

データベースシステムにおける 2 レベルのデータ単位を
考慮した直列化可能性について

和田 聖治 (Seiji WADA)[†], 西尾 章治郎 (Shojiro NISHIO)[†],
長谷川 利治 (Toshiharu HASEGAWA)[†]

[†] 京都大学工学部数理工学科

1. はじめに

データベースシステムにおける並行処理制御に関する従来の研究の多くにおいては, 各トランザクションからの読取り, 書込み命令は, 例えば $\text{read}(x)$, $\text{write}(y)$ と書かれ, x , y は操作命令の対象となるデータ項目を表わすものとして用いられている ([3],[9]). ところが, これら x , y で表わされるデータ項目が実際にデータベースにアクセスする際の物理的なデータ単位 (例えば, ページとかレコードという単位) を表わすものなのか, それとも利用者が参照あるいは更新しようとしている論理的なデータ単位 (例えば, 銀行口座の預金高を格納しているデータの単位) なのかを明確に示さないまま, 並行処理制御を論ずるものがほとんどであった. しかし, より厳密に並行処理制御の問題を論ずるには, 利用者がアクセスの対象とする論理データ単位と, その論理データ単位にアクセスするためにデータベース管理システムがデータベースに対して実際にアクセスする物理データ単位とを区別すべきであり, これら 2 レベルのデータ単位をもとにしたトランザクションモデルについて並行処理制御の正当性を論ずることは実システムへの応用を考えるうえでも重要である. これらをアクセスするレベルを明確にしたうえでの並行処理制御を考えることの重要性は多くの文献で論じられており, 例えば文献 [1] では, 実際の関係データベースシステムである System R を対象とした議論がなされている.

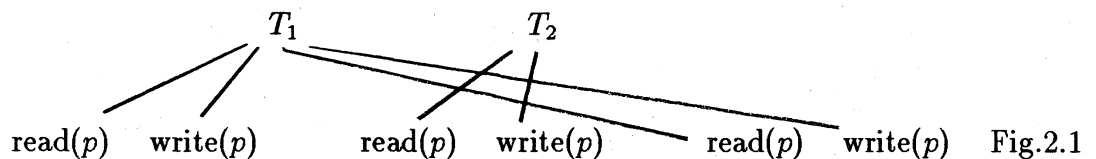
データベースシステムの並行処理制御の正当性を保証するものとして, 一般的に直列化可能性の概念 ([5],[10]) が用いられている. ここで, 直列化可能とは, 複数のトランザクションから発せられた操作命令をインターリーブして実行した際の結果が, それらの操作命令をトランザクションごとに直列に実行した結果と等価であることをいう. 今までに, 複数レベルのトランザクションあるいは複数レベルのデータ単位を考慮した直列化可能性の研究はなされているが ([2],[4],[7],[11]), それらの研究では, 論理データ単位, 物理データ単位という二つのデータのレベルを明確に意識した議論はない. 特に, 文献 [11] では, 2 レベル以上のデータ単位を考慮したトランザクションモデルが提案されているが, 直列化可能性に関する厳密な議論は行なわれていない.

本稿では, 最初に, 2 レベルのデータ単位を考慮したトランザクションモデルを導入する. 次に, そのトランザクションモデルをもとにした直列化可能性について論ずる.

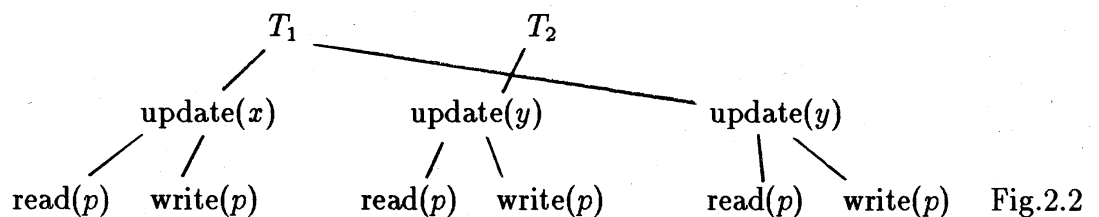
特に、本稿では従来の1レベルのデータ単位しか考慮しないモデルにおいて、直列化可能性の判定が多項式時間で可能なクラスとして重要であった競合保存直列化可能性[1] (D-直列化可能性[9], WW-直列化可能性[6]ともよばれる) の概念を2レベルのデータ項目を考慮したトランザクションモデルに拡張し、その直列化可能性のクラスに対する効率の良い判定条件を示す。

2. 2レベル直列化可能性

データベースシステムにおいては、実際のアクセスは物理データ単位に対する操作命令であることから、仮に従来の1レベルトランザクションの操作命令を物理データ単位に対するものとみなして直列化可能性を考えた場合、2レベルのデータ単位を考慮したトランザクションモデルでは本質的に直列化可能であるスケジュールも直列化可能でないと判定される可能性がある。例えば、1レベルトランザクションに対して、次のようにインターリーブされた物理レベルの操作命令の(左から右への)入力系列(スケジュール)は、特に $\text{write}(p)$ によって物理データ単位 p のどの部分が書換えられるかが不明な場合には、トランザクションを直列に実行した結果に等しくならない可能性があり、直列化可能ではないと考えられた。ところが、同一の物理データ



単位に対する操作命令が実際には別々の論理データ単位に対する操作命令であり、次のような構造をもっているとしよう。このとき、図 2.1のようにインターリーブされ



た操作命令の系列は本質的には直列化可能であると考えてよい。なぜなら、図 2.2の入力系列は T_2-T_1 の順にそれぞれのトランザクションを単独に実行した場合と実行結果が“等価”だからである。一方、 x, y を論理データ単位と考え、実際にアクセスされる物理データ単位を無視した並行処理制御をすると問題が起こる場合がある。以下では、複数の2レベルのデータ単位を考慮したトランザクション(2レベルトランザクション)から得られる図 2.2のような入力系列を2レベルスケジュールとよび、2レベルスケジュールの直列化可能性について論ずる。

論理データ単位に対する操作命令を L-アクション，物理データ単位に対する操作命令を P-アクション とよぶ。また，データベースに対するアクセスにより，データベース中のデータの値に変更が起こる可能性がある場合，このアクセスを タイプ W のアクセスであるといい，その可能性がないアクセスを タイプ R のアクセスという。さらに，あるアクションがタイプ R のアクセスのみから構成される場合，このアクションをタイプ R のアクションであるといい，少なくとも 1 つのタイプ W のアクセスをもつアクションをタイプ W のアクションという。本稿では，タイプ W の L-アクションとしては $\text{update}(x)$ ，タイプ R の L-アクションとしては $\text{fetch}(y)$ を考える（ただし， x, y は論理データ単位である）。また，タイプ W の P-アクションとしては $\text{write}(p)$ ，タイプ R の P-アクションとしては $\text{read}(q)$ を考える（ただし， p, q とは物理データ単位である）。トランザクション T に対して， T に対応する L-アクションの集合を $\text{L-act}(T)$ で示す。また，L-レベルのアクション a に対応する P-アクションの集合を $\text{P-act}(a)$ と記す。

次に，競合 の概念について述べる。あるスケジュールにおいて二つの L-アクションが可換でないときこれらは L-競合 しているといい，あるスケジュールにおいて二つの P-アクションが可換でないときこれらは P-競合 しているという。ただし，ある二つのアクション a_1, a_2 が可換でないとは，直感的にはこれらのアクションを $a_1 - a_2$ の順に実行した結果と， $a_2 - a_1$ の順に実行した結果が異なる可能性がある場合をいう。さらに二つの L-アクション a_1, a_2 に対し， $\text{P-act}(a_1)$ と $\text{P-act}(a_2)$ のうちで少なくとも一つのペアが P-競合している場合， a_1 と a_2 は L-準競合 しているという。

あるスケジュールの中で，二つの P-アクション b と b' に対して， b が b' よりも先に実行されるとき $b <_P b'$ と記す。また，二つの L-アクション a と a' との間の先行関係は $<_L$ で示され，一般には半順序関係である。

定義 2.1 [2 レベルスケジュール]

ある 2 レベルスケジュール s は，次の三つが成立するときかつそのときに限り 妥当 (well-formed) であるといわれる。

- (1) $<_P$ が全順序関係をもつ。
- (2) L-競合した L-アクション a_1, a_2 の間には，必ず先行関係 $a_1 <_L a_2$ または $a_2 <_L a_1$ のどちらか一方だけが成立する。
- (3) 二つの L-アクション a_1, a_2 に対して，次の関係が成立する。

$$a_1 <_L a_2 \implies \forall b_1 \in \text{P-act}(a_1), \forall b_2 \in \text{P-act}(a_2) \quad (b_1 <_P b_2). \quad \square$$

以下では特に断わらない限り妥当な 2 レベルスケジュールに関してのみ議論を行なう。スケジュールの正当性を議論するには，そのスケジュールと直列スケジュールとの

等価性を考える必要がある。直感的には、2レベルスケジュールが直列（2レベル直列）であるとは、そのスケジュールがL-アクション、P-アクションの両方で一切のインターリーブを許していないことを言う。

定義 2.2 [L-準先行関係]

異なるL-アクション a_1, a_2 が、ある物理データ単位 p に関してL-準競合しているとき、 p に関するL-準先行関係 $\Rightarrow_L^{(p)}$ が定義できる。

$$a_1 \Rightarrow_L^{(p)} a_2 \iff \exists b_1 \in P\text{-act}(a_1), \exists b_2 \in P\text{-act}(a_2) \\ \text{s.t. } (b_1, b_2 \text{ は } p \text{ に関して } P\text{-競合}) \wedge b_1 <_P b_2. \quad \square$$

定義 2.3 [トランザクション間先行関係]

2レベルトランザクション T_1, T_2 に関して、トランザクション間先行関係 $\Rightarrow_T$ を次のように定義する。

$$T_1 \Rightarrow_T T_2 \iff \exists a_1 \in L\text{-act}(T_1), \exists a_2 \in L\text{-act}(T_2) \\ \text{s.t. } (a_1, a_2 \text{ は } L\text{-競合}) \wedge a_1 <_L a_2 \quad \square$$

次に、スケジュールの等価性についての議論を行う。1レベルスケジュールに対して良く知られている競合保存等価性（以下、CP等価性と略す）を拡張して、2レベルスケジュールに対するCP等価性の概念を考える。二つの2レベルスケジュールがCP等価であるには、まずL-アクションの競合が保存されなければならないと考えるのは、1レベルの場合を拡張する意味から自然である。しかし、P-アクションでも競合が完全に保存されていることを要求するのは、制約が強過ぎることを図2.2の例で考えてみよう。図2.2において、L-アクション、P-アクションはそれぞれ左から右の順に $<_L$, $<_P$ 関係にあるとする。このスケジュールは、P-アクションについては競合は保存されなくても、直列スケジュール T_2-T_1 と等価であると考えられる。あるスケジュールに関し、すべての物理データ単位 p に対して $\Rightarrow_L^{(p)}$ によるサイクルが存在しなければ、そのスケジュールによる実行はすべての物理単位 p に関して正当であることを意味している。それゆえ、直列化可能性を考える場合には、あるスケジュールに $\Rightarrow_L^{(p)}$ によるサイクルがなければ、それに等価なスケジュールにおいても $\Rightarrow_L^{(p)}$ によるサイクルが存在してはならないという条件を入れるべきである。さらに、サイクルを発生させない $\Rightarrow_L^{(p)}$ は保存される必要はない。なぜなら、二つのL-アクション a_1, a_2 が $a_1 \Rightarrow_L^{(p)} a_2$ の関係を満たしていて、 $a_2 - a_1$ の順に実行されたとしても、これらがL-競合していない限り問題はないからである。以上のことを考慮し、等価性についての定義を与える。

定義 2.4 [2レベル競合保存 (CP) 等価]

二つの2レベルスケジュール s, s' が2レベルCP等価であるのは、次の三つの条件が成立するとき、かつそのときに限る。

- (1) s, s' は同一のトランザクション集合から構成される.
- (2) 各物理データ単位 p に対し, p に関する L 準依存関係によるサイクルが一致する.
- (3) L 競合が一致する. $\square$

定義 2.5 [2 レベル競合保存 (CP) 直列化可能性]

s をある 2 レベルスケジュールとする. このとき, s が 2 レベル CP 直列化可能であるのは, s に 2 レベル CP 等価な直列スケジュールが存在するときかつそのときに限る. $\square$

次に, ある 2 レベルスケジュールが CP 直列化可能であるための必要十分条件を示す.

定理 2.1 [CP 直列化可能性の必要十分条件]

s をある 2 レベルスケジュールであるとする. このとき, s が 2 レベル直列化可能であるための必要十分条件は, トランザクション先行関係 $\Rightarrow_T$ によるサイクルが存在せず, かつ, 任意の物理データ単位 p に対して, L 準先行関係 $\Rightarrow_L^{(p)}$ によるサイクルが存在しないことである.

(証明略)

$\square$

ある 2 レベルスケジュール s が与えられているとき, s が 2 レベル CP 直列化可能であるかどうかを判定するのは, $\Rightarrow_T$ および, $\Rightarrow_L^{(p)}$ を表現するグラフにおけるサイクルの存在性を調べることに帰着できるので, 多項式時間で行える. 等価性の定義は CP 等価性以外に考えることが可能であるが, 2 レベルトランザクションに関して自然に拡張できることを考えて CP 等価性の概念を用いた.

3. 2 レベルのデータ単位を考慮した直列化可能性

実際のデータベースシステムでは, L -アクションの系列からなるトランザクションがデータベース管理システムに入力され, その L -アクションに対する P -アクションがデータベース管理システムで生成されてデータベースへのアクセスが行なわれるものと考えられる. このような現実を十分に反映したスケジュールモデルとしては, データベースへの P -アクションの対象となるデータ単位の情報として, 単にアクセスする物理データ単位のみでなく, 実際に参照, 更新の対象となる論理データ単位の情報も加えた P -アクションのスケジュールを考え, その正当性を考えることが可能である. その利点として, 2 章で述べたようにスケジュールを単に物理データ単位へのアクセスの系列と構文的に考えた場合に比べて, アクセスデータ単位に関する意味

的競合を考慮した並行処理制御方式が実現され、より並列性の高いP-アクションの実行が可能になる場合がある。以下では、各P-アクションが、対応するL-アクションのアクセスする論理データ単位のための情報を付加的にもつものとし、その仮定のもとでのスケジュールの正当性を前章での2レベルCP直列化可能性によって保証することを考える。

あるP-アクションのスケジュールにおいて、各アクションが自分がアクセスする物理データ単位 p のほかに、実際にトランザクションが必要とする論理データ単位 x が分かっているという仮定のもとで、例えば、それが読取り（書込み）操作である場合、

$\text{read}(x; p) \quad (\text{write}(x; p))$

のように記す。このような操作命令を混合アクションとよぶ。より一般には、混合アクションはそれが必要とする論理データ単位の集合 LS と、実際にアクセスする物理データ単位の集合 PS を用いて、タイプRの場合には $\text{read}(LS; PS)$ 、また、タイプWの場合には $\text{write}(LS; PS)$ と示される。ただし、複数の物理データ単位にアクセスするアクションは複数のアクションに分割ができるので、一つのアクションが一度にアクセスする物理データ単位は一つであると仮定する。以上の混合アクションをもとにしたスケジュールを混合スケジュールと呼ぶ。直感的には、ある混合スケジュールが直列であるとは、すべての混合アクションの間に一切のインターリーブが許されていないことをいう。

混合アクションの間の先行関係を $<$ で示すことにする。また、すべてのトランザクションはある論理データ単位を対象とするタイプWのL-アクション、タイプRのL-アクションはそれぞれ高々一回であると仮定する。ある混合スケジュール s 内には、あるL-アクション（例えばタプル x を更新）を実行するために同一トランザクションから一つ以上の混合アクション（例えば $\text{read}(x; p)$ と $\text{write}(x; p)$ ）がこの目的のために発せられる。ある混合アクション a に対して、 a と同一の目的で同一のトランザクションから発せられた混合アクションの集合を a も含めて $\text{broth}(a)$ と記す。

定義 3.1 [依存関係 $\rightarrow$]

二つのアクション a_1, a_2 に対して、依存関係 $\rightarrow$ を次のように定義する

$$a_1 \rightarrow a_2 \iff (a_1, a_2 \text{ はある論理データ単位に関して競合}) \wedge a_1 < a_2.$$

定義 3.2 [混合スケジュールの妥当性]

混合スケジュール s は、次の二つが成立するときかつそのときに限り妥当であるといわれる。

- (1) $<$ 関係が全順序関係である。
- (2) s を構成する任意の混合アクション a_1, a_2 に対して、

$$a_1 \rightarrow a_2 \iff \text{for } \forall a'_1 \in \text{broth}(a_1), \forall a'_2 \in \text{broth}(a_2) (a'_2 \not\prec a'_1). \quad \square$$

任意の2レベルスケジュールは混合スケジュールに変換可能であり、その結果はユニークである。また、以下に述べる規則 TRを用いて変換した場合、妥当な混合スケジュールは必ず妥当な2レベルスケジュールに変換され、また、その変換による結果はユニークである。

変換規則 TR

ある混合スケジュール s を2レベルスケジュール s' に変換する場合、次の rule-1~rule-3が守られる。

rule-1: s' を構成する二つの L-アクション a, a' に対して、

$$a <_L a' \iff \forall b \in P\text{-act}(a), \forall b' \in P\text{-act}(a') (b <_P b').$$

rule-2: s' においてタイプ W の L-アクションがアクセスする論理データ単位は、すべてタイプ W のアクセスの対象になっているとする。

rule-3: s' において複数論理データ単位にアクセスする L-アクション a を考える。この論理データ単位の集合を LS とすると、 $\forall x \in LS$ に対して

$$\exists y \in LS \quad \text{s.t.} \quad (x \neq y) \wedge (SPT(x) \cap SPT(y) \neq \phi).$$

ただし、 $SPT(x)$ で x のアクセスに必要な十分な物理データ単位の集合を示す。□

上記の rule-1 は、 $<_L$ 関係をユニークに定めるために置かれた規則である。rule-1 を仮定しない場合、あるタイプ 2 スケジュールから変換して得られる2レベルスケジュールが複数考えられるが、それらはすべて2レベル CP 等価である。それゆえ、変換により得られる2レベルスケジュールの直列化可能性を考える場合には、それらの代表を一つ取って考えれば問題ない。rule-2, rule-3 は、L-アクションの集合をユニークに定めるために置かれた仮定である。

これまでの2レベルスケジュールについての議論をそのまま用いれば、混合スケジュールについての議論が行える。以下で扱う混合スケジュールは妥当であることを前提としている。オフラインのスケジューリングを考える際には、この混合トランザクションは2レベルトランザクションと同じ能力を有することがいえる。

定義 3.3

1. 二つの混合スケジュール s_1, s_2 に対し、これらを変換して得られる2レベルスケジュールをそれぞれ s'_1, s'_2 とすると、

$$s_1 \text{ と } s_2 \text{ は CP 等価} \iff s'_1 \text{ と } s'_2 \text{ は 2 レベル CP 等価}$$

2. s をある混合スケジュールとする。このとき、 s が CP 直列化可能であるのは、 s に CP 等価な直列スケジュールが存在するとき、かつそのときに限る。□

直列な混合スケジュール s を変換して得られる 2 レベルスケジュール s' は 2 レベル直列であり、また直列な 2 レベルスケジュール s' を変換して得られる混合スケジュールも直列である。それゆえ、ある混合スケジュールが直列化可能であるかどうかを判定するには、 s を変換して得られる 2 レベルスケジュール s' が 2 レベル直列化可能であるかどうかを判定すればよい。また直列化可能性に関して、次の定理が成立する。

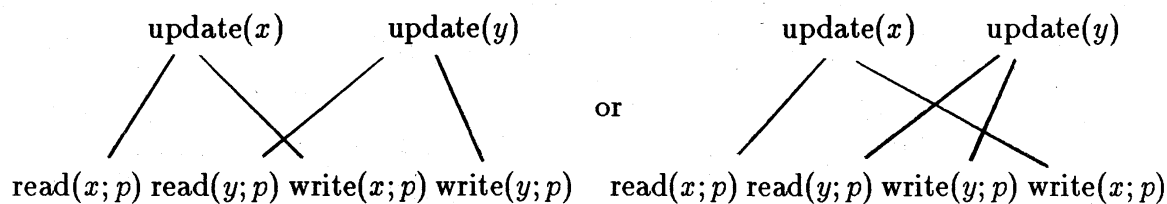
補題 3.1 [$\Rightarrow_L^{(p)}$ 関係のサイクルに関する性質]

混合スケジュールから導かれた 2 レベルスケジュールにおいて、 $\Rightarrow_L^{(p)}$ 関係によるサイクルが存在するなら、それらのうちの少なくとも一つは長さが 2 である。

(証明略)

□

上述の補題によると、変換して得られた 2 レベルスケジュールに対する直列化可能性の判定の際に、 $\Rightarrow_L^{(p)}$ 関係によるサイクルの存在性のテストを行なわなくても、より簡単な手続きで代行できることが示せる。なぜなら、 $\Rightarrow_L^{(p)}$ による長さ 2 のサイクルが発生するのは、二つのタイプ W の L-アクション (L-競合していない) が次の図に示されるようにインターリーブされたときに限るからである。



この二つの場合の発生を検出するには、アンダーラインのような、別のタイプ W の L-アクションの有する二つの P-アクションに挟まれたタイプ W の P-アクションが存在するかどうかを確かめればよい。つまり、混合スケジュールを 2 レベルスケジュールに変換してその 2 レベルスケジュールの直列化可能性を判定するという手続きを踏まなくても、上図でアンダーラインを引かれたようなアクションが存在するかどうかを検出するという 1 レベルのみの議論で済ませることができる。以下では上で示されたようなタイプ W のアクションを無効更新と呼ぶ。

定理 3.1 [混合スケジュールの CP 直列化可能性の必要十分条件]

s を混合スケジュールであるとする。このとき、 s が直列化可能であるのは次の二つの条件が成立するとき、かつそのときに限る。

- (1) s 内の各アクションから物理データ単位に関する情報を取り除き、1 レベルスケジュールに変換したとき、このスケジュールが (従来の意味で) CP 直列化可能である。

(2) 各物理データ単位に関して無効更新が存在しない。

(証明略)

□

4. おわりに

本稿では、従来の並行処理制御方式に関する研究において各操作命令の対象とするデータ単位が実際システムにおいてどのデータ単位を示しているのかが曖昧であったことを問題にし、物理データ単位、論理データ単位という二つのデータ単位を考慮したトランザクションモデルを考え、そのトランザクションモデルのもとでのスケジュールに対する直列化可能性について議論した。特に、2レベルのデータ単位を考慮することにより、正当なスケジュールを実現する上での並行実行可能な操作命令の集合がより明確となり、物理データ単位だけを考慮した1レベルスケジューラに対する並行処理よりも高い並列度を保った操作命令の実行可能性が示唆された。本稿で用いられたトランザクションモデルは2レベルのデータ単位を考慮したものであったが、物理データ、論理データのどちらかが（あるいは両方が）2レベル以上であったり、物理、論理以外のレベルを考えるとすれば、議論の対象となるトランザクションモデルは3レベル以上である。この場合も、本稿で述べられた理論を（データモデルに合わせて）拡張することは可能である。

最後に、本稿では、与えられた2レベルのデータ単位を考慮したスケジュールが依存関係保存直列化可能であるかを判定する効率の良い判定法を示したが、さらに、この判定法を用いたオンラインスケジューラの構築法に関する研究、さらには、他の直列化可能性の定義を用いた2レベルスケジュールの正当性に関する研究は将来の課題である。

参考文献

1. Bernstein, P.A., Shipman, D.W., and Wong, S.W. : "Formal Aspects of Serializability in Database Concurrency Control", IEEE Trans. Softw. Eng. Vol. SE-5, No. 3, pp.203-216, 1979.
2. Bernstein, P.A., Goodman, N., and Lai, M.-Y. : "Analyzing Concurrency Control Algorithms When User and System Operations Differ", IEEE Trans. Softw. Eng., Vol. SE-9, No. 3, pp.233-239, 1983.
3. Bernstein, P.A., Hadzilacos, V., and Goodman, N. : Concurrency Control and Recovery in Database Systems, Addison-Wesley, Reading, Massachusetts, 1987.

4. Brzozowski, J.A. : "On Models of Transactions", Technical Report No. 84001, Department of Applied Mathematics and Physics, Kyoto University, Japan, 1984.
5. Brzozowski, J.A. and Muro, S. : "On Serializability", Int. J. Comput. Inf. Sci., Vol 14, No. 6, pp.384-403, 1985.
6. Ibaraki, T., Kameda, T., and Minoura, T. : "Serializability with Constraints", ACM Trans. Database Syst., Vol. 12, No. 3, pp.429-452, 1987.
7. Minoura, T. : "Multi-Level Concurrency Control of a Database System", Proc. 4th IEEE Symp. on Reliability in Distributed Software and Database Systems, pp.156-168, 1984.
8. Papadimitriou, C.H. : "The Serializability of Concurrent Database Updates", J.ACM, Vol. 26, No. 4, pp.631-653, 1979.
9. Papadimitriou, C.H. : The Theory of Database Concurrency Control, Computer Science Press, Rockville, Maryland, 1986.
10. Vidyasanker, K. : "Generalized Theory of Serializability", Acta Inf., Vol 24, No. 1, pp.105-119, 1987.
11. Welkum, G. : "A Theoretical Foundation of Multi-Level Concurrency Control", Proc. 5th ACM SIGACT-SIGMOD Symp. on Principles on Database Systems, pp.31-42, 1986.