数理解析研究所講究録
第494巻 1983年 102-113

Serializable Classesの構造について

京都大学工学部　茨木 俊秀
Simon Fraser大　亀田 恒彦
Oregon State大　箕浦 敏美

1. まえがき

データベース管理システム（DBMS）における同時処理制御（concurrency control）は，基本的な問題の一つとして，さまざまな立場から研究されている．同時処理の妥当性の判定にはserializabilityの概念が主として用いられるが，その一般的な判定問題はNP完全であって[4]，効率良いスケジューラーが存在するとは考えられない．そこで，serializabilityの定義にいくつかの付加条件を加えることで，判定が多項式時間で可能であるようなクラスの構成が試みられている．そのようなクラスの例にPapadimitriouによるDSR[4]（Bernsteinら[1]のCPSRと同一）がある．

DSRは，与えられたスケジュール内のoperationsの順序のある部分を保存するという付加条件によって定義される

が，この考え方を延長すると，保存すべき順序部分をどう定義するかで，種々のクラスが定義できる．ここでは，同一のdata itemを「読む」および「書く」というoperationsに関して定義されるいくつかの順序制約を導入し，得られたクラス間の包含関係を明らかにした．また，各クラスの所属問題が多項式時間のアルゴリズムをもつか，あるいはNP完全になるかどうかも示した．

以上のクラスの多くは新しく定義されたものであるが，とくに，DSRを真に含み，しかも多項式時間アルゴリズムをもつクラスWR+RWの存在は注目に値しよう．また，これらの議論は，transaction input/output graph (TIO graph)上のspan nonoverlapping topological sort (SNOTS)という概念を用いて統一的になされる．従来，serializabilityの議論が，クラス毎に異なる道具を用いてなされていたため，全体の見通しが悪いという欠点を有していたことに対し，同一の道具を一貫して用いることで，構造上の相違を際立たせることに成功したと思われる．

2. 諸定義

DBMSに transactions $T_1, T_2, \ldots, T_n$ が入力される．各transactionはいくつかのread operationsとwrite

operationsから成る。ただし，data item X へのread operation を $R_j[X]$，write operation を $W_j[X]$ と記す．j は transaction T_j を示す添字である．transactionsに対しては，一般モデルと2ステップモデルがよく用いられる．一般モデルでは，T_j を半順序の定義されたいくつかの $R_j[X]$ と $W_j[X]$ の集合と捉え（items X は種々存在する），2ステップモデルでは，$R_j[X, Y, \ldots]$ に $W_j[Z, \ldots]$ が後続すると考える．$R_j[X, Y, \ldots]$ は items $X, Y, \ldots$ へのread operation であり，$W_j[Z, \ldots]$ も同様に解釈する．本論文の結果は，どちらのモデルに対しても成立する．

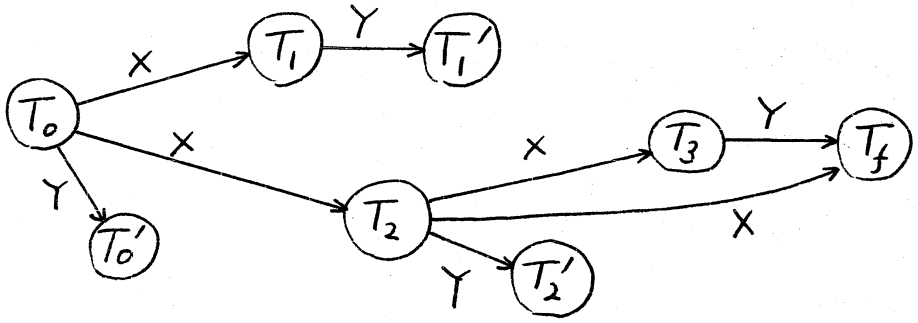
$T_1, T_2, \ldots, T_n$ に含まれるoperationsは，あらかじめ各 T_j 内で指定された順序を除けば，勝手な順序でDBMSに到着する．到着順にoperationsを並べたものをscheduleという．たとえば，

$$S = W_0[X, Y]\, R_1[X]\, R_2[X]\, W_2[X, Y]\, R_3[X]\, W_1[Y]\, W_3[Y]\, R_f[X, Y] \quad (1)$$

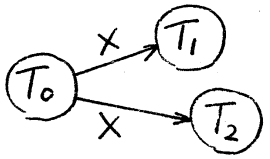
である．ここに，最初の W_0 と最後の R_f は形式的に付されたもので，それぞれ最初にすべてのitemsを書き，最後にすべてのitemsを読むoperationsである．

通常のDBMSでは，各itemのコピーを常に1個だけ保持する（backup用は別にして）ので，$R_j[X]$ という

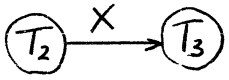
operation は，sにおいてそのoperation以前に書かれた最新のitem Xを読むことになる．このようなモデルをsingle versionモデルという．スケジュールsに対し，グラフTIO(s)をつぎのように定義する。TIO(s)は各transaction T_jを節点とし，$W_h[X]$で書かれたXを$R_i[X]$が読むとき，ラベルXをもつ枝(T_h, T_i)を作る．$W_k[X]$で書かれたXが，どのread operationによっても読まれないとき，$W_k[X]$をuselessであるといい，T_kにダミー節点T_k'を付し，ラベルXをもつ枝(T_k, T_k')を書く．式(1)のsに対するTIO(s)は



である．ある節点から発し，同じラベルをもつ枝の集合をspanという．TIO(s)の全節点を左から右に，枝方向に矛盾しないように一列に並べるとき，同じラベルをもつ2個以上のspansが重ならないならば，SNOTSであるという．上の例では

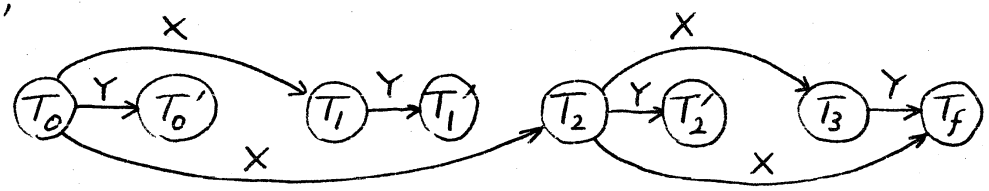


や



がspans

であり，



はSNOTSである．

スケジュールSとS'において，TIO(S)とTIO(S')が一致するならば，SとS'は互いに等価であるという．スケジュールS'において，operationsがtransaction毎に整列しているとき，つまり(添字は1, 2, …, nの順でなくてよいが)

$$s' = W_0 R_1 W_1 R_2 W_2 \cdots R_n W_n R_f$$

という形をとるとき，serialスケジュールと呼ぶ．スケジュールSがあるserialスケジュールS'に等価ならば，Sはserializableであるという．serializableなスケジュールはデータ内の首尾一貫性を保持することが知られており，concurrentに処理することの妥当性を示すものである．以下，serializableなスケジュールのクラスをSRと記す．つぎの定理は，以下の議論において基本的である．

<u>定理1</u>．$S \in SR$の必要十分条件は，TIO(S)がacyclicかつSNOTSをもつことである．□

式(1)のSは，上のSNOTSから導びかれるように，serialスケジュール

$$s' = W_0[X,Y] R_1[X] W_1[Y] R_2[X] W_2[X,Y] R_3[X] W_3[Y] R_f[X,Y]$$

に等価であるから，$s \in SR$ を得る．

3．付加条件

スケジュール S において W_i が R_j に先行するとき

$$W_i \ll_S R_j$$

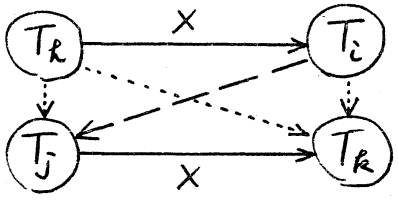
と書く．他の組合せについても同様である．serializability の定義において，等価となる serial スケジュール S' に関しつぎの付加条件を考える．

WW-条件：　ある item X に対し，$W_i[X] \ll_S W_j[X]$ ならば，S' において T_i は T_j に先行する．

同様に，WR-条件，RW-条件，RR-条件も定義しよう．WW-条件の下で serializable なスケジュールのクラスを WW と記す．WR，RW，RR 等のクラスも同様である．つぎに，クラス WR+RW は，WR-条件と RW-条件の両方をみたす S' に対し serializable であるようなスケジュールのクラスを記す．他の組合せについても同様に定義する．

付加条件に対応して，グラフ $TIO(S)$ に条件枝を追加する．たとえば，WW-条件の場合，$W_i[X] \ll_S W_j[X]$ ならば，枝 (T_i, T_j) を加えるわけである．これらの枝と SNOTS 条件から，さらに枝が追加される．次図の状況において（T_i と T_k はダミー節点であってもよい），点線の枝の一つでも加われ

ば，破線の枝が追加され，T_i を T_j に先行させるのである（そうでなければ SNOTS が存在しない）．

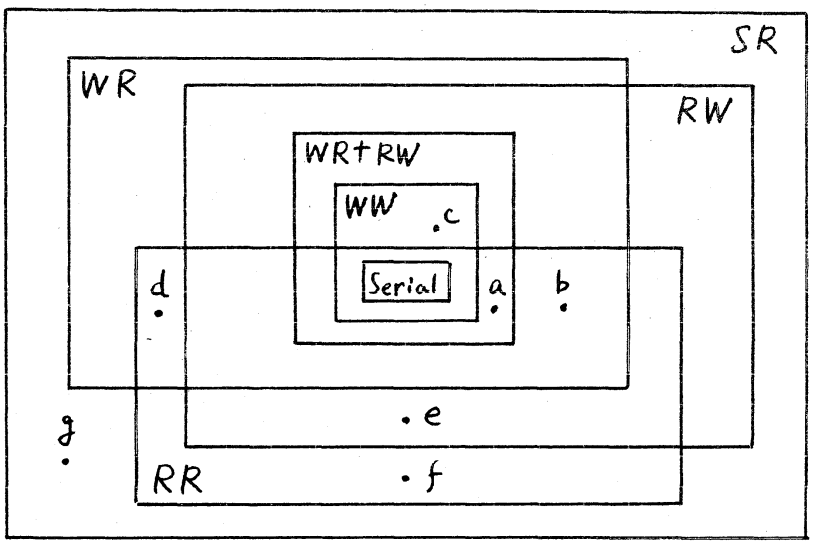


これは Sethi [5] のいう exclusion rule に本質的に同一である．得られたグラフを $TIO_{WW}(S)$ と呼ぶ．他のクラスについても同様に定める．

定理2．各クラス $*$ = WW, WR, RW, … などに対し，$S \in *$ の必要十分条件は，$TIO_*(S)$ が acyclic かつ SNOTS をもつことである．□

5．クラス間の包含関係

上記の WW, WR+RW などのクラスの間の包含関係は下図



の如くである．とくに，

$$WW = WW + WR + RW = WW \cap WR \cap RW$$

$$WW \subsetneq WR + RW \subsetneq WR \cap RW$$

などが興味深い．前述のDSRは本来 $WW+WR+RW$ として定義されているが，この結果からWWに等しいことが判明した．図中のスケジュール a，b 等は以下のとうりである．

$$a = W_0[X,Y]\, R_1[X]\, R_2[X]\, W_2[X,Y]\, R_3[X]\, W_1[Y]\, W_3[Y]\, R_f[X,Y]$$

$$b = W_0[X]\, W_1[X]\, R_2[X]\, W_3[X]\, W_2[X]\, R_4[X]\, W_5[X]\, R_f[X]$$

$$c = W_0[X]\, R_2[X]\, R_1[X]\, W_2[X]\, R_f[X]$$

$$d = W_0[X]\, R_3[X]\, W_1[X]\, R_2[X]\, W_3[X]\, W_2[X]\, R_f[X]$$

$$e = W_0[X,Y]\, R_2[Y]\, R_1[X]\, W_2[X]\, W_1[X]\, R_3[X]\, W_4[X]\, R_f[X,Y]$$

$$f = T_0\, d^*\, T_1\, e^*\, T_f$$

$$g = T_0\, d^*\, T_1\, e^*\, T_2\, c^*\, T_f$$

ただし，T_1 と T_2 はすべての items を読んで書く transactions，d^* は d から W_0 と R_f を除いたもの．e^*，c^* も同様である．

6．特別な場合の包含関係

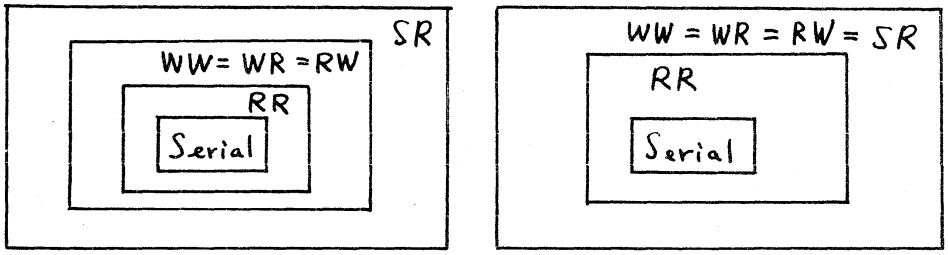
transactions の同時処理に際して，以下の制約がしばし

ば付される.

(1) useless operations W_j は存在しない。

(2) T_j が item X を書くならば, X を必ず読んでいなければならない.

これらの制約の下では, クラス間の包含関係は下図のように単純化される.



制約 (1)　　制約 (2)

7. 多項式時間アルゴリズムをもつクラス

各クラスの所属問題に関連して, つぎの結果が得られる.

定理3. $s \in WW$, $s \in WR+RW$ はいずれも多項式時間で判定可能である. □

WW (=DSR) については [4] などに証明がある. WR+RW は WW を真に含み, しかも所属問題が多項式時間で判定可能であるのは興味深い.

TIO(S) における SNOTS に基づく定理2を用いると, これらの判定アルゴリズムを統一的に記述できる. すなわち, 4節の $TIO_{WW}(S)$ および $TIO_{WR+RW}(S)$ を多項式時間で構成す

ると，これらがacyclicであるとき，かつその時に限りS∈WWおよびS∈WR+RWとなる．acyclicityはよく知られているように，多項式時間で判定可能である．サイクルの有無に帰される理由は，WWの場合，WW-条件の枝とexclusion ruleの枝を考慮すると，同一のitem Xを読み書きするすべてのtransactions間にtotal orderが定義されるので，SNOTSの判定が自明となるからである．WR+RWについては，T_iとT_jの両者ともuseless Xを書く場合を除いて，やはり順序が定まる．useless Xを書くT_iとT_jはどちらを先にしてもSNOTSの存在に影響を与えないので，SNOTSの判定はやはり自明になされる。

8. NP完全であるクラス

WWとWR+RW以外については，次の結果がある．

定理4. クラスSR, WR, RW, RR, WR+RR, RW+RRの所属問題はすべてNP完全である．□

SRに関しては[4]に証明があるが，他は新しい結果である．いずれも充足可能性問題（詳しくはNOT-ALL-EQUAL SAT [2]）を，対応するクラスの$TIO_*(S)$に帰着することで証明される．

9. Multi-Version モデルへの拡張

これまで，各 item のコピーは常に1個だけ存在するという前提の下に話を進めてきた．複数個（一般にはいくつでも）のコピーの存在を許せば，operation $R_j[X]$ に際して，それまでに書かれているコピーの中から適当なものを選んで読むことができる．この結果，これまでの定義では serializable ではないスケジュールも，multi-version の下では serializable となる可能性があり，同時処理の機能を高めることができる．本論文で導入したグラフ TIO(s) と SNOTS という概念は，multi-version へも自然に拡張され，同種の議論を展開できる．得られた結果は別の機会に報告したい．

なお，本論文の定理の証明など，詳しい内容は，文献[3]を参照されたい．

文献

[1] P. A. Bernstein, D. W. Shipman and W. S. Wong, Formal aspects of serializability in database concurrency control, IEEE Trans. Software Eng. SE-5, 3, pp. 203-215, May 1979.

[2] M. R. Garey and D. S. Johnson, Computers and

Intractability: A Guide to the Theory of NP-completeness, Freeman, 1979.

[3] T. Ibaraki, T. Kameda and T. Minoura, SNOTS and its applications: Serializability theory made simple, Simon Fraser University, Department of Computing Science, TR 82-12, December 1982.

[4] C. H. Papadimitriou, The serializability of concurrent database updates, JACM, 26, 4, pp. 631-653, Oct. 1979.

[5] R. Sethi, A model of concurrent database transactions, Proc. 22nd IEEE FOCS, pp. 175-184, Oct. 1981.