並行処理制御におけるホーン節で表された一貫性情報の利用

5AA－6

史 一華　徐 海燕　古川哲也
福岡工業短期大学　福岡工業大学　九州大学

1 はじめに

従来の並行処理制御は，データベースの一貫性は未知のものとされ，各作業を逐次的に実行する直列実行の結果がデータベースの一貫性を保持するという前提で行われてきた．しかし，直列可能性基準に従うデータベースの制御機能は，協同作業制御を支援するのに適切でない．この問題は広く認識されており，様々な角度から研究されてきている．著者らは，協同作業における作業手順によって定義される一貫性制約を利用することにより，独立化可能という直列可能より範囲を拡大した正当なスケジュールのクラスを提案している[2]．本稿では，ホーン節で表された一貫性情報に対して，実用的な並行処理制御方式の下で，クラスの性質や大きさについて検討し，それをデータベースの並行性制御機能として実装する時の具体的な性質を明らかにする．

2 ホーン独立化可能性

協同作業のプロジェクトは，通常複数個の部分作業に分けて行われる．これらの部分作業は並行に行えるものもあれば，一定の順序に従って行わなければならないものもある．このため，各々の作業者は「一定のルール，規定，順序や約束」という作業手順に従って協同作業を行う．一貫性制約Cは，そのような作業手順を節で表したものである．C中のすべての制約がデータベースにおいて真である時，その時点のデータベースはCを満たす（データベースは一貫している）という．

個々の利用者のデータベースに対する操作は，処理単位Tという全順序関係$<$を持つ検索操作$(R(X))$，変更操作$(W(X))$の集合で表される．複数の処理単位（処理単位集合$\boldsymbol{T}$）の並行実行をスケジュールHという．一貫性情報を利用することによって，並行処理制御の正当性基準を直列可能性から独立化可能性に拡張できる．

[定義 1] [2] 一貫性制約Cを持つデータベースにおける独立化可能スケジュールHとは，各$T_i \in \boldsymbol{T}$が次の３つの性質を満たす等価なスケジュールが存在するものである．

(1) 検索一貫性：T_iによって検索されるのは，ある一貫したデータベースの部分集合である．

(2) 結果一貫性：検索されたデータベースに対してT_iを単独で実行した結果のデータベースはCを満たす．

(3) 論理性：T_iによって操作されたデータ項目は，T_iが終了するまで他の処理単位によって変更されない．　□

ホーン独立化可能クラスは，一貫性制約Cをホーン節に限定した独立化可能クラスの部分クラスである．

3 ホーン独立化可能クラスの性質

本稿では，ホーン独立化可能性に基づく並行処理制御方式の具体的な性質を検討する．

ここでは，Cをホーン節に限定することによって，独立化可能性の３つの条件に与える影響を簡単に述べる．

(1)（検索一貫性）T_iが検索一貫性を満たすための必要十分条件は，$(T(\Delta^+)\uparrow\omega)\cap\Delta^- = \phi$である[3]。ただし，$T(S)$は，$C$を用いて，データ集合$S$からボトムアップ推論で得られるデータの集合で，$T(S)\uparrow\omega$は$S$から繰り返し推論できる全てのデータの集合で，$\Delta^+$と$\Delta^-$は，$T_i$で検索されたデータをその値の真偽で分けたものである．

ホーン節においては，この計算は多項式時間でできる．更に，通常$\Delta(=\Delta^+\cup\Delta^-)$のデータ項目と関連しているデータのサイズに対して定数倍になる．

(2)（結果一貫性）次のような検証アルゴリズムで判定できる．その計算量は，$|\Gamma| * |C|$の定数倍となる．ただし，θは代入例，Γは変更されたデータ集合，ΠはΓの検証に必要なデータの集合である．

アルゴリズム 1[3]

foreach $x \in \Gamma^- \cup \Gamma^+$ **do**
 foreach $(A \leftarrow B_1,\cdots,B_n) \in C$ **do**
  if $\exists\theta$: $(A\theta = x)$ **or** $(\exists B_i\theta = x)$ **then**
   if $(\forall B_i\theta \in \Pi^+ \cup \Gamma^+)$ **and** $(A\theta \in \Gamma^- \cup \Pi^-)$ **then**
    中止：結果一貫性検証は失敗
 end
end

(3)（論理性）一貫性制約Cと直接関係がないため，ここでの制限による影響はない．論文[2]では，独立化可能性のための並行処理制御方法が与えられ，論理性の判定が多項式時間でできることも示されている．

したがって，Cをホーン節に限定することによって独立化可能性の３つの条件の判定がそれぞれ実用的な時間でできるようになっている．

4 ホーン独立化可能クラスの実現

本節では，実用的な並行処理制御方式を用いて，独立化可能性の３つの条件の相互関連を明らかにする．まず，論理性を保証するために，次の並行処理制御方式を用いる．

Utilizing Consistency Information Expressed by Horn Clauses for Concurrency Control
Yihua Shi[1], Haiyan Xu[2], and Tetsuya Furukawa[3]
[1]Fukuoka Junior College of Technology
[2]Fukuoka Institute of Technology
[3]Kyushu University

[定義 2] [2] 各検索操作 $R_i(X) \in H$ または変更操作 $W_i(X) \in H$ を行う前に X 中の各要素に対して共有施錠または専有施錠を行い，かつ各処理単位 $T_i \in T$ は施錠部分と解錠部分とに分かれる並行処理制御方式を拡張2相施錠(2PLE)方式という．ただし，すでに施錠されているデータ項目を他の処理単位が再び専有施錠することはない． □

2相施錠方式(2PL)は，提案されている数多くの並行処理制御方式[1]の中で最も多く利用されているものである．2PLE方式は，2PL方式から専有施錠とその後に施錠される共有施錠間の競合関係を無くしたものである．

結果一貫性の問題，つまり作業者が作業手順を遵守しているかどうかの問題は，並行実行に依存しないので，各処理単位はアルゴリズム1の判定を通過できるものと仮定する．しかし，検索一貫性の問題は，完全に並行実行に依存する．協同作業は複雑でかつ知的なものが多いため，検索一貫性問題による設計作業（処理単位）の後退復帰は従来のデータベースよりもっと深刻な影響を及ぼす．したがって，ホーン独立化可能性クラスを実用化するために，検索一貫性の判定は処理単位終了時ではなく，各々の検索操作ごとに行うことにより，早期に問題を発見し，対策をたてることができるようにする．

処理単位 T_i の検索操作 $R_i(x)$ が実行されるまでに検索されたデータの集合を Δ_i とする．$R_i(x)$ が実行される時点で検索一貫性で問題が生じるなら，2PLE方式の制御の下で，T_i の検索操作と他の処理単位の変更操作間の施錠関係は次のようになる．

(a) $\{x\} \cup \Delta_i$ の対象が全て他の処理単位によって専用施錠されている．

(b) $\{x\} \cup \Delta_i$ の対象が一つも他の処理単位によって専用施錠されていない．

(c) x は専用施錠されており，Δ_i の一部は他の処理単位によって専用施錠されていない．

(d) x は専用施錠されておらず，Δ_i の一部は他の処理単位によって専用施錠されている．

(e) x と Δ_i の一部が自分によって専用施錠されている．

これらのケースの簡単な例を表1に示している．

まず，専用施錠してる処理単位が結果一貫性を満たすならば，(a)は起こり得ない．(b)についてもそれらのデータを変更した処理単位（すでに終了している）が結果一貫性を満たしたので，有り得ない．

(c)は，x と Δ_i の一部を専用施錠している処理単位はすでに実行される検索操作の対象に対して新たな変更操作を行わないと，結果一貫性を満たせないことを意味する．しかし，2PLE方式によって専用施錠が禁止されているので，スケジュールはすくみに陥る．

(d)で検索一貫性の問題が検出されることは，Δ_i の一部を専用施錠している処理単位が x や Δ_i の一部（専用施錠していないもの）に対して新たな変更操作を行わないと，結果一貫性を満たせないことを意味する．この場合，現在の $R_i(x)$ をその変更操作後に延期させる方法が考えられる．ただし，(c)のようにすくみになってしまう可能性もある．

(e)については，自分自身の変更結果をもう一度検索することによって簡単に対処できる．

表1：検索操作と他の処理単位の変更操作間の施錠関係

	T_1	T_2	状態	T_1	T_2	状態
t_1		$W_2(y)$	偽	$R_1(y)$		偽
t_2	$R_1(y)$		偽		$W_2(y)$	真
t_3		$W_2(x)$	偽	$R_1(x)$		偽
t_4	$R_1(x)$		偽		$W_2(x)$	真
	H_1	(a)		H_2	(b)	
t_1	$R_1(y)$		真		$W_2(y)$	真
t_2		$W_2(x)$	偽	$R_1(y)$		真
t_3	$R_1(x)$		偽	$R_1(x)$		偽
t_4		$W_2(y)$	偽		$W_2(x)$	真
	H_3	(c)		H_4	(d)	
t_1	$R_1(y)$		偽			
t_2	$W_1(y)$		真	条件：		
t_3		$R_2(y)$	真	2PLE 方式		
t_4		$W_2(x)$	真	$C = \{y \leftarrow x\}$		
t_5	$R_1(x)$		真	結果一貫性 通過		
	H_5	(e)				

したがって，(c)と(d)の対策のみ考慮すればよい．(c)の場合は，$R_i(x)$ をとりやめるか，すでに実行している T_i の検索操作を削除するかというような処理単位の動的再構成の必要がある．さらに，(c)とならないように事前に対策を講じる必要もある．例えば，検索されたデータから推論できたデータは，もし，後で検索する必要があるなら，その時点で検索しておく．したがって，他の処理単位が専用施錠を掛けられないようになる．

5 おわりに

本稿では，ホーン独立化可能性に基づくシステムを実現するための一つの案を論理性，結果一貫性，検索一貫性の全ての面で総合的に検討し，その対処方法を与えた．特に検索一貫性で生じうる問題を明らかにした．

参考文献

[1] Bernstein, P. A., Hadzilacos, V., and Goodman, N.: *Concurrency Control and Recovery in Database Systems*, Addison-Wesley Publishing Company (1987).

[2] 徐 海燕，古川哲也，史 一華：並行処理制御方式による独立化可能クラスと直列可能クラスの比較，情報処理論文誌，Vol. 37, No. 8, pp. 1600-1609 (1996).

[3] 史 一華，徐 海燕，古川哲也：ホーン節で表された一貫性情報を持つ高水準データベースにおける並行処