

## 一貫性と隔離性の細分に基づく並行実行の制御

徐 海燕<sup>†1</sup> 古川 哲也<sup>†2</sup> 史 一華<sup>†3</sup>

Web サービスのような一貫性を満たす部分トランザクションから構成されるトランザクションが直列可能でなくても正当であることを理論的に検証するため、われわれはトランザクションの一貫性を細分し、その上で隔離性も細分した正当性基準を提案している。本論文では、トランザクションの検索に関する隔離性、内部一貫性に関する隔離性、部分トランザクションの隔離性に基づく並行実行の制御方式を提案し、それを通して3つの隔離性の性質とそれらの関わり、および直列可能性との関係について検討する。具体的には、細分された3つの隔離性のための予約施錠、施錠、施錠緩和、解錠を行う施錠方式3ILを導入する。検索一貫性を満たさなくするような変更操作は予約R施錠によって、内部一貫性を満たさなくするような変更操作は、施錠緩和後のR施錠によって禁止している。3IL施錠方式は3つの隔離性を別々に扱っており、それぞれの特徴を個別に対処している。

### Concurrency Control Based on Dividing Consistency and Isolation Properties

HAIYAN XU,<sup>†1</sup> TETSUYA FURUKAWA<sup>†2</sup> and YIHUA SHI<sup>†3</sup>

By dividing consistency and isolation properties into the ones related to read, write and relationship, we have proved the following results of transactions that consist of Web services as their subtransactions: the class of the schedules which satisfy the divided three types of isolations each other is larger than the serializable class. This paper introduces a concurrency control protocol called three isolation locking (3IL), for the three isolations. In order to deal with the three kinds of isolations, 3IL protocol uses four kinds of actions: lock reservation, lock, lock degradation and unlock. By analyzing the characteristics of 3IL schedules, we show the roles of the three types of isolations and the feature of non-serializable but correct schedules.

### 1. はじめに

Web サービスのような一貫性を満たす部分トランザクションから構成されるトランザクションが直列可能でなくても正当であることを理論的に検証するため、われわれはトランザクションの一貫性の細分とそれに伴う隔離性の細分を行った正当性基準を提案している<sup>12)</sup>。具体的には、一貫性を、検索したデータが一貫していることを表す検索一貫性、終了時のデータベースが一貫している等価なスケジュールが存在することを表すDB一貫性、データ関連に関する一貫性に細分した上で、それぞれの一貫性のためのxR隔離性、S隔離性、xW隔離性を提案した。本論文では、xR隔離性、S隔離性、xW隔離性の3つの隔離性を保証できる並行実行の制御方式3ILを導入する。3IL施錠方式を用いて3つの隔離性の性質とそれらの関わり、および直列可能性との関係を明確にする。

トランザクション管理は、原子性、一貫性、隔離性、耐久性によって表されるACID属性に基づいている<sup>2)</sup>。トランザクションが隔離性を満たすこと、すなわち、並行実行が直列可能であることが一般的な正当性基準である。しかし、直列可能でない並行実行でも特に問題が生じない場合がある。次の旅行予約がその典型的な例である。

例1 ホテルと切符の予約からなる旅行予約で、ホテルと切符の両方を予約できれば支払を行うが、一方で予約できなければもう一方をキャンセルするといったように旅行予約が行われるとする。複数の人が同時に旅行予約を行い、隔離性を満たさなくても特に問題は生じない。 □

われわれは、一貫性の細分に伴って隔離性を検索に関するxR隔離性、部分トランザクション間の隔離性(S隔離性)、データ関連に関するxW隔離性に細分し、それぞれの隔離性に対する判定グラフを導入した<sup>12)</sup>。本論文では、その3つの隔離性を満たすことのできる施錠による制御方式3ILを導入し、その性質を分析することで次の結果を示す。

- 検索に関するxR隔離性は、データベースの一貫性制約に依存する。一貫性制約上で関連するデータ集合が異なる分割に分けられるならば、検索一貫性の単位は操作されるデータ項目の属する分割によって細分できる。

<sup>†1</sup> 福岡工業大学情報工学部情報工学科  
Dept. Comp. Sci. and Eng., Fukuoka Institute of Tech.

<sup>†2</sup> 九州大学大学院経済学研究院  
Dept. Economic Engineering, Kyushu University

<sup>†3</sup> 西南学院大学商学部  
Dept. Commerce, Seinan Gakuin University

- データ関連に関する  $xW$  隔離性は、各時点での実行結果がその後どのように利用されるかというトランザクションの構造に依存する。
  - 一貫したデータベースから一貫したデータベースへの遷移である部分トランザクションが存在する場合は、検索に関する  $xR$  隔離性、データ関連に関する  $xW$  隔離性の詳細に依存せず、直列可能でない正当なスケジュールが存在する。
- これまでの直列可能性の拡張に関する研究には、次のものがある<sup>4),6)</sup>。

## [意味論に基づく方式]

一貫性制約に基づいてデータベースのデータを一貫性制約上で関連する部分集合に分け、トランザクションもそれぞれの部分集合の中のデータを操作する部分トランザクションに分ける PW(Predicatewise) 直列可能性基準<sup>9)</sup> や、意味論に基づく競合関係の減少によって等価となるスケジュール集合を拡大する研究<sup>1)</sup> が行われている。

## [ユーザ定義条件に基づく方式]

検索されるデータ、実行結果に対して事前条件、事後条件を明記することによって、直列可能性を要求しない正当性基準が提案されている<sup>6),10)</sup>。

## [制御方式による対策]

マルチバージョン時刻印方式<sup>2)</sup>、後復帰の代わりに補償の部分トランザクションを利用する方式<sup>4)</sup>、CAD などの応用分野に対するアドホックなトランザクションモデル<sup>5)</sup> が提案されている。

## [Web サービスに対する並行処理制御]

Web サービスのような疎結合である結合環境においては、ACID 属性に基づくトランザクション管理の問題<sup>8)</sup> に対処するため、Business Transaction Processing (BTP)<sup>3)</sup> や、WS-Transaction<sup>11)</sup> が公開されている。長大トランザクションの形式的な記述などに関する研究も活発に行われている<sup>7)</sup>。

正当なクラスを拡大するこれまでの研究アプローチに対し、本論文は、一貫性の細分に伴う隔離性の細分によって、直列可能でないスケジュールが正当である理由を明らかにすることを目的としている。具体的に、意味論に基づくアプローチやユーザ定義条件に基づく方式と比較すると、一貫性制約に基づいて検索一貫性の単位と DB 一貫性の単位が細分できて、内部一貫性のための  $xW$  隔離性を満たすクラスには影響しない。一方、トランザクションに関する意味論で  $xW$  隔離性を満たすスケジュールが拡大しても、検索一貫性のための  $xR$  隔離性を満たすクラスには影響しない。制御方式による対策と比べると、それぞれの隔離性について検討している。さらに、Web サービスの特徴を制御方式へ活用することにつ

いても議論している。

本論文は、次のように構成される。2章でトランザクションモデルを定義する。3章では、検索に関する一貫性、更新に関する一貫性、データ関連に関する一貫性に細分された一貫性に対応する隔離性の定義とそれぞれの判定グラフを説明する。3つの隔離性を保証できる 3IL 制御方式を4章で導入する。5章では、3IL スケジュールの性質を分析することでそれぞれの隔離性の性質と3つの隔離性を合わせた性質を分析する。6章は全体のまとめである。

## 2. トランザクションモデル

本章では、Web サービスのような複数のタスクから構成されるトランザクションモデルについて述べる。

## 2.1 トランザクション

タスク  $S_i$  は、入力データの集合  $I_i$  と出力データの集合  $O_i$  に対する検索操作と変更操作からなる組  $\langle R_i(I_i), W_i(O_i) \rangle$  である。トランザクション  $T$  は、全順序関連  $<$  を持つタスク  $S_1, S_2, \dots, S_m$  の集合である。ただし、同じデータ項目に対する変更操作は1回のみである。したがって、 $O_i \cap O_j (i \neq j)$  は空であるが、 $I_i \cap I_j (i \neq j)$  は空であるとは限らない。また、トランザクションにおいて変更操作の前に必ず同じデータ項目に対する検索操作を行うとする。すなわち、 $O_i \subseteq \bigcup_{j=1}^i I_j$  である。

各トランザクションは一貫性を満たす、すなわち、一貫したデータベースで単独実行された結果のデータベースは一貫している。トランザクションが長大である場合など、中間結果のデータベースが一貫している場合もあり、本論文では、一貫したデータベースから一貫したデータベースへの遷移である部分トランザクションを、一貫性を満たす部分トランザクションと呼び、部分  $T$  で表す。

同じデータ項目に対する検索操作が複数のタスクに存在することがあり、 $S_i$  の出力データが後続の  $S_j (j > i)$  によって検索されることもある。本論文では、検索されるデータが祖先タスクによって検索または生成された結果であるかどうかを区別するために、それらを内部入力データ、その他を外部入力データと呼ぶ。内部入力データとなるデータの変更操作は、他のトランザクションとの競合関係に影響しないので、例題ではその前の検索操作を省略している場合がある。

例2 顧客 ID を  $x_1$ 、申請を  $x_2$ 、受付を  $x_3$ 、受付番号  $x_4$ 、処理を  $x_5$  で表すとする。ローン申請処理トランザクション  $T_1$  は、次のようにタスク  $S_{11}, S_{12}$  からなる。 $S_{11}$  までの実行結果が一貫するならば、 $\{S_{11}\}$  および  $\{S_{12}\}$  は部分  $T$  である。

$$\begin{aligned}
 & T_1 \\
 S_{11} & : \langle R_{11}\{x_1, x_2, x_3\}, W_{11}\{x_3, x_4\} \rangle \\
 S_{12} & : \langle R_{12}\{x_1, x_2, x_4, x_5\}, W_{12}\{x_5\} \rangle \quad \square
 \end{aligned}$$

$S_i$  の外部入力データに対する検索操作のみからなるタスクを  $S_i^R$ 、全順序関連  $<$  を持つタスク  $S_1^R, S_2^R, \dots, S_m^R$  の集合を  $T^R$  とする。例えば、 $T_1$  での  $S_{12}$  に対して  $x_5$  は外部入力データ、 $x_1, x_2, x_4$  は内部入力データなので、 $T_1^R$  は次のようになる。

$$\begin{aligned}
 & T_1^R \\
 S_{11}^R & : \langle R_{11}\{x_1, x_2, x_3\}, W_{11}(\phi) \rangle \\
 S_{12}^R & : \langle R_{12}\{x_5\}, W_{12}(\phi) \rangle
 \end{aligned}$$

## 2.2 スケジュール

トランザクションの並行実行を表すスケジュールを定義する。

定義 1 トランザクション集合  $T = \{T_1, T_2, \dots, T_n\}$  上のスケジュール  $H$  は、各  $T_i \in T$  の全順序関連  $<_i$  を保持した次のような全順序関係  $<_H$  を持つタスクの集合である。

- $H = \bigcup_i T_i$
- $<_H \supseteq \bigcup_i <_i$  □

タスク  $S_{is} \in T_i$  と  $S_{jt} \in T_j (i < j)$  に対して、 $I_{is} \cap O_{jt} \neq \phi$ 、 $O_{is} \cap O_{jt} \neq \phi$ 、 $O_{is} \cap I_{jt} \neq \phi$  ならば、 $S_{is}$  と  $S_{jt}$  間にはそれぞれ RW 競合、WW 競合、WR 競合が存在するという。

例 3 図 1 は顧客  $x_1$  と  $y_1$  がこの順で同一データ  $x_5$  に対する処理を申請し、逆の順で処理が行われるスケジュール  $H_1$  を示している。 $S_{11}$  と  $S_{21}$  が  $x_3$  に対する操作によって、 $S_{12}$  と  $S_{22}$  が  $x_5$  に対する操作によって競合している。 □

$$\begin{array}{ccc}
 T_1 & & T_2 \\
 S_{11} : \langle R_{11}\{x_1, x_2, x_3\}, W_{11}\{x_3, x_4\} \rangle & & \\
 & S_{21} : \langle R_{21}\{y_1, y_2, x_3\}, W_{21}\{x_3, y_4\} \rangle & \\
 & S_{22} : \langle R_{22}\{y_1, y_2, y_4, x_5\}, W_{22}\{x_5\} \rangle & \\
 S_{12} : \langle R_{12}\{x_1, x_2, x_4, x_5\}, W_{12}\{x_5\} \rangle & & 
 \end{array}$$

図 1 スケジュール  $H_1$   
Fig.1 Schedule  $H_1$

$T$  上の 2 つのスケジュール  $H$  と  $H'$  は、競合するタスク  $S_{is}$  と  $S_{jt} (i \neq j)$  間の順序関連がすべて同じであるならば、等価であるという<sup>2)</sup>。 $H$  はある直列スケジュールと等価であれば、直列可能であるという。

例 4 トランザクション  $T_3$  および  $T_4$  で、ホテル予約の予約数、回答数、支払金額を  $h_r, h_a, h_p$ 、および  $h'_r, h'_a, h'_p$  で、切符予約の予約数、回答数、支払金額を  $t'_r, t'_a, t'_p$ 、および  $t_r, t_a, t_p$  で表し、ホテルの利用可能数を  $h_b$ 、切符予約の利用可能数を  $t_b$  とする。旅行予約トランザクション  $T_3$  および  $T_4$  は、ホテル予約、ホテル予約結果確認、切符予約、ホテル支払、切符支払の 5 つのタスク  $S_{31}, S_{32}, S_{33}, S_{34}, S_{35}$  および  $S_{41}, S_{42}, S_{43}, S_{44}, S_{45}$  から構成され、旅行予約スケジュール  $H_2$  は、ホテル予約は  $T_3, T_4$  の順で、切符予約は  $T_4, T_3$  の順で行うとする (図 2)。 $S_{33}$  では  $h_a$  と  $t'_a$  が、 $S_{43}$  では  $h'_a$  と  $t_a$  が内部入力データである。また、各タスクの終了時のデータベースは一貫しており、各タスクはそれのみで部分 T となる。 □

$$\begin{array}{ccc}
 T_3 & & T_4 \\
 S_{31} : \langle R_{31}\{h_b, h_r\}, W_{31}\{h_b, h_a\} \rangle & & \\
 & S_{41} : \langle R_{41}\{h_b, h'_r\}, W_{41}\{h_b, h'_a\} \rangle & \\
 & S_{42} : \langle R_{42}\{h'_a\}, W_{42}(\phi) \rangle & \\
 & S_{43} : \langle R_{43}\{t_b, t_r\}, W_{43}\{t_b, t_a\} \rangle & \\
 S_{32} : \langle R_{32}\{h_a\}, W_{32}(\phi) \rangle & & \\
 S_{33} : \langle R_{33}\{t_b, t'_r\}, W_{33}\{t_b, t'_a\} \rangle & & \\
 S_{34} : \langle R_{34}\{h_a\}, W_{33}\{h_p\} \rangle & & \\
 & S_{44} : \langle R_{44}\{h'_a\}, W_{43}\{h'_p\} \rangle & \\
 S_{35} : \langle R_{35}\{t'_a\}, W_{35}\{t'_p\} \rangle & & \\
 & S_{45} : \langle R_{45}\{t_a\}, W_{43}\{t_p\} \rangle & 
 \end{array}$$

図 2 旅行予約スケジュール  $H_2$   
Fig.2 Reservation schedule  $H_2$

定義 2 トランザクション集合  $T$  上のスケジュール  $H$  に対して、 $T$  のトランザクション  $T_i$  によって操作されたデータ項目が、 $T_i$  が終了するまで他のトランザクションによって操作されていない等価なスケジュールが存在するならば、 $T_i$  は隔離性を満たすという。 $T$  のすべてのトランザクションが隔離性を満たすとき、 $H$  は隔離性を満たすという。 □

スケジュール  $H$  が隔離性を満たすことは、 $H$  が直列可能であることと等価であり、直列

可能性判定グラフで判定できる<sup>2)</sup>。

[直列可能性判定グラフ  $SG(H)$ ]

節点集合： $T$  の各  $T_i$  に対応する節点からなる。

枝集合：トランザクション  $T_i, T_j$  間に WR, WW, RW の競合が存在すれば、 $T_i$  から  $T_j$  へのラベル WR, WW, RW の枝が存在する。

$SG(H)$  で閉路に含まれるトランザクションは  $H$  において隔離性を満たさず、 $SG(H)$  に閉路が存在しないことはスケジュール  $H$  が直列可能であるための必要十分条件である。以降、直列可能なスケジュールからなるクラスを SR で表す。

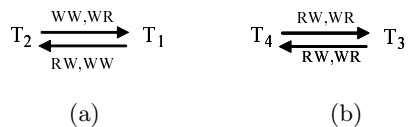


図3 直列可能性グラフ  $SG(H_1)$  と  $SG(H_2)$   
Fig. 3 Serialization graph  $SG(H_1)$  and  $SG(H_2)$

$SG(H_1), SG(H_2)$  をそれぞれ図 3(a), 図 3(b) に示している。 $T_1$  と  $T_2$  間、 $T_3$  と  $T_4$  間に閉路が存在するので、 $T_i (i = 1, 2, 3, 4)$  は隔離性を満たさず、 $H_1$  と  $H_2$  は直列可能ではない (SR に属さない)。

### 3. 一貫性と隔離性による正当性基準

本章では、論文<sup>12)</sup> で提案している検索に関する一貫性、更新に関する一貫性、データ関連に関する一貫性およびそれぞれに対応する隔離性について説明し、一貫したデータベースへの遷移である部分トランザクションが存在する場合における3つの隔離性のそれぞれの判定方法を述べる。

#### 3.1 正当性基準

トランザクションが一貫したデータベースで個別に実行されると、検索された外部入力データは一貫している。また、内部入力データはトランザクションのその他の操作されたデータ項目と一貫してしており、終了時のデータベースも一貫している。トランザクションの一貫性を、これに対応して細分する。

定義3 トランザクション集合  $T$  上のスケジュール  $H$  に対して、 $T$  のトランザクション  $T_i$  の実行は、検索した外部入力データが一貫しているとき検索一貫性を、終了時のデータ

ベースが一貫している等価なスケジュールが存在するとき DB 一貫性を、内部入力データが一貫しているとき内部一貫性を満たすといひ、検索一貫性、DB 一貫性、内部一貫性を満たすトランザクションの実行は正当であるという。

トランザクション集合  $T$  上のスケジュール  $H$  は、 $T$  のすべてのトランザクションの実行が正当であるとき、正当である。 □

トランザクションに関する意味論情報がない場合は、細分された一貫性のための隔離性は隔離性と等価である<sup>12)</sup>。

#### 3.2 検索一貫性と xR 隔離性

一貫性を満たす部分  $T$  が存在する場合、検索一貫性のための xR 隔離性は次のように定義される。

定義4<sup>12)</sup> トランザクション集合  $T$  上のスケジュール  $H$  に対して、 $T$  のトランザクション  $T_i$  の  $T_i^R$  によって検索されたデータ項目が、 $T_i^R$  が終了するまで  $T$  のどの部分  $T$  によっても変更されておらず、さらに、 $T_i^R$  がある部分  $T$  の実行結果を検索しているならばその部分  $T$  によって変更されたデータ項目が部分  $T$  が終了するまで  $T_i^R$  によって検索されていない等価なスケジュールが存在するとき、 $T_i$  は xR 隔離性を満たすという。 $T$  のすべてのトランザクションが xR 隔離性を満たすとき、 $H$  は xR 隔離性を満たすという。 □

[xR 隔離性判定グラフ  $xRI(H)$ ]

節点集合： $T$  の各  $T_i$  の部分  $T$  と  $T_i^R$  に対応する節点からなる。

枝集合：節点間の競合関係を枝とする。

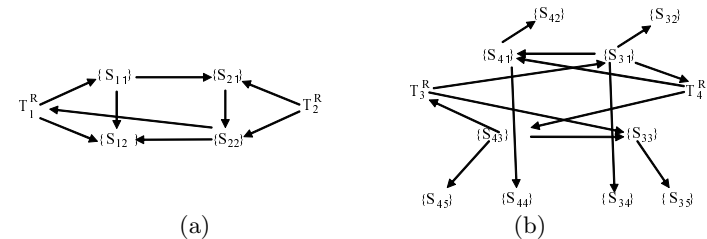


図4 xR 隔離性判定グラフ  $xRI(H_1)$  と  $xRI(H_2)$   
Fig. 4 xR-Isolation graph  $xRI(H_1)$  and  $xRI(H_2)$

例5  $T_1$  と  $T_2$  の各タスクをそのみで部分  $T$  となるとすると、 $xRI(H_1)$  には図 4(a) に示しているように  $T_1^R, \{S_{11}\}, \{S_{21}\}, \{S_{22}\}, T_1^R$  間に閉路が存在する。一方、 $T_3$  と  $T_4$  中の各タ

スクもそのみで部分 T となるので、 $xRI(H_2)$  は図 4(b) となる。 $T_3^R, \{S_{31}\}, T_4^R, \{S_{43}\}$  間には閉路が存在するが、 $T_3^R$  と部分 T からなる閉路や  $T_4^R$  と部分 T からなる閉路は存在しない。 □

補題 1 <sup>12)</sup> トランザクション集合  $T$  上のスケジュール  $H$  において、 $xRI(H)$  に 1 つの  $T_i^R$  と部分 T からなる閉路が存在しないことは、 $H$  が xR 隔離性を満たすための必要十分条件である。 □

一方、 $T$  上のスケジュール  $H$  に対して、部分 T が隔離性を満たすとき、 $H$  は S 隔離性を満たすという。xR 隔離性と S 隔離性を満たすスケジュールからなるクラスをそれぞれ、xRI と SSR とする。スケジュールの正当性は、部分 T の存在まで考えると、xR 隔離性と S 隔離性で規定できる。

補題 2 <sup>12)</sup> トランザクション集合  $T$  上のスケジュール  $H$  において、 $T$  のすべてのトランザクションが内部一貫性を満たしているとき、 $H$  が xR 隔離性と S 隔離性を満たすことは、 $H$  が正当であるための必要十分条件である。 □

### 3.3 内部一貫性と xW 隔離性

内部一貫性を検討するために、操作されたデータが他のトランザクションによって変更されたら、後に操作されるデータの一貫性に影響を与えることを操作間のデータ従属性として表す。検索されたデータが祖先タスクによって操作されたものであるとき、その値が一致しなければならない場合があり、祖先タスクの操作と内部入力データに対する検索操作は、データ従属性の候補となる。

$$(1)(R_i(x), R_j(x)) (i < j)$$

$$(2)(W_i(x), R_j(x)) (i < j)$$

変更操作は、そこまで検索された値に基づいていれば、同一タスク内の検索操作と変更操作もデータ従属性の候補となる。

$$(3)(R_i(x), W_i(y)) (R_i(x) < W_i(y))$$

また、(3) と (2) の推移関連もデータ従属性の候補である。以上が与えられたトランザクションから得られるすべての候補である。

例 6  $H_2$  においては、 $T_3(T_4)$  の意味論によりタスク間のデータ従属性は、予約結果を表す内部入力データ  $h_a, t'_a(h'_a, t'_a)$  に対する検索操作との種類 (2) の次の 3 つである。

$$\{(W_{31}(h_a), R_{32}(h_a)), (W_{31}(h_a), R_{34}(h_a)), (W_{33}(t'_a), R_{35}(t'_a))\}$$

$$\{(W_{41}(h'_a), R_{42}(h'_a)), (W_{41}(h'_a), R_{44}(h'_a)), (W_{43}(t_a), R_{45}(t_a))\} \quad \square$$

定義 5 <sup>12)</sup> トランザクション集合  $T$  上のスケジュール  $H$  に対して、 $T$  のトランザクション  $T_i$  によって実行された操作が、データ従属性のある操作の実行が終了するまで他のトランザクションによって変更されていない等価なスケジュールが存在するならば、 $T_i$  は xW 隔離性を満たすという。 $T$  のすべてのトランザクションが xW 隔離性を満たすとき、 $H$  は xW 隔離性を満たすという。 □

[xW 隔離性判定グラフ  $xWI(H)$ ]

節点集合： $T$  の各  $T_i$  のタスク  $S_{ij}$  に対応する節点からなる。

枝集合：次のような枝からなる。

- (1)  $S_{iu} <_i S_{iv}$  なら、 $S_{iu}$  から  $S_{iv}$  への枝が存在する。
- (2)  $S_{iu} <_H S_{jv}$  かつ  $S_{iu}$  と  $S_{jv}$  が競合するなら、 $S_{iu}$  から  $S_{jv}$  への枝が存在する。
- (3)  $S_{iu} <_H S_{jv}$  かつ  $S_{iu}$  と  $S_{jv}$  間に WW、RW 競合が存在するなら、 $S_{iu}$  中の競合している操作とデータ従属性を持つ後続のタスク中の最後のタスクから  $S_{jv}$  への枝が存在する。

例 7  $H_2$  においては、予約結果は他のトランザクションによって変更されていないので、 $xWI(H_2)$  には種類 (3) の枝が存在しない。したがって、 $xWI(H_2)$  に閉路は存在しない。 □

$T$  上のスケジュール  $H$  において、 $xWI(H)$  が非巡回であることは、 $H$  が xW 隔離性を満たすための必要十分条件である<sup>12)</sup>。一方、 $T$  のトランザクション  $T_i$  によって実行された操作が、 $T_i$  が終了するまで他のトランザクションによって変更されていない等価なスケジュールが存在するならば、 $T_i$  は W 隔離性を満たすという。明らかに、 $T_i$  が W 隔離性を満たせば xW 隔離性を満たす。 $T$  のすべてのトランザクションが内部一貫性を満たすための必要十分条件は、 $H$  が xW 隔離性を満たすことである<sup>12)</sup>。xW 隔離性、W 隔離性を満たすスケジュールからなるクラスをそれぞれ、xWI、WI とする。

スケジュールの正当性は、xR 隔離性、S 隔離性、xW 隔離性で表すことができる。

定理 1 <sup>12)</sup> トランザクション集合  $T$  上のスケジュール  $H$  において、 $H$  が xR 隔離性、S 隔離性、xW 隔離性を満たすことは、 $H$  が正当であるための必要十分条件である。 □

## 4. xR 隔離性、S 隔離性と xW 隔離性のための施錠方式 3IL

xRI クラスの性質は 5 章で検討するが、本論文ではまず、トランザクションの意味論に

よって  $T_i^R$  がいくつかの検索一貫性の単位に分けられるという仮定の下で検討を進める。すなわち、各検索一貫性の単位の外部入力データが一貫したデータベースで検索していれば、検索一貫性を満たす。ただし、異なる検索一貫性の単位の共通部分は空とは限らないとする。

$xR$  隔離性、 $S$  隔離性、 $xW$  隔離性という3つの隔離性に対する施錠方式を与える。

定義6 次のルールにしたがって予約施錠、施錠、施錠緩和、解錠を行う方式を、3隔離性施錠方式(3IL)という。

ルール1: 検索一貫性の単位の開始時に検索予定のデータ項目に予約R施錠をする。

ルール2: 部分Tは開始時に入力データ項目にR施錠、出力データ項目にW施錠を行う。すでにW施錠されているデータ項目には他のトランザクションが再び施錠することはなく、すでにR施錠されているデータ項目には他の部分Tが再びW施錠をすることはない。部分Tが終了する時に、W施錠はデータ従属性があればR施錠に緩和し、なければ解錠する。R施錠はデータ従属性がなければ解錠する。検索したデータ項目に対する予約R施錠は解錠する。

ルール3: 変更操作が終了した時に、および変更操作を伴わない検索操作が終了した時に、所属する部分TのIDと操作種類を表すタグ  $\langle ID, W \rangle$ 、または  $\langle ID, R \rangle$  をデータ項目のタグリストに記録する。

部分TのIDリストLの初期値は、所属する検索一貫性の単位のIDである。部分Tの開始時に変更するデータ項目に対しては、そのデータ項目のタグリストに属する部分TのIDリストをLに、検索するデータ項目に対しては、そのデータ項目のタグリスト中の操作種類がWである部分TのIDリストをLに加える。

検索一貫性の単位の終了時に、部分TのIDリストからそのIDを含むタグを削除する。

ルール4: 予約R施錠がされているデータ項目には、IDリストに所属する検索一貫性の単位のIDが含まれている部分TによるW施錠をすることはない。 □

3IL方式に従うスケジュールHを3ILスケジュールという。データ項目のタグリストには、最後に行った変更操作の所属する部分TのIDを示すタグ  $\langle ID, W \rangle$  とそれより後に行われた検索操作の所属する部分TのIDを示すタグ  $\langle ID, R \rangle$  が記録されている。部分Tのタグリストには、 $XRI(H)$  判定グラフに自分より先行する検索一貫性の単位のIDが記録されている。

例8  $H_1$  に対しては、 $T_1$  の開始時に  $x_5$  に  $T_1^R$  による予約R施錠がされ、 $S_{11}$  の終了時にデータ項目  $x_1$  から  $x_4$  のタグリストは次のようになる。

$$\begin{aligned} x_1 &: \langle \{S_{11}\}, R \rangle \\ x_2 &: \langle \{S_{11}\}, R \rangle \\ x_3 &: \langle \{S_{11}\}, W \rangle \\ x_4 &: \langle \{S_{11}\}, W \rangle \end{aligned}$$

部分  $T\{S_{11}\}$  のIDリストは、 $\{T_1^R\}$  である。 $S_{21}$  が実行される時、 $x_3$  に対する変更操作により  $\{S_{11}\}$  のIDリストが加えられ、 $\{T_1^R, T_2^R\}$  になる。 $S_{21}$  の終了時にデータ項目のタグリストは次のようになる。

$$\begin{aligned} x_1 &: \langle \{S_{11}\}, R \rangle \\ x_2 &: \langle \{S_{11}\}, R \rangle \\ x_3 &: \langle \{S_{21}\}, W \rangle \\ x_4 &: \langle \{S_{11}\}, W \rangle \\ y_1 &: \langle \{S_{21}\}, R \rangle \\ y_2 &: \langle \{S_{21}\}, R \rangle \\ y_4 &: \langle \{S_{21}\}, W \rangle \end{aligned}$$

$S_{22}$  が実行される時、 $y_4$  に対する検索操作により  $S_{11}$  の属する部分TのIDリストが加えられ、 $\{T_1^R, T_2^R\}$  になる。 $x_5$  に  $T_1^R$  による予約R施錠がされているので、 $S_{22}$  の実行が待たされる。図5に3ILにしたがって実行されたスケジュールが示されている。 □

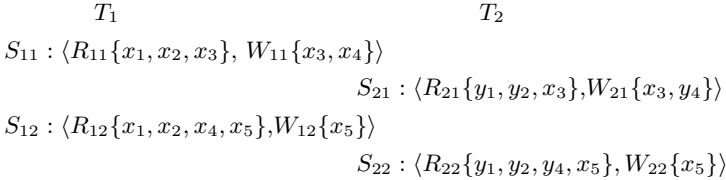


図5  $H_1$  に対する実行された3ILスケジュール  
Fig.5 3IL Schedule of  $H_1$

例9  $H_2$  では、例6に示されているデータ従属性より、 $S_{31}$  終了時に  $h_b$  に対するW施錠は解錠され、 $h_a$  に対するW施錠はR施錠に緩和されるので、 $S_{41}, S_{42}$  が実行できる。 $\{S_{43}\}$  のIDリストは  $\{T_4^R\}$  であるので、 $S_{43}$  の実行が  $T_3^R$  の  $t_b$  に対する予約R施錠によって禁

止されていない。したがって、 $H_3$  は 3IL スケジュールである。 □

$H_2$  において、次のように  $S_{43}$  がデータ項目  $h_b$  に関する検索操作を含む ( $S_{43} : \langle R_{43}\{h_b, t_b, t_r\}, W_{43}\{t_b, t_a\} \rangle$ ) であれば、結論は異なる。これは  $h_b$  による  $S_{41}$  と  $S_{43}$  の競合関係によって  $\{S_{43}\}$  の ID リストに  $T_3^R$  の ID が含まれるためである。 $\{S_{43}\}$  の ID リストに  $T_3^R$  の ID が含まれると、 $S_{43}$  の実行が  $T_3$  の  $t_b$  に対する予約 R 施錠によって禁止される。

次の例は、予約 R 施錠による W 施錠の許可および禁止を示している。

例 10 図 6 に示している  $H_3$  は、操作されているデータ項目が他のトランザクションによって変更されていないため、WI クラスに属する。各タスクを部分 T、 $\{S_{51}, S_{52}\}$  と  $\{S_{51}, S_{53}\}$  をそれぞれ検索一貫性の単位とする。 $S_{51}$  の開始時に、属しているそれぞれの検索一貫性の単位の検索予定のデータ項目  $z_1, z_2$  と  $z_1, z_3$  に予約 R 施錠をする。 $S_{51}$  の終了時には  $z_1$  に対する予約 R 施錠が、 $S_{52}$  の終了時には  $z_2$  に対する予約 R 施錠が解錠される。一方、部分 T  $\{S_{61}\}$  の ID リストに検索一貫性の単位  $\{S_{51}, S_{53}\}$  の ID が含まれていないため、 $\{S_{51}, S_{53}\}$  による  $z_3$  に対する予約 R 施錠によって、部分 T  $\{S_{61}\}$  による  $z_3$  に対する W 施錠は禁止されない。したがって、 $H_3$  は 3IL スケジュールである。 □

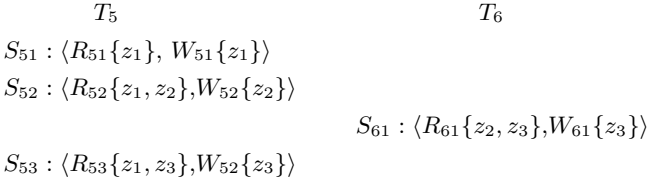


図 6 3IL スケジュール  $H_3$   
Fig. 6 3IL Schedule  $H_3$

例 11 図 7 に示している  $H_4$  に対して、各タスクを部分 T、 $\{S_{71}, S_{73}\}$  と  $\{S_{72}, S_{73}\}$  をそれぞれ検索一貫性の単位とすると、 $S_{72}$  と  $S_{81}$  の  $z_3$  に対する競合によって部分 T  $\{S_{81}\}$  の ID リストに、 $\{S_{72}, S_{73}\}$  の ID が含まれる。 $S_{81}$  の実行は  $\{S_{72}, S_{73}\}$  の開始時に  $z_3$  に対する予約 R 施錠によって禁止されるので、 $H_4$  は 3IL スケジュールではない。 □

$H_4$  から分かるように、3IL 方式における予約 R 施錠によっては、検索一貫性の単位が検索一貫性を満たさなくなるような変更操作の W 施錠を禁止している。一般には、次の性質

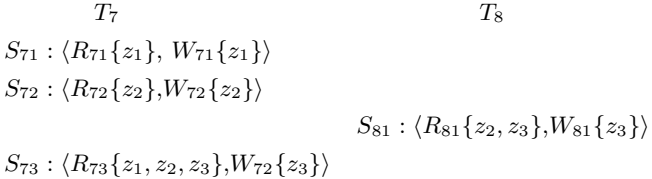


図 7 非 3IL スケジュール  $H_4$   
Fig. 7 non-3IL Schedule  $H_4$

がある。

補題 3 トランザクション集合  $T$  上の 3IL スケジュールは、 $xRI$  に属する。 □

証明: 判定グラフ  $xRI(H)$  で検索一貫性の単位からの経路中の部分 T には、検索一貫性の単位の ID を記録している。一方、検索一貫性の単位への枝がある部分 T とは WR 競合が存在する。判定グラフ  $xRI(H)$  に検索一貫性の単位を含む閉路が存在することは、検索一貫性の単位の ID を記録している部分 T が、検索一貫性の単位の検索予定のデータ項目を変更することになる。ルール 4 によって禁止されているので、矛盾である。 □

補題 4 トランザクション集合  $T$  上の 3IL スケジュールは、 $xWI$  に属する。 □

証明: ルール 2 によって操作されたデータがデータ従属性のある操作が終了するまで、W 施錠または R 施錠がなされている。変更操作の前には W 施錠がされ、すでに施錠がされているデータ項目に W 施錠はできないので、3IL スケジュールにおいては、操作されたデータがデータ従属性のある操作が終了するまで他のトランザクションによる変更はできない。このため、補題が成り立つ。 □

補題 5 トランザクション集合  $T$  上の 3IL スケジュールは、 $SSR$  に属する。 □

証明: ルール 2 によって部分 T における操作に対しては、部分 T の開始から終了まで、検索されるデータ項目には R 施錠、変更されるデータ項目には W 施錠が行われるので部分 T 間の直列可能性が保証される。このため、補題が成り立つ。 □

補題 3、補題 4、補題 5 を総合すると、次の結果が成り立つ。

定理 2 トランザクション集合  $T$  上の 3IL スケジュール  $H$  は、正当である。 □

5. 3IL スケジュールの性質

本章では、3IL スケジュールを通して 3 つの隔離性の性質について分析する。

5.1 xR 隔離性について

xR 隔離性について、まず次の結果がある。

系 1 検索されるデータが内部入力データであるような検索操作のみなるタスクの存在は、スケジュールが  $xRI$  に属するかどうかには影響しない。 □

証明：そのようなタスクは  $xRI$  判定グラフにおいて入る枝しか持たない節点になるので、閉路の有無には関わらない。 □

$H_2$  における予約結果確認タスク  $S_{32}$  と  $S_{42}$  がそのようなタスクの例である。Web サービスのようなフロー構造を持つアプリケーションに対して、系 1 はフローの条件分岐に対応する部分は、トランザクションの検索一貫性には関わらないことを示している。

一方、部分 T が事前に定義されている場合、同一の部分 T によって操作されるデータ間には関連が存在し、推移的な関連を持つデータ項目は同一の分割に属するならば、次の結論が成り立つ。

系 2 操作するデータ項目が異なる分割に属する部分 T は、同じ検索一貫性の単位に含まれるかどうかは、 $H$  が  $xRI$  に属するかどうかには影響しない。 □

証明： $xRI$  判定グラフにおいて、異なる分割に属するデータ項目を操作する部分 T に対応する節点間には経路は存在しない。したがって、同一の検索一貫性の単位に含まれるかどうかは閉路の有無には関わらない。 □

$H_2$  におけるホテル予約と切符予約のタスクがその例である。系 1 と系 2 は、部分 T が定義されれば、検索一貫性の単位は、データベースの一貫性制約に関連するが、内部入力データしか持たないタスクなどのトランザクションの構造には無関係であることを示している。

さらに、補題 1 より、検索一貫性の単位の定義方法について、次の性質がある。

系 3 内部入力データを持つ部分 T、 $S_{i1}(I_{i1}, O_{i1})$  と  $S_{i2}(I_{i2}, O_{i2})(O_{i1} \cap I_{i2} \neq \phi)$  が同一の検索一貫性の単位に含まれている場合、 $I_{i2} - O_{i1}$  中のデータ項目は  $S_{i1}$  の開始時のデータベースのデータを検索していなければならない。一方、同一の検索一貫性の単位に含まれない場合、 $I_{i2} - O_{i1}$  中のデータ項目は  $S_{i1}$  の終了後のデータベースのデータを検索していなければならない。 □

証明： $S_{i1}(I_{i1}, O_{i1})$  と  $S_{i2}(I_{i2}, O_{i2})(O_{i1} \cap I_{i2} \neq \phi)$  が同一の検索一貫性の単位に含まれている場合、補題 1 より、 $S_{i2}$  中の外部入力データ  $I_{i2} - O_{i1}$  は、検索一貫性の単位の開始時、すなわち、 $S_{i1}$  の開始時の一貫したデータベースのデータを検索していなければならない。

一方、同一の検索一貫性の単位に含まれない場合、 $I_2$  中のデータ項目がすべて外部入力

データになり、 $I_{i2} - O_{i1}$  も  $S_{i2}$  開始時、すなわち、 $S_{i1}$  の終了後のデータベースのデータを検索していなければならない。 □

したがって、内部入力データが存在する部分 T に対しては、検索一貫性の単位は、トランザクションの意味論によって定まる。

逆に、検索一貫性の単位を不必要に大きく定義すると、xR 隔離性を満たさなくなる場合がある。

系 4 内部入力データを持つ部分 T、 $S_{i1}(I_{i1}, O_{i1})$  と  $S_{i2}(I_{i2}, O_{i2})(O_{i1} \cap I_{i2} \neq \phi)$  が同一の検索一貫性の単位に含まれず、代わりに推移的な内部入力データを持つ部分 T、 $S_{i1}(I_{i1}, O_{i1})$  と  $S_{ij}(I_{ij}, O_{ij})$  が一つの検索一貫性の単位に含まれると、そのスケジュール  $H$  は  $xRI$  に属さず、3IL スケジュールでもない。 □

証明： $xRI(H)$  判定グラフに、検索一貫性の単位  $\{S_{i1}^R, S_{ij}^R\}$  から  $S_{i1}$  の属する部分 T への枝と、 $S_{i1}$  の属する部分 T から  $S_{ij}$  の属する部分 T への経路と、 $S_{ij}$  の属する部分 T から  $\{S_{i1}^R, S_{ij}^R\}$  への枝によって、検索一貫性の単位  $\{S_{i1}^R, S_{ij}^R\}$  を含む閉路が生じる。補題 1 より、 $H$  は xR 隔離性を満たさない。一方、3IL を用いた場合  $S_{i1}(I_{i1}, O_{i1})$  と  $S_{ij}(I_{ij}, O_{ij})$  からなる検索一貫性の単位の開始時に、 $I_{ij}$  に対する予約 R 施錠が行われる。それによって他のトランザクションの  $O_{ij-1} \cap I_{ij}$  に対する W 施錠が禁止されるので、3IL スケジュールではない。 □

したがって、検索一貫性の単位は、内部入力データは推移的ではない部分 T からなるものに限るという結論になる。

5.2 xW 隔離性について

xW 隔離性については、系 1 とは逆の結果となる。

系 5 検索されるデータが内部入力データであるような検索操作のみなるタスクの存在は、スケジュールが  $xWI$  に属するかどうかに影響する。 □

証明：内部入力データに対する検索操作と、祖先タスクのそれに対する操作間にはデータ従属性が存在しうる。さらに、推移律によってさらに先行の操作とデータ従属性が存在しうる。したがって、内部入力データに対するタスクの存在によって、データ従属性のある後続のタスクが後方になることがあるため、補題が成り立つ。 □

審査などの場合、元のデータが変更されると、得られた審査結果に影響を及ぼす可能性がある。元々のデータに対する検索操作と審査結果に対する検索操作間にデータ従属性が存在する。そのようなデータ従属性は推移律によるものである。推移律を満たすものが存在す



るかどうかは、 $xW$  隔離性の制御方法にも影響する。

系 6  $xW$  隔離性に対して、マルチバージョンを利用した制御法を用いることができるのは、データ従属性に推移律によるものが存在しない場合に限る。 □

証明：種類 (1) と (2) のデータ従属性は、祖先タスクの操作と内部入力データに対する検索操作間のものであり、同一データ項目に対する操作間のものである。内部入力データに対しては祖先タスクの操作されたバージョンを検索させる方法で、 $xW$  隔離性が満たされることになるが、種類 (3) のデータ従属性との推移律によるものが存在すれば、データ従属性は異なるデータ項目に対する操作間に広がる。マルチバージョン法では対処できなくなる。 □

5.3 3IL スケジュールについて

$xRI \cap SSR \cap xWI$  に属するスケジュールは正当であるが、 $xRI$  に属するかどうかは検索一貫性の単位に関係しており、 $xWI$  に属するかどうかはデータ従属性というトランザクションの構造に関係する。しかし、部分 T が存在すれば、直列可能でない 3IL スケジュールは存在する。次の  $H_5$  がその例である。 $H_5$  は  $W$  隔離性と  $S$  隔離性を満たしており、 $T_9$  の各タスクが部分 T であれば  $H_5$  は  $xR$  隔離性も満たすので、正当である。

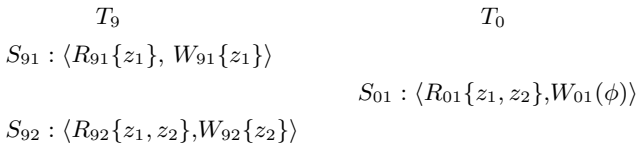


図 8 直列可能でない 3IL スケジュール  $H_5$   
Fig. 8 non-serializable but 3IL schedule  $H_5$

一方、検索一貫性は必ず満たさなければならないものではなく、満たさないときには検索一貫性の単位の動的再構成を行うならば、楽観的な制御方式も可能である。

6. ま と め

本論文では、部分トランザクションが一貫性を満たす場合は、トランザクションの検索に関する隔離性 ( $xR$  隔離性)、内部一貫性に関する隔離性 ( $xW$  隔離性)、部分トランザクションの隔離性 ( $S$  隔離性) に基づいて 3IL 制御方式を導入した。さらに 3IL スケジュールの性質について分析することにより、各々の隔離性の性質を明確にした。 $xR$  隔離性は外部入力データの一貫性、 $xW$  隔離性は内部入力データの一貫性に関わっている。外部入力データ

の一貫性を満たさなくするような変更操作は  $xR$  隔離性のための予約 R 施錠によって、内部入力データの一貫性を満たさなくするような変更操作は、 $xW$  隔離性のための R 施錠によって禁止されている。3IL 施錠方式は 3 つの隔離性を別々に扱っており、それぞれの特徴を個別に対処している。楽観的な制御方式やトランザクションの定義方法に関する研究は今後の課題である。

参 考 文 献

- 1) Agrawal, D., Abadi, A. E., and Singh, A. K.: Consistency and Orderability: Semantics-Based Correctness Criteria for Databases, *ACM Trans. Database Syst.*, Vol. 18, No. 3, pp. 460-486 (1993).
- 2) Bernstein, P. A., Hadzilacos, V., and Goodman, N.: *Concurrency Control and Recovery in Database Systems*, Addison-Wesley (1987).
- 3) Business Transaction Protocol (BTP). [http://www.oasis-open.org/committees/tc\\_home.php?wg\\_abbrev=business-transaction](http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=business-transaction) (2002).
- 4) Elmagarmid, A. K. (ed.): *Database Transaction Models for Advanced Applications*, Morgan Kaufmann (1992).
- 5) Kim, W., Lorle, R., McNabb, D., and Plouffe, W.: A Transaction Mechanism for Engineering Design Databases, *Proc. 10th Conf. on Very Large Databases* (1984).
- 6) Korth, H. F. and Speegle, G. S.: Formal Model of Correctness without Serializability, *ACM SIGMOD Record*, Vol. 17, Iss. 3, pp. 379-386 (1988).
- 7) Lanotte, R., Maggiolo-Schettini, A., Milazzo, P., and Troina, A.: Modeling Long-Running Transactions with Communicating Hierarchical Timed Automata, *Lecture Notes in Computer Science*, Vol. 4037, pp. 108-122 (2006).
- 8) Little, M.: Transactions and Web Services, *Comm. ACM*, Vol. 46, No. 10, pp. 49-54 (2003).
- 9) Ramamritham, K. and Chrysanthis, P. K.: A Taxonomy of Correctness Criteria in Database Applications, *VLDB Journal: Very Large Data Bases*, Vol. 5, No. 1, pp. 85-97 (1996).
- 10) Reuter, A. and Schwenkreis, F.: ConTracts - A Low-Level Mechanism for Building General-Purpose Workflow Management Systems, *IEEE Data Eng. Bulletin*, Vol. 18, pp. 4-10 (1995).
- 11) Web Services Transaction (WS-Transaction). <http://www-106.ibm.com/developerworks/webservices/library/ws-transpec/> (2002).
- 12) 徐海燕, 古川哲也, 史一華: 一貫性と隔離性の細分による並行実行の正当性の検証, 情報処理学会論文誌データベース Vol. 2 No. 1 pp. 22-32 (2009).