and Correctness of transactions in Multilevel databases. DBSec 1993: 309-328.
- [24] Kamal Zellag and Bettina Kemme. Real-time quantification and classification of consistency anomalies in multi-tier

architectures. In Proceedings of the 27th IEEE International Conference on Data Engineering, ICDE '11, pages 613–624. IEEE Computer Society, 2011

[25] Kamal Zellig and Bettina Kemme. How consistent is your cloud application? In Proceedings of the 3rd ACM Symposium on Cloud Computing, SoCC '12, pages 6:1–6:14. ACM, 2012.

[26] Kamal Zellig and Bettina Kemme. Consistency anomalies in multi-tier architectures: Automatic detection and prevention. the VLDB Journal, 3(1):147–172, 2014

[27] Kamal Zellig, Bettina Kemme: ConsAD: a real-time consistency anomalies detector. SIGMOD Conference 2012: 641–644

[28] Alan Fekete, Shirley N. Goldrei, and Jorge Pérez Asenjo. Quantifying isolation anomalies. Proc. VLDB Endow., 2(1):467–478, August 2009.

[29] Michael J. Cahill, Uwe Röhm, Alan David Fekete: Serializable isolation for snapshot databases. ACM trans. Database Syst. 34(4): 20:1–20:42 (2009)

[30] H. t. Kung, John t. Robinson: On Optimistic Methods for Concurrency Control. VLDB 1979: 351

[31] Christos H. Papadimitriou, Paris C. Kanellakis: On Concurrency Control by Multiple Versions. ACM trans. Database Syst. 9(1): 89–99 (1984)

附中文参考文献:

[12] 杜小勇等. 大数据管理. 高等教育出版社, 2017

[18] 王珊, 萨师焯. 数据库系统概论(第五版). 高等教育出版社, 2014

附录 1: $W_i \dots W_j \dots (A_i \text{ or } C_i) \text{ and } (A_j \text{ or } C_j)$ 异常的详细推导

首先对事务状态操作的定义如下:

- $W_i[x_n]$ 表示事务 t_i 的写操作, 其存储变量的原始值 x_{n-1} 和写操作后的更改值 x_n ;
- A_i 表示变量的当前值更新为事务 t_i 最初写操作存储的原始值;
- C_i 表示变量提交当前值.

因此 $W_i \dots W_j \dots (A_i \text{ or } C_i) \text{ and } (A_j \text{ or } C_j)$ 的组合情况有 8 种:

1. $W_i[x_1] \dots W_j[x_2] \dots C_i \dots C_j$:

操作 $W_i[x_1] \dots W_j[x_2]$ 表示变量 x 的当前值为 x_2 ;

操作 C_i 表示提交当前值 $x = x_2$; 此时与 t_i 希望提交的 $x = x_1$ 存在异常;

2. $W_i[x_1] \dots W_j[x_2] \dots C_i \dots A_j$: 与情形 1 相同, 在 C_i 发生时即存在了异常;

3. $W_i[x_1] \dots W_j[x_2] \dots A_i \dots C_j$:

操作 A_i 表示将当前值回滚为事务 t_i 最初写操作存储的原始值 $x = x_0$;

操作 C_j 表示提交当前值 $x = x_0$; 此时与 t_j 希望提交的 $x = x_2$ 存在异常;

$$W_i[x_1] \dots W_j[x_2] \dots A_i \dots A_j$$

操作 A_i 表示将当前值回滚为事务 t_i 最初写操作存储的原始值 $x = x_0$;

操作 A_j 表示将当前值回滚为事务 t_j 最初写操作存储的原始值 $x = x_1$; 但是 x_1 为数据库中从未存储过的数据, 因此存在数据异常;

$$W_i[x_1] \dots W_j[x_2] \dots C_j \dots C_i$$

操作 C_j 表示提交当前值 $x = x_2$;

操作 C_i 表示提交当前值 $x = x_2$, 此时与 t_i 希望提交的 $x = x_2$ 存在异常

$$W_i[x_1] \dots W_j[x_2] \dots C_j \dots A_i$$

操作 C_j 表示提交当前值 $x = x_2$;

操作 A_i 将当前值回滚为事务 t_i 最初写操作存储的原始值 $x = x_0$, 不产生数据异常;

$$W_i[x_1] \dots W_j[x_2] \dots A_j \dots C_i$$

操作 A_j 将当前值回滚为事务 t_j 最初写操作存储的原始值 $x = x_1$;

操作 C_i 表示提交当前值 $x = x_1$, 不产生异常;

$$W_i[x_1] \dots W_j[x_2] \dots A_j \dots A_i$$

操作 A_j 将当前值回滚为事务 t_j 最初写操作存储的原始值 $x = x_1$;

操作 A_i 将当前值回滚为事务 t_i 最初写操作存储的原始值 $x = x_0$, 不产生异常.

因此, $W_i[x_1] \dots W_j[x_2] \dots C_i$ (情形 1-2) 和 $W_i[x_1] \dots W_j[x_2] \dots A_i$ (情形 3-4) 一定存在异常; $W_i[x_1] \dots W_j[x_2] \dots C_j$ 是否存在异常不确定, 需要后续操作, 因此作为偏序对; $W_i[x_1] \dots W_j[x_2] \dots A_j$ 一定非异常.

附录 2: 单变量双边偏序环形式化推导

所有偏序对组成集合为 $Pop = \{W_iC_jR_j, W_iC_iW_j, R_iC_iW_j, W_iW_j, W_iR_j, R_iW_j, W_iR_iA_i, W_iW_jC_i, W_iW_jA_i\}$ 共 9 种, 其中 $W_iW_j = W_iW_j, W_iW_jC_j,$

$W_iR_j = W_iR_j, W_iR_jC_j, W_iR_jC_i, R_iW_j = R_iW_j, R_iW_jC_j, R_iW_jC_i, R_iW_jA_i$. 其双边组合推导如下.

当调度 s 中所有偏序对组成的集合 $Pop(s)$ 中包含 $\{W_iW_jA_i, W_iW_jC_i, R_iW_jA_i\}$ 三种偏序对时, 本身即为偏序环:

➤ WA 表示存在偏序环 $W_i[x_m]W_j[x_{m+1}] - W_j[x_{m+1}]A_i$

形式化定义为: $W_i[x_m] \dots W_j[x_{m+1}] \dots A_i$, 即 Dirty Write. (1)

➤ WC 表示存在偏序环 $W_i[x_m]W_j[x_{m+1}] - W_j[x_{m+1}]C_i$

形式化定义为: $W_i[x_m] \dots W_j[x_{m+1}] \dots C_i$, 即 Dirty Write, 同(1).

➤ RA 表示存在偏序环 $W_i[x_m]R_j[x_m] - R_j[x_m]A_i$

形式化定义为: $W_i[x_m] \dots R_j[x_m] \dots A_i$, 即 Dirty Read. (2)

因此只考虑 $\{W_iC_jR_j, W_iC_iW_j, R_iC_iW_j, W_iW_j, W_iR_j, R_iW_j\}$ 的组合情形.

对二事务单变量偏序环 $PCG(s) = (\{t_i, t_j\}, \{p_{ij}[x], p_{ji}[x]\})$, 由 $t_i <_s t_j$, 组成的偏序环有如下可能的组合方式:

1. p_{ij} 不可能为偏序对 $W_iC_jR_j, W_iC_iW_j$ 或 $R_iC_iW_j$. 否则由于 $t_i <_s t_j$, 则偏序对 p_{ij} 中不会出现 t_i 的操作;

2. 若偏序对 $p_{ji}[x]$ 中出现偏序对 $W_jC_jR_i, W_jC_jW_i$ 或 $R_jC_jW_i$, 则 p_{ij} 必然存在 C_j , 因此偏序对的组合方式为以下几种情况:

➤ $W_iW_jC_j - W_jC_jR_i$ 或 $W_iR_jC_j - W_jC_jR_i$

形式化定义为: $W_i[x_m] \dots W_j[x_{m+1}] \dots C_j \dots R_i[x_{m+1}]$, 即 Lost Self Update Committed. (3)

➤ $W_iW_jC_j - W_jC_jW_i$ 或 $W_iR_jC_j - W_jC_jW_i$ 或 $W_iW_jC_j - R_jC_jW_i$

形式化定义为: $W_i[x_m] \dots W_j[x_{m+1}] \dots C_j \dots W_i[x_{m+2}]$, 即 Full-Write Committed. (4)

➤ $R_i W_j C_j - W_j C_j R_i$

形式化定义为: $R_i[x_m] \dots W_j[x_{m+1}] \dots C_j \dots R_i[x_{m+1}]$, 即 Non-repeatable Read Committed. (5)

➤ $R_i W_j C_j - W_j C_j W_i$ 或 $R_i W_j C_j - R_j C_j W_i$

形式化定义为: $R_i[x_m] \dots W_j[x_{m+1}] \dots C_j \dots W_i[x_{m+2}]$, 即 Lost Update Committed. (6)

➤ $W_i R_j C_j - R_j C_j W_i$, 此时不存在异常.

3. 根据偏序对的定义, 偏序对 $W_i W_j = \{W_i W_j, W_i W_j C_j\}$, 则偏序环的组合方式为以下几种

1) 若 $p_{ji}[x] = W_j W_i$, 可构成偏序环

➤ $W_i W_j - W_j W_i$ 或 $W_i R_j - W_j W_i$

形式化定义为: $W_i[x_m] \dots W_j[x_{m+1}] \dots W_i[x_{m+2}]$, 即 Full Write. (7)

➤ $R_i W_j - W_j W_i$

形式化定义为: $R_i[x_m] \dots W_j[x_{m+1}] \dots W_i[x_{m+2}]$, 即 Lost Update. (8)

2) 若 $p_{ji}[x] = W_j W_i C_i$, 则偏序对 $p_{ij}[x]$ 中必然含操作 C_i , 因此可构成偏序环

➤ $W_i W_j C_i - W_j W_i C_i$, 此时 $p_{ij}[x] = W_i W_j C_i$ 已经构成异常的偏序环, 偏序环种类同(1).

➤ $W_i R_j C_i - W_j W_i C_i$

形式化定义为: $W_i[x_m] \dots W_j[x_{m+1}] \dots W_i[x_{m+2}]$, 同(7).

➤ $R_i W_j C_i - W_j W_i C_i$, 此时 $p_{ij}[x] = R_i W_j C_i$, 同(8).

4. 根据偏序对的定义, 偏序对 $WR = \{W_i R_j, W_i R_j C_j, W_i R_j C_i\}$, 则偏序环的组合方式为以下几种

1) 若 $p_{ji}[x] = W_j R_i$, 可构成偏序环

➤ $W_i W_j - W_j R_i$ 或 $W_i R_j - W_j R_i$

形式化定义为: $W_i[x_m] \dots W_j[x_{m+1}] \dots R_i[x_{m+1}]$, 即 Lost Self Update. (9)

➤ $R_i W_j - W_j R_i$

形式化定义为: $R_i[x_m] \dots W_j[x_{m+1}] \dots R_i[x_{m+1}]$, 即 Non-repeatable Read. (10)

2) 若 $p_{ji}[x] = W_j R_i C_i$, 则偏序对 $p_{ij}[x]$ 中必然含操作 C_i , 因此可构成偏序环

➤ $W_i W_j C_i - W_j R_i C_i$, 此时 $p_{ij}[x] = W_i W_j C_i$ 已经构成偏序环, 偏序环种类同(1).

$$\triangleright W_i R_j C_i - W_j R_i C_i$$

形式化定义为: $W_i[x_m] \dots W_j[x_{m+1}] \dots R_i[x_{m+1}]$, 同(9).

$$\triangleright R_i W_j C_i - W_j R_i C_i, \text{此时 } p_{ij}[x] = R_i W_j C_i, \text{同(10).}$$

3) 若 $p_{ji}[x] = W_j R_i C_j$, 则偏序对 $p_{ij}[x]$ 中必然含操作 C_j , 因此可构成偏序环

$$\triangleright W_i W_j C_j - W_j R_i C_j$$

形式化定义为: $W_i[x_m] \dots W_j[x_{m+1}] \dots R_i[x_{m+1}]$, 同(9).

$$\triangleright W_i R_j C_j - W_j R_i C_j$$

形式化定义为: $W_i[x_m] \dots W_j[x_{m+1}] \dots R_i[x_{m+1}]$, 同(9).

$$\triangleright R_i W_j C_j - W_j R_i C_j,$$

形式化定义为: $R_i[x_m] \dots W_j[x_{m+1}] \dots R_i[x_{m+1}]$, 同(10).

5. 根据偏序对的定义, 偏序对 $RW = \{R_i W_j, R_i W_j C_j, R_i W_j A_i\}$, 则偏序环的组合方式为以下几种

1) 若 $p_{ji}[x] = R_j W_i$, 可构成偏序环

$$\triangleright W_i W_j - R_j W_i$$

形式化定义为: $W_i[x_m] \dots W_j[x_{m+1}] \dots W_i[x_{m+1}]$, 同(7).

$$\triangleright W_i R_j - R_j W_i$$

形式化定义为: $W_i[x_m] \dots R_j[x_m] \dots W_i[x_{m+1}]$, 即 Intermediate Read. (11)

$$\triangleright R_i W_j - R_j W_i$$

形式化定义为: $R_i[x_m] \dots W_j[x_{m+1}] \dots W_i[x_{m+2}]$, 同(8).

2) 若 $p_{ji}[x] = R_j W_i C_i$, 则偏序对 $p_{ij}[x]$ 中必然含操作 C_i , 因此可构成偏序环

$$\triangleright W_i W_j C_i - R_j W_i C_i, \text{此时 } p_{ij}[x] = W_i W_j C_i \text{ 已经构成偏序环, 偏序环种类同(1).}$$

$$\triangleright W_i R_j C_i - R_j W_i C_i$$

形式化定义为: $W_i[x_m] \dots R_j[x_m] \dots W_i[x_{m+1}]$, 同(11).

$$\triangleright R_i W_j C_i - R_j W_i C_i, \text{此时 } p_{ij}[x] = R_i W_j C_i, \text{同(8).}$$

3) 若 $p_{ji}[x] = R_j W_i C_j$, 则偏序对 $p_{ij}[x]$ 中必然含操作 C_j , 因此可构成偏序环

$$\triangleright W_i W_j C_j - R_j W_i C_j, \text{此时 } p_{ij}[x] = W_j W_i C_j \text{ 已经构成偏序环, 偏序环种类同(1).}$$

- $W_i R_j C_j - R_j W_i C_j$, 此时不存在异常;
 - $R_i W_j C_j - R_j W_i C_j$, 此时 $p_{ij}[x] = R_i W_j C_i$, 同(8).
- 4) 若 $p_{ji}[x] = R_j W_i A_j$, 则偏序对 $p_{ij}[x]$ 中必然含操作 A_j , 因此可构成偏序环
- $W_i W_j A_j - R_j W_i A_j$, 此时存在操作组合 $W_j \dots W_i \dots A_j$, 偏序环种类同(2);
 - $W_i R_j A_j - R_j W_i A_j$, 此时存在操作组合不产生异常;
 - $R_i W_j A_j - R_j W_i A_j$, 此时存在操作组合 $W_j \dots W_i \dots A_j$, 偏序环种类同(2);

附录 3: 两变量双边偏序环形式化推导

设二事务双变量偏序环为 $PCG(s) = (\{t_i, t_j\}, \{p_{ij}[x], p_{ji}[y]\})$, 由 $t_i <_s t_j$, 组成的偏序环有如下可能的组合方式:

1. 显然, 双变量下调度 s 中所有偏序对组成的集合 $Pop(s)$ 中若包含 $\{W_i W_j A_j, W_i W_j C_i, R_i W_j A_j\}$ 三种偏序对, 偏序对本身构成单变量偏序环, 不会形成双变量环.
2. 同样的 p_{ij} 不可能为偏序对 $W_i C_i R_j$ 或 $W_i C_i W_j$. 否则由于 $t_i <_s t_j$, 则偏序对 p_{ji} 中不会出现 t_i 的操作;
3. 若偏序对 $p_{ji}[y]$ 中出现偏序对 $W_j C_j R_i$ 或 $W_j C_j W_i$, 则 p_{ij} 必然存在 C_j , 因此偏序对的组合方式为以下几种情况:

- $W_i W_j C_j[x] - W_j C_j R_i[y]$

形式化定义为: $W_i[x_m] \dots W_j[x_{m+1}] \dots W_j[y_n] \dots C_j \dots R_i[y_n]$, 即 Double-Write Skew2 Committed (12)

- $W_i W_j C_j[x] - W_j C_j W_i[y]$

形式化定义为: $W_i[x_m] \dots W_j[x_{m+1}] \dots W_j[y_n] \dots C_j \dots W_i[y_{n+1}]$, 即 Full-Write Skew Committed. (13)

- $W_i R_j C_j[x] - W_j C_j R_i[y]$

形式化定义为: $W_i[x_m] \dots R_j[x_m] \dots W_j[y_n] \dots C_j \dots R_i[y_n]$, 即 Write-Read Skew Committed. (14)

- $W_i R_j C_j[x] - W_j C_j W_i[y]$

形式化定义为: $W_i[x_m] \dots R_j[x_m] \dots W_j[y_n] \dots C_j \dots W_i[y_{n+1}]$, 即 Double-Write Skew1 Committed. (15)

- $R_i W_j C_j[x] - W_j C_j R_i[y]$

形式化定义为: $R_i[x_m] \dots W_j[x_{m+1}] \dots W_j[y_n] \dots C_j \dots R_i[y_n]$, 即 Read Skew Committed. (16)

- $R_i W_j C_j[x] - W_j C_j W_i[y]$

形式化定义为: $R_i[x_m] \dots W_j[x_{m+1}] \dots W_j[y_n] \dots C_j \dots W_i[y_{n+1}]$, 即 Read-Write Skew1 Committed. (17)

4. 根据偏序对的定义, 偏序对 $WW = \{W_i W_j, W_i W_j C_j\}$, 则偏序环的组合方式为以下几种

1) 若 $p_{ji}[y]=W_jW_i$,可构成偏序环

$$\triangleright W_jW_j[x]-W_jW_i[y]$$

形式化定义为: $W_i[x_m]...W_j[x_{m+1}]...W_j[y_n]...W_i[y_{n+1}]$,即 Full Write Skew. (18)

$$\triangleright W_iR_j[x]-W_jW_i[y]$$

形式化定义为: $W_i[x_m]...R_j[x_m]...W_j[y_n]...W_i[y_{n+1}]$,即 Double-Write Skew1. (19)

$$\triangleright R_jW_j[x]-W_jW_i[y]$$

形式化定义为: $R_i[x_m]...W_j[x_{m+1}]...W_j[y_n]...W_i[y_{n+1}]$,即 Read-Write Skew1. (20)

2) 若 $p_{ji}[y]=W_jW_iC_i$,则偏序对 $p_{ij}[x]$ 中必然含操作 C_i ,因此可构成偏序环

$$\triangleright W_jW_jC_i[x]-W_jW_iC_i[y],此时 p_{ij}[x]=W_jW_jC_i已经构成偏序环,偏序环种类同(1).$$

$$\triangleright W_jR_jC_i[x]-W_jW_iC_i[y]$$

形式化定义为: $W_i[x_m]...R_j[x_m]...W_j[y_n]...W_i[y_{n+1}]$,同(19).

$$\triangleright R_jW_jC_i[x]-W_jW_iC_i[y],此时 p_{ij}[x]=R_jW_jC_i,同(20).$$

5. 根据偏序对的定义,偏序对 $WR=\{W_iR_j, W_iR_jC_j, W_iR_jC_i\}$,则偏序环的组合方式为以下几种

1) 若 $p_{ji}[y]=W_jR_i$,可构成偏序环

$$\triangleright W_jW_j[x]-W_jR_i[y]$$

形式化定义为: $W_i[x_m]...W_j[x_{m+1}]...W_j[y_n]...R_i[y_{n+1}]$,即 Double-Write Skew2. (21)

$$\triangleright W_iR_j[x]-W_jR_i[y]$$

形式化定义为: $W_i[x_m]...R_i[x_m]...W_j[y_n]...R_i[y_n]$,即 Write-Read Skew. (22)

$$\triangleright R_iW_j[x]-W_jR_i[y]$$

形式化定义为: $R_i[x_m]...W_j[x_{m+1}]...W_j[y_n]...R_i[y_n]$,即 Read Skew. (23)

2) 若 $p_{ji}[y]=W_jR_iC_i$,则偏序对 $p_{ij}[x]$ 中必然含操作 C_i ,因此可构成偏序环

$$\triangleright W_jW_jC_i[x]-W_jR_iC_i[y],此时 p_{ij}[x]=W_jW_jC_i已经构成偏序环,偏序环种类同(1).$$

$$\triangleright W_iR_jC_i[x]-W_jR_iC_i[y]$$

形式化定义为: $W_i[x_m]...R_j[x_m]...W_j[y_n]...R_i[y_n]$,同(22).

$$\triangleright R_iW_jC_i[x]-W_jR_iC_i[x],此时 p_{ij}[x]=R_iW_jC_i,同(23).$$

3) 若 $p_{ji}[y] = W_j R_i C_j$, 则偏序对 $p_{ij}[x]$ 中必然含操作 C_j , 因此可构成偏序环

$$\triangleright W_j W_j C_j [x] - W_j R_i C_j [y]$$

形式化定义为: $W_i [x_m] \dots W_j [x_{m+1}] \dots W_j [y_n] \dots R_i [y_n]$, 同(21).

$$\triangleright W_i R_j C_j [x] - W_j R_i C_j [y]$$

形式化定义为: $W_i [x_m] \dots R_j [x_m] \dots W_j [y_n] \dots R_i [y_n]$, 同(22).

$$\triangleright R_i W_j C_j [x] - W_j R_i C_j [y],$$

形式化定义为: $R_i [x_m] \dots W_j [x_{m+1}] \dots W_j [y_n] \dots R_i [y_n]$, 同(23).

6. 根据偏序对的定义, 偏序对 $RW = \{R_i W_j, R_i W_j C_j, R_i W_j C_i, R_i W_j A_i\}$, 则偏序环的组合方式为以下几种

1) 若 $p_{ji}[y] = R_j W_i$, 可构成偏序环

$$\triangleright W_i W_j [x] - R_j W_i [y]$$

形式化定义为: $W_i [x_m] \dots W_j [x_{m+1}] \dots R_j [y_n] \dots W_i [y_{n+1}]$, 即 Read-Write Skew2. (24)

$$\triangleright W_i R_j [x] - R_j W_i [y]$$

形式化定义为: $W_i [x_m] \dots R_j [x_m] \dots R_j [y_n] \dots W_i [y_{n+1}]$, 即 Read Skew2. (25)

$$\triangleright R_i W_j [x] - R_j W_i [y]$$

形式化定义为: $R_i [x_m] \dots W_j [x_{m+1}] \dots R_j [y_n] \dots W_i [y_{n+1}]$, 即 Write Skew. (26)

2) 若 $p_{ji}[y] = R_j W_i C_i$, 则偏序对 $p_{ij}[x]$ 中必然含操作 C_i , 因此可构成偏序环

$$\triangleright W_i W_j C_i [x] - R_j W_i C_i [y], \text{此时 } p_{ij}[x] = W_i W_j C_i \text{ 已经构成偏序环, 偏序环种类同(1).}$$

$$\triangleright W_i R_j C_i [x] - R_j W_i C_i [y]$$

形式化定义为: $W_i [x_m] \dots R_j [x_m] \dots R_j [y_n] \dots W_i [y_{n+1}]$, 同(25).

$$\triangleright R_i W_j C_i [x] - R_j W_i C_i [y], \text{此时 } p_{ij}[x] = R_i W_j C_i,$$

形式化定义为: $R_i [x_m] \dots W_j [x_{m+1}] \dots R_j [y_n] \dots W_i [y_{n+1}]$, 同(26).

3) 若 $p_{ji}[y] = R_j W_i C_j$, 则偏序对 $p_{ij}[x]$ 中必然含操作 C_j , 因此可构成偏序环

$$\triangleright W_i W_j C_j [x] - R_j W_i C_j [y],$$

形式化定义为: $W_i [x_m] \dots W_j [x_m] \dots R_j [y_n] \dots W_i [y_{n+1}]$, 同(24).

$$\triangleright W_i R_j C_j [x] - R_j W_i C_j [y]$$

形式化定义为: $W_i[x_m] \dots R_j[x_m] \dots R_j[y_n] \dots W_i[y_{n+1}]$, 同(25).

- $R_i W_j C_j[x] - R_j W_i C_j[y]$, 此时 $p_{ij}[x] = W_j W_i C_j$,

形式化定义为: $R_i[x_m] \dots W_j[x_m] \dots R_j[y_n] \dots W_i[y_{n+1}]$, 同(26).

- 4) 若 $p_{ji}[y] = R_j W_i A_j$, 则偏序对 $p_{ij}[x]$ 中必然含操作 A_j , 因此可构成偏序环

- $W_i W_j A_j[x] - R_j W_i A_j[y]$,

形式化定义为: $W_i[x_m] \dots R_j[x_m] \dots R_j[y_n] \dots W_i[y_{n+1}]$, 同(24).

- $W_i R_j A_j[x] - R_j W_i A_j[y]$, 此时事务 t_j 回滚了两个读操作, 无数据异常.

形式化定义为: $W_i[x_m] \dots R_j[x_m] \dots R_j[y_n] \dots W_i[y_{n+1}]$, 同(25).

- $R_i W_j A_j[x] - R_j W_i A_j[y]$, 此时 $p_{ij}[x] = W_j W_i C_j$,

形式化定义为: $R_i[x_m] \dots W_j[x_m] \dots R_j[y_n] \dots W_i[y_{n+1}]$, 同(26).