データベース・システム 68－2
（1988. 11. 17）

動的な版の選択を行うk版先読みスケジューラ

木庭淳　加藤直樹

神戸商科大学管理科学科

後退復帰が不要でデッドロックの起きない先読みスケジューラのうち，単版方式よりも並行性能を増しかつ多版方式よりも現実性を考慮したk版方式のスケジューラ$CS(\overline{WRW}(k))$（元の文献では$\overline{CS}(WRW(k))$と表記）が既に提案されている．本稿では筆者らの1版スケジューラ$CS_1(WRW_1^{\pm})$における，必要性の高い版を優先的に残す方法をk版方式に拡張した，$CS_k(WRW^-)$を提案した．$CS_k(WRW^-)$は$CS(\overline{WRW}(k))$より並行処理性能が高い．さらにこのスケジューラでの実行が取消し異常を生じないこと，および多項式時間で可能であることを示し，実用上問題のない点も明らかにした．

A k-version Cautious Transaction Scheduler with Dynamic Version Selection

Jun Kiniwa and Naoki Katoh

Department of Management Science, Kobe University of Commerce
4-3-3, Seiryodai, Tarumi-ku, Kobe-shi, 655 Japan

In this paper, we present a k-version cautious scheduler with version selection control, $CS_k(WRW^-)$, which is an extension of recently proposed k-version schedulers $CS(\overline{WRW}(k))$ and a 1-version cautious scheduler $CS_1(WRW_1^{\pm})$. The set of schedules that our scheduler $CS_k(WRW^-)$ accepts without delay properly includes the one that $CS(\overline{WRW}(k))$ does without delay. It is also shown that our $CS_k(WRW^-)$ is free from cancellation anomaly and runs in polynomial time.

1. まえがき

データベースシステムで複数のトランザクションが共有データにアクセスする環境下で，首尾一貫性を損なわないように，かつ効率をいかに高めるかというのが並行処理の問題である．この問題に対して時刻印方式，二相ロック方式があるが，後退復帰やデッドロックのため，システムが新しいトランザクションの実行を一時停止してしまう可能性がある．そこで，各トランザクションがあらかじめ将来アクセスするデータ集合をスケジューラに知らせるという条件のもとで，上記の欠点をなくした，先読みスケジューラが近年提案された[2],[3],[4],[5],[6],[7]．

最初，各種の単版先読みスケジューラが提案され[2],[4]，その後，並行性を高めるため多版先読みスケジューラが提案された[3]が，多版方式は理論的には版数を無限に保持できるという非現実的な仮定に基づいている．そこで版数に制限をつけたk版先読みスケジューラが提案された[6],[7]．

以上の研究では，いずれも全ての版を残すかもしくはどの版を残すかということは，版が生成された時刻に従い現在からさかのぼって一定数の新しい版を残すことに決っているが，もし版数制限のもとで必要度の高いと思われる版を優先的に残すことにすれば，同じ版数で版の選択を最新の版から優先的に残す場合よりも，並行性が向上する可能性がある．筆者らは先にこの方針に基づいた1版先読みスケジューラを提案した[5]が，本稿ではこれを拡張して，動的な版選択を行うk版先読みスケジューラを提案する．すなわち取消し異常のないスケジューラのうち最も並行性が優れているものとして先に提案されている，k版スケジューラ[7]より本稿のスケジューラの方が並行処理性能が優れていることを示し，また取消し異常のない点および多項式時間でスケジュール可能であることを示した．

2. 諸定義

2.1 データベースシステム

データベースシステムはトランザクションの集合$T=\{T_0, T_1, \cdots, T_n, T_f\}$とデータ項目の集合$D=\{x, y, z, \cdots\}$からなる．トランザクション$T_i \in T$は$d \in D$（または$B \subseteq D$）に対するいくつかの読込みステップ$R_i[d]$（または$R_i[B]$）と書込みステップ$W_i[d]$（または$W_i[B]$）からなる．$T_i$のステップ集合をSTEP($T_i$)と表す．またステップqのトランザクションをtr(q)と表し，$W_i[d] \in$ STEP(T_i)なるトランザクションT_iをd書込みトランザクションと呼ぶ．

トランザクションモデルは多ステップモデルを考える．すなわち各トランザクションは最小単位の$R_i[d]$, $W_i[d']$（$d, d' \in D$）に至るまで何ステップに分割されていてもよく，ある$d \in D$に対し読込みも書込みも行う場合，読込みステップの後で書込みステップを実行する．また1つのデータ項目に対し，2度以上の読込みまたは書込みステップをもたない．

各ユーザの発するトランザクションはスケジューラに入力されるが，その到着順に左から右へステップを並べた系列を考える．系列中のステップに関する全順序を$<$で表す．系列の前後に$W_0[D]$のみからなる初期トランザクションT_0と，$R_f[D]$のみからなる最終トランザクションT_fを付加し，$W_i[d]<R_j[d]$に対して読込みステップ集合から書込みステップ集合への写像である割り当て(interpretation) Iを考える．すなわち$I(R_j[d])=W_i[d]$と表すとき，$W_i[d]$による値を$R_j[d]$が読込むものとする．Iを考慮した系列hをスケジュールといい，$\langle h, I\rangle$の組で表す．すべての読込みステップに直前の書込みステップによる値を割り当てる場合を特にI_0と表す．スケジュール$\langle h, I\rangle$が直列(serial)であるとは，$I=I_0$かつ任意のトランザクションの実行中に他のトランザクションのステップが割り込まないことをいう．同じステップからなる2つのスケジュール$\langle h_A, I_A\rangle$, $\langle h_B, I_B\rangle$が$I_A=I_B$を満たし，$\langle h_B, I_B\rangle$が直列スケジュールならば，$\langle h_A, I_A\rangle$を直列化可能(serializable)であるという．

ある$d \in D$に着目したとき，スケジュール内での複数の$W_i[d]$（$T_i \in T$）による世代的な値を版(version)という．各データ項目に対して高々k個の版の存在を許す方式をk版方式，$k=\infty$すなわち過去に書込んだ版すべての存在を許すものを多版方式，$k=1$で最新版のみの存在を許すものを単版方式という．

2.2 TIOグラフとDITS

スケジュールの直列化可能性を判定するのに，文献(1)によるDITS(disjoint-interval topological sort)という概念が有効である．

〔定義2.1〕[1] スケジュール$\langle h, I\rangle$に対するTIOグラフ(transaction input/output graph) TIO($\langle h, I\rangle$)$=(V \cup V', A \cup A')$とは，次の節点集合$V \cup V'$と有向枝集合$A \cup A'$（V'はダミー節点集合，A'はダミー有向枝集合）からなる．ここでラベルd付き有向枝(T_i, T_j)を(T_i, T_j):dと表すことにする．

V: $\{T_i \mid T_i \in T\}$

V': $\{T_i' \mid I(R_j[d])=W_i[d]$なる$T_j \in V$が存在しない$\}$

A: $\{(T_i, T_j)$:d $\mid$ ある$d \in D$に対し，$I(R_j[d])=W_i[d]\}$

A': $\{(T_i, T_i')$:d $\mid I(R_j[d])=W_i[d]$なる$T_j \in V$が存在しない$\}$

以下ではT_iからT_jへの有向路を$[T_i, T_j]$と表す．またAの有向枝を割り当て読込み枝(reads-from arc)という．□

〔定義2.2〕[1] TIO($\langle h, I\rangle$)の節点に関するある全順序$\ll$が次の条件を満たすとき，DITSであるという．

(1) $T_i \ll T_j$ならば，有向路$[T_j, T_i]$は存在しない．

(2) $g \neq j$を満たすような(T_g, T_i):dおよび(T_j, T_k):dが存在し，$T_g \ll T_k$ならば$T_i \ll T_j$が成り立つ．

(3) $T_i \neq T_0, T_f$なる任意のT_iに対して$T_0 \ll T_i \ll T_f$である．□

〔定義2.3〕[1] 定義2.2(2)を満たすT_i, T_jに対し，(T_i, T_j)を排除枝(exclusion arc)と呼び，排除枝を可能な限り付加したグラフを排除閉包(exclusion closure)という．□

スケジュール$\langle h, I\rangle$が直列化可能であるための必要十分条件は，TIO($\langle h, I\rangle$)がDITSをもつことである[1]．なお多版方式での直列化可能スケジュール全体の集合のクラスをMSRと表す．

2.3 制約付き直列化可能性

直列化可能性の所属判定を多項式オーダーで抑えるために，下記の制限を設けることが考えられてきた．

〔定義2.4〕[1] 系列hでのステップの全順序$<$により，トランザクションの順序制約$\ll$を付けたDITSを考える．例えば$W_i[d]<R_j[d]$ならば$T_i \ll T_j$とすることをwr制約という．同様にww制約，rw制約も定義できる．これらを組み合わせて，DITSにおいてwr制約とrw制約が満たされる場合を特にwrw制約という．□

$TIO_c^*(\langle h, I\rangle)$は，TIO($\langle h, I\rangle$)にc制約枝（c＝ww, wrwなど）を付けた上での排除閉包を表す（以下，他のグラフについても同様の表記を行う）．

3．k版先読みスケジューラ

k版先読みスケジューラは，各データ項目につきk個以内の版を保持することができ，かつ各トランザクションT_iがスケジューラ入力時にＳＴＥＰ(T_i)を予告するという条件下で，デッドロックが無く，後退復帰の不要なスケジュールを構成する．$\langle P, I\rangle$を既に出力された部分スケジュール，ｑを現在のリクエスト，ＰＥＮＤを宣言されているがまだ未実行のステップの集合，ＤＥＬを実行が遅延されているステップの集合とし，各トランザクションは前のステップの完了後に次のステップの実行要求を出す．このとき先読みスケジューラは，$\langle PqQR_f[D], II_qI_Q\rangle$がＭＳＲの部分クラスに属するような，ＰＥＮＤ中の全ステップからなる系列Ｑおよびｑ, Ｑの割り当てI_q, I_Qの存在を調べ（k版完成テスト），もし存在すればｑを許可し，ＤＥＬ中で実行可能なステップを調べる．存在しなければｑをＤＥＬに加え，次の到着ステップについて実行可能かどうかを調べる．

k版先読みスケジューラには，データベースに残す版を一律に新しい版からk個残す方式(6),(7)と，本稿で提案する，必要と思われる版のうちで新しい版からk個残す方式がある．前者を$CS(\overline{WRW}(k))$と表し，後者を$CS_k(WRW^-)$と表す．以下3.1節で我々の$CS_k(WRW^-)$の概略を述べ，3.3節でその手続きを記述するが，比較のため3.2節で$CS(\overline{WRW}(k))$についても言及する．

3.1　動的版選択を行う$CS_k(WRW^-)$

〔定義3.1〕(2) ATIOグラフ（active TIO graph）$\mathrm{ATIO}(\langle P, I\rangle, q, \mathrm{PEND}) = (V \cup V', A \cup A')$とは，次の節点集合$V, V'$と有向枝集合$A, A'$（$V'$はダミー節点集合，$A'$はダミー有向枝集合）からなる．

V：$\{T_i \mid$ ある$s \in \mathrm{STEP}(T_i)$に対し $s \in Pq \cup \mathrm{PEND}\} \cup T_f$.

V'：$\{T_i' \mid I(R_j[d]) = W_i[d]$なる$T_j \in V$が存在しない$\}$

A：$\{(T_i, T_j):d \mid$ ある$d \in D$に対し$I(R_j[d]) = W_i[d]\}$

A'：$\{(T_i, T_i'):d\} \mid W_i[d] \in Pq$に対して $I(R_j[d]) = W_i[d]$を満たすT_jがない$\}$

$\cup \{(T_i, T_i'):d\} \mid W_i[d] \in \mathrm{PEND}\}$ □

$R_i[d] \in \mathrm{PEND}$の存在を明らかにするために，上の定義には含まれていないが，ATIOグラフ中にT_iに向かうラベルd付き有向枝を入れて考える（図3.1(a), (b)）．なお$W_i[d]$（$R_i[d]$）$\in \mathrm{PEND}$を<u>処理待ち書込み（読込み）ステップ</u>という．

ATIOグラフで，節点T_iに対し有向路$[T_j, T_i]$が存在するような節点T_jの集合をT_iの<u>祖先</u>といい，$T_{ASCEND(i)}$と表し，有向路$[T_i, T_j]$が存在するような節点T_jの集合をT_iの<u>子孫</u>といい，$T_{DESCEND(i)}$と表す．また$W_i[d] \in P$に対し，DITSにおいて$T_i \ll T_j$かつ$W_j[d] \in P$なるT_jが存在しないとき，$W_i[d]$を$W_{right(d)}[d]$と表す．ATIOグラフでのwr制約，rw制約は次のように考える．

(1) Ｐｑにおいて$R_i[d] < W_j[d]$のとき，または$R_i[d] \in Pq$, $W_j[d] \in \mathrm{PEND}$のときrw制約$T_i \ll T_j$．(2) Ｐｑにおいて$W_i[d] < R_j[d]$のとき，または$W_i[d] \in Pq$, $R_j[d] \in \mathrm{PEND}$のときwr制約$T_i \ll T_j$．

〔定義3.2〕<u>wrw^-制約</u>はwr制約のすべてとrw制約の一部を満たす．つまり，トランザクションT_iの読み込むデータ集合をRS_i，書き込むデータ集合をWS_iとすると，次の(A), (B)の制約をATIOグラフに付加する．

(A) トランザクションT_iの最初のリクエスト到着時
節点T_iを付加し，更に$d \in RS_i$に対して$W_j[d] \in P$なるトランザクションからwr制約$T_j \ll T_i$，$d \in WS_i$に対して$R_j[d] \in P$なるトランザクションからrw制約$T_j \ll T_i$を付加する．

(B) リクエストｑが(A)に該当するか否かにかかわらず
$q = R_i[d]$のとき：
有向路$[T_j, T_i]$が存在するような$W_j[d] \in \mathrm{PEND}$に対してrw制約$T_i \ll T_j$は考えない．有向路$[T_j, T_i]$が存在しない$W_j[d] \in \mathrm{PEND}$に対してrw制約$T_i \ll T_j$を付加する．
$q = W_i[d]$のとき：
$R_j[d] \in \mathrm{PEND}$に対してwr制約$T_i \ll T_j$を付加する．

(B)はｑが遅延されれば対応する有向枝を除くが(A)は除かない． □

ATIOグラフにwrw^-制約枝を加えた上での排除閉包を$\mathrm{ATIO}_{wrw^-}{}^*(\langle P, I_k^-\rangle, q, \mathrm{PEND})$と表し，以後は$G^-$と略記する．

〔定義3.3〕$W_j[d]$による版を$[d]$と表すが，特に書込んだトランザクション名を示すとき，$[d]_j$と表すことがある．

更に$CS_k(WRW^-)$における$d \in D$の版を区別するために$[d_{(n)}]$（$n = 0, 1, 2, \cdots$）と表すことがある．ここで版番号nは，$\mathrm{SPOIL}(d) = \{[d]_i \mid$ ある$W_j[d] \in Pq$に対し$[T_i, T_j]$がATIOグラフの有向路$\}$とおくとき，$\mathrm{SPOIL}(d)$以外の版についてのみ定義し，$W_j[d_{(n)}] \in Pq$のとき$1 \leqq n$，$W_j[d_{(n)}] \in \mathrm{PEND}$のとき$n = 0$とする．但し$1 \leqq n$については最新のものから$[d_{(1)}], [d_{(2)}], \cdots, [d_{(k)}], \cdots$とつける． □

〔定義3.4〕$CS_k(WRW^-)$では，各$d \in D$について$[d] \notin \mathrm{SPOIL}(d)$かつ版番号が$1 \leqq n \leqq k$なる版を残す．すなわち$CS_k(WRW^-)$における<u>現存版</u>（current version）$CV^-(d)$，<u>活性版</u>（active version）$AV^-(d)$，<u>不要版</u>（unreadable version）$UV^-(d)$は次のように定義される．

$CV^-(d) = \{[d_{(n)}] \mid 1 \leqq n \leqq k\}$,

$AV^-(d) = \{[d_{(n)}] \mid 0 \leqq n \leqq k\}$,

$UV^-(d) = \{[d] \mid [d] \in \mathrm{SPOIL}(d)\} \cup \{[d_{(n)}] \mid k < n\}$. □

k版完成テストにおいてG^-にDITSが存在するかどうかを調べるため，トランザクションの直列化を後ろから（T_fからT_0の方向へ）行うことを試み，直列化した節点を次々にグラフから除いていく．つまりG^-から一部のトランザクションを除いたグラフを$H^- = (V_{H^-} \cup V_{H^-}', A_{H^-} \cup A_{H^-}')$（$V_{H^-}'$, A_{H^-}'はH^-のダミー節点集合，ダミー有向枝集合）とする．直列化がすべてのトランザクションについて完成したときのトランザクション列をτといい，τを構成途上の列をσということにする．

〔定義3.5〕k版完成テストにおける主要な集合を以下のように定義する．

$D_{ACT}^- = D - \{d \mid \forall T_i \in \sigma, [d]_i \in CV^-(d)\}$

$T_{SELECT}^- = \{T_r \mid (\forall T_j \in V_{H^-}, (T_r, T_j) \notin A_{H^-}) \wedge (\forall d \in D_{ACT}^-, [d]_r \notin UV^-(d))\}$ □

すなわちD_{ACT}^-はσのトランザクションによる現存版のデータ集合をＤから除いたもの，T_{SELECT}^-はH^-における<u>シンク節点</u>（ダミー節点以外に子孫を持たない節点）のうちD_{ACT}^-に関して不要版を書込まないトランザクション集合である．T_{SELECT}^-に属するようなトランザクションT_rを直列化の候補として選び，T_rがD_{ACT}^-（初期設定はＤ）に対して現存版を書込むデータ項目をD_{ACT}^-から除いていくという操作を繰り返す（例3.2(B)）．

3.2　$CS(\overline{WRW}(k))$との機構の比較

〔定義3.6〕$CS(\overline{WRW}(k))$では各$d \in D$について新しい版から一律にk個を残す．従ってすべての版に版番号ｎをつけたとき，$CS(\overline{WRW}(k))$における現存版CV^+

(d)，活性版 $AV^+(d)$，不要版 $UV^+(d)$ は次のように定義される．但し $CS_k(WRW^-)$ とは残している版が異なるため，定義3.3に対して $[d^{(n)}]$ のように表す．

$CV^+(d)=\{[d^{(n)}] \mid 1 \leqq n \leqq k\}$,
$AV^+(d)=\{[d^{(n)}] \mid 0 \leqq n \leqq k\}$,
$UV^+(d)=\{[d^{(n)}] \mid k < n\}$. □

H^- と同様 $CS(\overline{WRW(k)})$ のATIOグラフ $G^+=ATIO_{wrw}{}^*(\langle P, I_k{}^+\rangle, q, PEND)$ の部分グラフ $H^+=(V_H{}^+ \cup V_H{}^{+\prime}, A_H{}^+ \cup A_H{}^{+\prime})$ を定義する．

〔定義3.7〕[7] $CS(\overline{WRW(k)})$ における版に関する関数 α, β および $T_{SELECT}{}^-$ に相当する $T_{SELECT}{}^+$ を定義する．

$\alpha(d, T_i)=1$ （$[d^{(0)}]_i$ のとき）
　　$=0$ （上記以外）
$\beta(d, T_i)=k-m$ （$[d^{(m)}]_i$, $1 \leqq m$ のとき）
　　$=\infty$ （上記以外）
$D_{ACT}{}^+=D-\{d \mid \forall T_i \in \sigma, [d]_i \in CV^+(d)\}$
$T_{SELECT}{}^+=\{T_r \mid (\forall T_j \in V_H{}^+, (T_r, T_j) \notin A_H{}^+)$
　　$\wedge (\forall d \in D_{ACT}{}^+, \sum_{T \in V_H{}^+} \alpha(d, T) \leqq \beta(d, T_r))\}$ □

$CS(\overline{WRW(k)})$ のk版完成テストは $CS_k(WRW^-)$ と全く同様である．$CS(\overline{WRW(k)})$ が正しく直列化可能スケジュールを出力する理由の詳細は，文献(7)を参照のこと．以下 $\sum_{T \in V_H{}^+}$ は $\sum$ と略す．

〔例3.1〕 $k=3$ 版，$I=I_k{}^+=I_k{}^-$ とする．

$Pq=W_0[D]R_1[x]W_1[y]R_2[y]W_5[u]R_6[z]$
　　$W_4[x, z]W_1[x]W_2[x, v]R_3[x]$,

$I(R_1[x])=W_0[x]$, $I(R_2[y])=W_1[x]$, $I(R_6[z])=W_0[z]$, $I_q(R_3[x])=W_4[x]$,

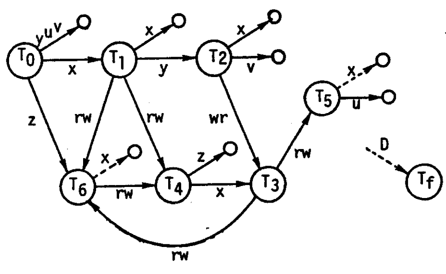
$PEND=\{W_5[x], W_6[x], R_f[D]\}$ に対するATIOグラフ G^+（図3.1(a)），G^-（図3.1(b)）を示す．図で点線は処理待ちステップの有向枝である．このとき G^+ には有向閉路があるためDITSは存在せず，$q=R_3[x]$ は遅延されてDELに入るが，G^- はwrw⁻制約でrw制約枝 (T_3, T_1), (T_3, T_6) を含まないため閉路がなく完成テストに成功し，q を遅延なく受付ける．なお版の区別について，$CS(\overline{WRW(k)})$ では，

$CV^+(x)=\{[x]_1, [x]_2, [x]_4\}$,
$AV^+(x)=\{[x]_1, [x]_2, [x]_4, [x]_5, [x]_6\}$,
$UV^+(x)=\{[x]_0\}$ となる．

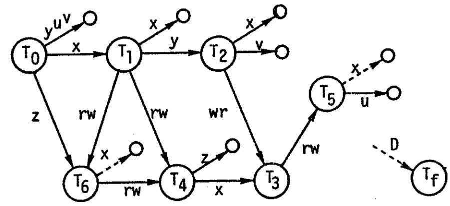
一方 $CS_k(WRW^-)$ では，

$CV^-(x)=\{[x]_2, [x]_4\}$,
$AV^-(x)=\{[x]_2, [x]_4, [x]_5\}$,
$UV^-(x)=\{[x]_0, [x]_1, [x]_6\}$ となる．

また $SPOIL(x)=\{[x]_0, [x]_1, [x]_6\}$. □



(a) $G^+ = ATIO_{wrw}{}^*(\langle P,I_k^+\rangle,q,PEND)$



(b) $G^- = ATIO_{wrw^-}{}^*(\langle P,I_k^-\rangle,q,PEND)$

図3.1 例3.1に対するATIOグラフ G^+, G^-

Fig. 3.1 ATIO graphs G^+ and G^- for Example 3.1

3.3 $CS_k(WRW^-)$ の手続き

以下で手続き $CS_k(WRW^-)$ を記述する．$CS(\overline{WRW(k)})$ は手続き中wrw⁻制約枝をwrw制約枝，「⁻」を「⁺」とし，手続きVERSIONとUPDATEの大半を除くこと等により容易に得られるので，記述を省略する．

```
手続きCSk(WRW⁻)〔k版先読みスケジューラ〕
CS0:(初期設定)
 P:=W0[D]; DEL:=φ; PEND:={Rf[D]}; Ik⁻:=(空関数);
 G⁻:=(V,A),V=T0,A=(T0,T0'):D;
 SPOIL(D):=φ; CV⁻(D):={[D]0};
for システム稼働中 do;
CS1:(新リクエストの処理)
 q:=(新リクエスト);
 if tr(q)が新トランザクション
 then begin;
        PEND:=PEND∪STEP(tr(q))−{q};
        G⁻にtr(q)の節点およびwrw⁻制約枝を加える;
      end;
 else PEND:=PEND−{q};
 手続きARCSの実行;
CS2:(完成テスト)
 手続きDCOMkの実行;
 if テストが成功(Yes)
 then begin;
        手続きVERSIONの実行;
        手続きUPDATEの実行;
        for すべてのDELのリクエスト do;
         q:=(DELの未検査のリクエスト);
         手続きARCSの実行;
         手続きDCOMkの実行;
         if テストが成功(Yes)
         then begin;
                手続きVERSIONの実行;
                手続きUPDATEの実行;
                PEND:=PEND−{q};
                DEL:=DEL−{q};
              end;
         else 手続きARCSで加えた
               qに関する有向枝をG⁻より除く;
        end;
      end;
 else begin;
        qを遅延し, DEL:=DEL∪{q}とする;
        PEND:=PEND∪{q};
        手続きARCSで加えた
```

```
        qに関する有向枝をG⁻より除く;
      end;
end;

手続きARCS〔qに対する有向枝をG⁻に付加する〕
if q=W_i[d] then (T_i, T_i'):dとwrw⁻制約枝,
      排除枝をG⁻に加える;
            else wrw⁻制約枝をG⁻に加える;

手続きDCOM_k〔k版完成テスト〕
DC0:(初期設定)
 D_ACT^- :=D; σ:=T_f; H^- :=G^- ;
 T_SELECT^- を計算する;
DC1:(トランザクションの直列化)
 for T_SELECT^- ≠ φ do;
  T_i ∈ T_SELECT^- なるT_iに対し, σ:=T_i σ;
  D_ACT^- :=D_ACT^- − {d | [d]_i ∈ CV^-(d)};
  節点T_iとT_iの付随有向枝をH^-より除く;
  T_SELECT^- を計算する;
 end;
DC2:(DITSの判定)
 if H^- ≠ φ then Noを出力;
            else Yesとτ:=σを出力;

手続きVERSION〔qの許可とqに対する版選択，版割当〕
 if q=R_i[d]
 then I_q(R_i[d])=W_right(d)[d]とし,
   (T_right(d), T_i):dと排除枝をG⁻に加える;
 else if [d]_i ∈ SPOIL(d)
       then qを許可するが版は残さない;
       else begin;
              if |CV^-(d)| ≧ k
              then 最古の版[d_(k)]を消し,
                   CV^-(d):=CV^-(d) − [d_(k)];
              else;
              qを許可して版を残し,
              CV^-(d):=CV^-(d) ∪ [d]_i;
            end;
 <P, I_k^-> := <Pq, I_k^- I_q>;

手続きUPDATE〔tr(q)の祖先の版選択処理〕
UP0:(初期設定)
 T_ASCEND(i), T_DESCEND(i)の計算;
 E:={d | ([d]_i ∈ CV^-(d)) ∨ ([d]_DESCEND(i) ∈ CV^-(d))};
UP1:(版選択)
 for すべてのe∈E do;
  for すべてのT_j ∈ T_ASCEND(i)の版[e]_j do;
   if [e]_j ∈ CV^-(e)
   then [e]_jを消し, CV^-(e):=CV^-(e) − [e]_j;
        SPOIL(e):=SPOIL(e) ∪ [e]_j;
  end;
 end;
```

3.4 手続きの正当性

以下補題3.1，補題3.2，定理3.1では手続きＤＣＯＭ$_k$の正当性を議論する．

〔定義3.8〕ｋ版完成テストが成功するとは，部分スケジュール$\langle P, I_k^- \rangle$，現在のリクエストｑとその割り当てI_q，およびＰＥＮＤ中の系列Ｑとその割り当てI_Qに関して以下の条件が成り立つことである．

(1) ＱはＰＥＮＤ中の全ステップからなる系列である．
(2) Ｑ中のステップの順序は各T_i中でのステップの順序に矛盾しない．
(3) $I_k^- I_q I_Q$によって割り当てられる版が割り当ての時点で存在する．
(4) $TIO_{wrw^-}(\langle PqQR_f[D], I_k^- I_q I_Q \rangle)$がDITSをもつ．

〔補題3.1〕G^-に有向閉路がないとき，ＰＥＮＤ中の系列Ｑとある割り当てI_Qに対し，$TIO_{wrw^-}(\langle PqQR_f[D], I_k^- I_q I_Q \rangle)$にDITSが存在する．

（証明）G^-に有向閉路がないとき節点の位相的ソート(topological sort)が可能で，ソート後のトランザクション列をτとする．次にτを未実行ステップを含むステップの列で置き換えた直列系列をμとし，$qQ=\mu/P$とおく（ただしs/s'はsからs'のステップを除いた系列）．またqQの任意の読込みステップをPqQにおける直前の書込みステップに割り当てるI_q, I_Qを考える．

(1) (T_i, T_j):d，(T_u, T_v):dがダミーでない有向枝の場合，$W_i[d] < W_u[d]$としても一般性を失わない．

(1a) PqQにおいて$W_i[d] < R_j[d] < W_u[d] < R_v[d]$のときもし有向路$[T_u, T_j]$があるとすると，wr制約枝$(T_i, T_v)$のため排除枝$(T_j, T_u)$が生じ，閉路があることになり仮定に矛盾．よって$[T_u, T_j]$は存在せず，wrw⁻制約の規則（定義3.2(B)）によりrw制約枝(T_j, T_u)が存在するため，$T_i \ll T_j \ll T_u \ll T_v$と直列化される．

(1b) PqQにおいて$W_i[d] < W_u[d] < R_j[d]$かつ$W_i[d] < W_u[d] < R_v[d]$のときwr制約枝(T_u, T_j)による排除枝(T_v, T_i)とwr制約枝(T_i, T_v)より閉路が生じるためこのような場合はない．

(1c) (1a)，(1b)以外のとき(T_i, T_j):d，(T_u, T_v):dの少なくとも一方が生じない．

(2) (T_i, T_i'):dがダミー有向枝，(T_u, T_v):dがダミーでない有向枝の場合

(2a) PqQにおいて$W_i[d] < R_v[d]$かつ$W_u[d] < R_v[d]$のときwr制約枝(T_i, T_v)による排除枝(T_i', T_u)のため，$T_i \ll T_i' \ll T_u \ll T_v$と直列化される．

(2b) PqQにおいて$W_u[d] < R_v[d] < W_i[d]$のとき有向路$[T_i, T_v]$が存在するとき，wrw⁻制約の規則により$q=R_v[d]$の時点でrw制約(T_v, T_i)は考えず，排除枝(T_i', T_u)のため$T_i \ll T_i' \ll T_u \ll T_v$と直列化される．$[T_i, T_v]$が存在しないとき，rw制約枝$(T_v, T_i)$のため$T_u \ll T_v \ll T_i \ll T_i'$と直列化される．

(2c) (2a)，(2b)以外のとき(T_u, T_v):dが生じない．

以上よりG^-に有向閉路がないとき，すべての$d \in D$に関して直列化が可能である．また２つのラベル付きダミー有向枝に関しては，各々のダミー節点をダミーでない節点の直後に置くような直列化が常に可能である．以上の直列化においてDITSの条件を満たす．（証明終）

〔補題3.2〕手続きＤＣＯＭ$_k$により生じたトランザクション列τにおいて，$T_{right(d)}$は$q=R_i[d]$に対してT_iの直前のｄ書込みトランザクションであり，$[d]_{right(d)} \in CV^-(d)$が成り立つ．

（証明）手続きＡＲＣＳでのwr制約枝$(T_{right(d)}, T_i)$のため，τにおいて$T_{right(d)} \ll T_i$と直列化されている．$T_{right(d)}$とT_iの間に他のｄ書込みトランザクションT_jが直列化されないことは，$W_j[d] \in P$については$W_{right(d)}[d]$の定義から，$T_{right(d)} \ll T_j$なる$W_j[d] \in$ＰＥＮＤについてはrw制約枝(T_i, T_j)が存在することより言える．次に$W_{right(d)}[d]$の定義より$[d_{(n)}]_{right(d)} \notin$ＳＰＯＩＬ(d

)かつ1≦nであり，$T_{right(d)}$が直列化候補として選ばれたとき，$d \in D_{ACT}^-$ゆえ$[d]_{right(d)} \notin UV^-(d)$．つまりn≦k．従って$[d]_{right(d)} \in CV^-(d)$．　（証明終）

〔定理3.1〕k版完成テストが成功するための必要十分条件は，手続きDCOM$_k$においてYesの判定がなされることである．

（証明）［十分性］DCOM$_k$においてYesの判定がなされるとき，トランザクション列τが生成される．このときT_{SELECT}^-の定義と手続きDCOM$_k$より，(a)τにおいて$T_i \ll T_j$ならばG^-に有向路$[T_j, T_i]$が存在せず，かつ，(b)すべての$d \in D$に関して$[d]_u \in UV^-(d)$なる任意のT_uに対し，$[d]_v \in CV^-(d)$なるあるT_vはτにおいて$T_u \ll T_v$を満たす（補題3.2）．いま補題3.1の証明と同様にしてQ，I_Qを構成する．このときqの読込みステップに対してはτの性質(b)より定義3.8(3)が満たされ，またQR_f[D]の各読込みステップに対しては，直列な実行で直前の書込みステップに割り当てれば定義3.8(3)が満たされる．

次にτの性質(a)よりG^-に有向閉路は存在しないため，補題3.1より定義3.8(4)が成り立つ．

また上のようなQのとり方により，定義3.8(1)，(2)は明らかに成り立つ．

［必要性］対偶をとる．すなわちDCOM$_k$においてNoの判定がなされるものとする．このとき$T_{SELECT}^- = \phi$かつ$H^- \neq \phi$である．そこで$T_{NEW} = \{T_r \mid \forall d \in D_{ACT}^-, [d]_r \notin UV^-(d)\}$とおくと

(i) $T_{NEW} \neq \phi$のとき

T_{SELECT}^-の定義により，$\{T_r \mid \forall T_j \in V_{H^-}, (T_r, T_j) \notin A_{H^-}\} = \phi$すなわち$H^-$に有向閉路があるため，$G^-$にDITSは存在しない．従ってTIO$_{WRW^-}$(<P q QR_f[D], I_k^- I_q I_Q>)にもDITSは存在せず，定義3.8(4)は成り立たない．

(ii) $T_{NEW} = \phi$のとき

各トランザクション$T_i \in V_{H^-}$が，ある$d \in D_{ACT}^-$に関して少なくとも1つの$[d]_i \in UV^-(d)$をもつ．従ってどのT_iを直列化候補として選んでも，$T_i \ll T_j$，$R_j[d] \in STEP(T_j)$なるT_j（σはT_fも含むのでこのようなT_jは必ず存在する）に対して定義3.8(3)または(4)が成り立たない．　（証明終）

定理3.2では版選択に関して議論する．

〔定理3.2〕手続きVERSIONと手続きUPDATEにより，$CV^-(d)$の定義に合致した版選択がなされる．

（略証）$q = W_i[d]$または$T_j \in T_{ASCEND(i)}$に対して，$[d]_i, [d]_j \in SPOIL(d)$を満たすような版は除かれなければならない．またk＜nとなるような版は手続きVERSIONにおいて除外され，現存版がk個を越えることはない．　（証明終）

〔例3.2〕ここでは例3.1でT_3が来なかった場合の<P, I>, q, PENDに対するCS($\overline{WRW}$(k))およびCS$_k$(WRW$^-$)のk版完成テストの例を示す．但し$I = I_k^+ = I_k^-$かつ同じ順序でトランザクションの直列化を行うものとする．初めG^+とG^-は図3.2(a)のようになっている．

(A)CS($\overline{WRW}$(k))のk版完成テスト

(A1)シンク節点はT_2, T_4, T_5で，$d \in D_{ACT}^+ (=D)$，$\Sigma\alpha(d, T) \leq \beta(d, T_r)$…(*)に対して，$(\Sigma\alpha(x,T)=2) \leq (\beta(x,T_2)=2)$かつ$(\Sigma\alpha(v,T)=0) \leq (\beta(v,T_2)=2)$より$T_2$は式(*)を満たす．$(\Sigma\alpha(x,T)=2) > (\beta(x,T_4)=0)$より$T_4$は式(*)を満たさない．$(\Sigma\alpha(x,T)=2) \leq (\beta(x,T_5)=\infty)$かつ$(\Sigma\alpha(u,T)=0) \leq (\beta(u,T_5)=2)$より$T_5$は式(*)を満たすから，$T_{SELECT}^+ = \{T_2, T_5\}$．そこで$\sigma = T_2T_5T_f$と直列化するとき，$T_2, T_5$の$D_{ACT}^+$に関する現存版のデータ項目{x, u, v}を$D_{ACT}^+$から除くと新$D_{ACT}^+ = \{y, z\}$．$H^+(=G^+)$より$T_2, T_5$とその付随有向枝を除く（図3.2(b)）．

(A2)シンク節点はT_4で，$D_{ACT}^+=\{y,z\}$に対して$(\Sigma\alpha(z,T)=0) \leq (\beta(z,T_4)=2)$より$T_4$は式(*)を満たすから$T_{SELECT}^+=\{T_4\}$．そこで$\sigma=T_4T_2T_5T_f$と直列化し，$T_4$の$D_{ACT}^+$に関する現存版のデータ項目{z}を$D_{ACT}^+$から除くと新$D_{ACT}^+=\{y\}$．$H^+$より$T_4$とその付随有向枝を除く（図3.2(c)）．

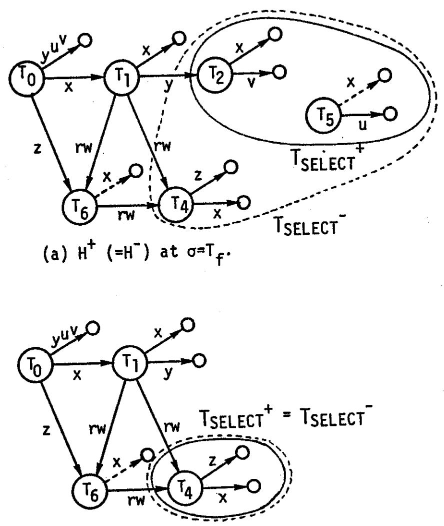
(A3)シンク節点はT_6で，$D_{ACT}^+=\{y\}$に対して$(\Sigma\alpha(y,T)=0) \leq (\beta(y,T_6)=\infty)$より$T_6$は式(*)を満たすから$T_{SELECT}^+=\{T_6\}$．そこで$\sigma=T_6T_4T_2T_5T_f$と直列化するが，$T_6$の現存版はないから新$D_{ACT}^+=\{y\}$．$H^+$より$T_6$とその付随有向枝を除く．以下同様$\sigma=T_0T_1T_6T_4T_2T_5T_f$と直列化することができる．

(B)CS$_k$(WRW$^-$)のk版完成テスト

(B1)シンク節点T_2, T_4, T_5は$D_{ACT}^-=D$に対して不要版を書込んでいないから$T_{SELECT}^-=\{T_2,T_4,T_5\}$．(A1)に合わせて$\sigma=T_2T_5T_f$と直列化するとき，$T_2,T_5$の$D_{ACT}^-$に関する現存版のデータ項目{x, u, v}を$D_{ACT}^-$から除くと新$D_{ACT}^-=\{y,z\}$．$H^-(=G^-)$より$T_2,T_5$とその付随有向枝を除く（図3.2(b)）．

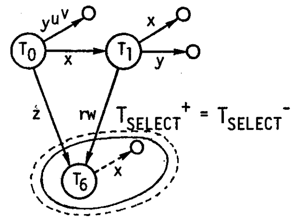
(B2)シンク節点T_4は$D_{ACT}^-=\{y,z\}$に対して不要版を書込んでいないから$T_{SELECT}^-=\{T_4\}$．そこで$\sigma=T_4T_2T_5T_f$と直列化し，T_4のD_{ACT}^-に関する現存版のデータ項目{z}をD_{ACT}^-から除くと，新$D_{ACT}^-=\{y\}$．H^-よりT_4とその付随有向枝を除く（図3.2(c)）．

(B3)シンク節点T_6は$D_{ACT}^-=\{y\}$に対して不要版を書込んでいないから$T_{SELECT}^-=\{T_6\}$．そこで$\sigma=T_6T_4T_2T_5T_f$と直列化するが，T_6の現存版はないから新$D_{ACT}^-=\{y\}$．H^-よりT_6とその付随有向枝を除く．以下(A3)と同様$\sigma=T_0T_1T_6T_4T_2T_5T_f$と直列化する．□



(a) H^+ $(=H^-)$ at $\sigma=T_f$.

(b) H^+ $(=H^-)$ at $\sigma=T_2T_5T_f$.



(c) H^{+} $(=H^{-})$ at $\sigma = T_4 T_2 T_5 T_f$.

図3.2　例3.2に対するk版完成テスト

Fig. 3.2 k-version completion tests for Example 3.2

4．先読みスケジューラの性能

4.1　無遅延クラスによる評価

スケジューラがどのステップも遅延させることなく出力するようなスケジュールの集合を考える．これを無遅延クラス(fixed point set)と呼び，CS($\overline{WRW}$(k))とCS$_k$(WRW$^-$)の無遅延クラスをそれぞれfp($\overline{WRW}$(k))，fp(WRW$_k^-$)と表す．以下でこの無遅延クラスの比較を行うが，両スケジューラとも複数個の版を保持しており，直列化に関しても任意性があるため，何らかの基準がないと例4.1のようにスケジュール集合の比較を行うことは不可能である．そこでG^+=ATIO$_{wrw}^+$(<P, I_k^+>, q, PEND)およびG^-=ATIO$_{wrw}^-$(<P, I_k^->, q, PEND)の割り当てが$I_k^+=I_k^-$の場合，可能な限り同順直列化（2つのグラフでトランザクションの順を同じにした直列化）を行うという条件下で，2つのスケジューラでの無遅延クラスの大きさすなわち並行処理性能を比較する．従って$H=(V_H \cup V_H', A_H \cup A_H')=H^+=H^-$と考える．

〔例4.1〕k＝2版とする．s＝$W_0[D]W_1[x]W_2[x]R_3[x]W_2[y]R_1[y]R_f[D]$に対してq＝$R_3[x]$のとき，$I(R_3[x])=W_1[x]$，$I'(R_3[x])=W_2[x]$とする．このとき同じsに対して$I=I_k^+$，$I'=I_k^-$のとき$s \in$fp($\overline{WRW}$(2))，$s \notin$fp(WRW$_2^-$)であるが，$I=I_k^-$，$I'=I_k^+$のとき$s \in$fp(WRW$_2^-$)，$s \notin$fp($\overline{WRW}$(2))である．すなわちIのように割り当てた場合のみ両スケジューラともsを遅延なしで受け付ける．□

〔補題4.1〕G^+, G^-にDITSが存在するような任意の<P, I>, q, PEND（$I=I_k^+=I_k^-$）に対して，qの割り当てI_qも等しくすることが可能である．

（証明）帰納法による．(i)P中に読込みステップがなくq＝$R_i[d]$が最初の読込みステップであるとする．このときdに関して$[d]_0$以外の版が存在しないなら，共に$I_q(R_i[d])=W_0[d]$とすることで題意を満たす．また$[d]_0$以外の版が存在するとき，G^+およびG^-でT_0, tr(q)以外は直列化が任意だから，最新の$[d_{(1)}]$または$[d^{(1)}]$（共に現存版である）をqに割り当てればよい．

(ii)P, q, PEND（但しPは少なくとも1つの読込みステップを含む）に対して，上記二つのグラフで<P, I>，$I=I_k^+=I_k^-$と割り当てられているものとする．いま制約枝の定義よりP中のトランザクションに関して（G^-の有向枝）⊆（G^+の有向枝）であるため，G^-でのトランザクションの直列化をG^+における直列化と同じにすることが可能である．このとき任意の$d \in D$に対してG^-での$T_{right(d)}$はG^+でのトランザクションと一致する．またDITSの順における$T_{right(d)}$はG^+およびG^-で現存版である．すなわちG^+およびG^-で等しいI_qが可能である．

(i), (ii)より本補題は証明された．　（証明終）

補題4.1よりG^+, G^-で容易に同順直列化を実行できることが示せる．

〔定義4.1〕CS($\overline{WRW}$(k))，CS$_k$(WRW$^-$)における$d \in D$に関する適格版（eligible version）$EV^+(d)$，$EV^-(d)$を次のように定義する．

$EV^+(d)=\{[d]_r \mid T_r \in T_{SELECT}^+$かつ$d \in D_{ACT}^+\}$，

$EV^-(d)=\{[d]_r \mid T_r \in T_{SELECT}^-$かつ$d \in D_{ACT}^-\}$．□

つまり適格版は，σ中のトランザクションの読込みステップが割り当て可能な版である．

〔補題4.2〕G^+, G^-にDITSが存在するような任意の<P, I>, q, PEND（$I=I_k^+=I_k^-$），任意の$d \in D$に対して以下が成り立つ．

(1) $CV^+(d)-SPOIL(d) \subseteq CV^-(d)$，
(2) $AV^+(d)-SPOIL(d) \subseteq AV^-(d)$，
(3) $CV^-(d) \subseteq CV^+(d) \cup UV^+(d)$．

（略証）版の定義よりそれぞれ明らか．　（証明終）

〔補題4.3〕G^+, G^-にDITSが存在するような任意の<P, I>, q, PEND（$I=I_k^+=I_k^-$），$d \in D_{ACT}^+$に対して，同順直列化の各時点において

$$EV^+(d) \subseteq AV^-(d).$$

（証明）$[d]_v \in EV^+(d)$とする．D_{ACT}^+の定義より，ATIOグラフG^+において$[T_v, T_i]$かつ$[d]_i \in CV^+(d)$なる$W_i[d]$は存在しない．従って$[d]_v \notin SPOIL(d)$．しかも$[d]_v \in EV^+(d)$より$[d]_v \in AV^+(d)$でなければならない．なぜならもし$[d]_v \in UV^+(d)$とすると$\beta(d, T_v)<0$となり，一方常に$\Sigma\alpha(d, T) \geq 0$ゆえ$T_v \notin T_{SELECT}^+$となるからである．すなわち$[d]_v \in AV^+(d)-SPOIL(d)$．従って補題4.2(2)より$[d]_v \in AV^-(d)$．　（証明終）

〔補題4.4〕G^+, G^-にDITSが存在するような任意の<P, I>, q, PEND（$I=I_k^+=I_k^-$）に対し，同順直列化の各時点において

$$D_{ACT}^+=D_{ACT}^-.$$

（証明）直列化の任意の時点でのトランザクション列σを考える．このときCS($\overline{WRW}$(k))とCS$_k$(WRW$^-$)においてσが同じ長さなら一致する．いま$d \in D-D_{ACT}^+$，$[d]_i \in CV^+(d)$，$T_i \in \sigma$とすると，dがD_{ACT}^+より除かれるとき$[d]_i \notin SPOIL(d)$だから，$[d]_i \in CV^+(d)-SPOIL(d)$なる$[d]_i$が存在する．従って補題4.2(1)より$[d]_i \in CV^-(d)$なる$[d]_i$が存在するから，定義3.5より$d \in D-D_{ACT}^-$．従って$D_{ACT}^- \subseteq D_{ACT}^+$．逆に$d \in D_{ACT}^+$のとき，任意の$T_j \in \sigma$について$[d]_j \in CV^+(d) \cup UV^+(d)$なる$[d]_j$は存在しない．従って補題4.2(3)より任意の$T_j \in \sigma$について$[d]_j \in CV^-(d)$なる$[d]_j$は存在しないため定義3.5より$d \in D_{ACT}^-$．すなわち$D_{ACT}^+ \subseteq D_{ACT}^-$．よって証明された．　（証明終）

〔補題4.5〕G^+, G^-にDITSが存在するような任意の<P, I>, q, PEND（$I=I_k^+=I_k^-$）に対し，同順直列化の各時点において

$$T_{SELECT}^+ \subseteq T_{SELECT}^-.$$

（証明）$T_r \in T_{SELECT}^+$とすると，T_rの$d \in D_{ACT}^+$に対する任意の版は，$[d]_r \in EV^+(d)$である．従って補題4.3，補題4.4よりT_rの$d \in D_{ACT}^-$に対する任意の版は$[d]_r \in AV^-(d)$すなわち$[d]_r \notin UV^-(d)$を満たす．$T_r \in T_{SELECT}^+$より，任意の$T_j \in V_H$に対して$(T_r, T_j) \notin A_H$は明らかゆえ，$T_r \in T_{SELECT}^-$．　（証明終）

補題4.2については例3.1，補題4.4，4.5については例3.2を参照のこと．

CS($\overline{WRW}$(k))とCS$_k$(WRW$^-$)の無遅延クラスに関して次の定理が成り立つ．

〔定理4.1〕CS($\overline{WRW}$(k))とCS$_k$(WRW$^-$)において

版割り当てを$I_k^+ = I_k^-$とし，同順直列化を行うものとすると，その無遅延クラスに関して

(1) $\mathrm{fp}(\overline{WRW}(k)) \subsetneq \mathrm{fp}(WRW_k^-)$,

(2) $\mathrm{fp}(\overline{WRW}(k+1)) \not\subseteq \mathrm{fp}(WRW_k^-)$,
$\mathrm{fp}(WRW_k^-) \not\subseteq \mathrm{fp}(\overline{WRW}(k+1))$,

(3) $\mathrm{fp}(WRW_1^-) \not\subseteq \mathrm{fp}(\overline{WRW}(k))$,

(4) $\mathrm{fp}(WRW_k^-) \subsetneq \mathrm{fp}(WRW_{k+1}^-)$

が成り立つ（但し$k=1,2,\cdots$）.

（証明）(1) 補題4.5より，直列化の任意の時点において$T_{SELECT}^+ \subseteq T_{SELECT}^-$. 従って$T_{SELECT}^+ \neq \phi$ならば$T_{SELECT}^- \neq \phi$すなわちCS($\overline{WRW}(k)$)で直列化候補が存在するときはいつでも$CS_k(WRW^-)$で直列化候補が存在し，遅延されることはない. 従って無遅延クラスについて$\mathrm{fp}(\overline{WRW}(k)) \subseteq \mathrm{fp}(WRW_k^-)$. ここで

$$a = W_0[D]R_1[y]W_1[y]R_2[y]W_2[y]\cdots R_{k+1}[y]W_{k+1}[y]$$
$$W_{k+1}[x]W_k[x]W_{k-1}[x]\cdots W_1[x]R_f[D]$$

とおくと，$a \notin \mathrm{fp}(\overline{WRW}(k))$かつ$a \in \mathrm{fp}(WRW_k^-)$ゆえ，$\mathrm{fp}(\overline{WRW}(k)) \subsetneq \mathrm{fp}(WRW_k^-)$.

(2) aでkをk＋1と置き換えたスケジュールをa'とすると，$a' \in \mathrm{fp}(WRW_k^-)$かつ$a' \notin \mathrm{fp}(\overline{WRW}(k+1))$ゆえ$\mathrm{fp}(\overline{WRW}(k+1)) \not\subseteq \mathrm{fp}(WRW_k^-)$. また，

$$b = W_0[D]R_{k+1}[y]W_{k+1}[y]R_1[y]W_1[x]R_2[y]$$
$$W_2[x]\cdots R_k[y]W_k[x]R_{k+1}[x]R_f[D]$$

とおくと，$b \in \mathrm{fp}(\overline{WRW}(k+1))$かつ$b \notin \mathrm{fp}(WRW_k^-)$ゆえ$\mathrm{fp}(WRW_k^-) \not\subseteq \mathrm{fp}(\overline{WRW}(k+1))$.

(3) $a \in \mathrm{fp}(WRW_1^-)$かつ$a \notin \mathrm{fp}(\overline{WRW}(k))$だから$\mathrm{fp}(WRW_1^-) \not\subseteq \mathrm{fp}(\overline{WRW}(k))$が成り立つ.

(4) $\mathrm{fp}(WRW_k^-) \subseteq \mathrm{fp}(WRW_{k+1}^-)$は明らかだから，$b \notin \mathrm{fp}(WRW_k^-)$かつ$b \in \mathrm{fp}(WRW_{k+1}^-)$ゆえ$\mathrm{fp}(WRW_k^-) \subsetneq \mathrm{fp}(WRW_{k+1}^-)$. 図4.1を参照のこと.　（証明終）

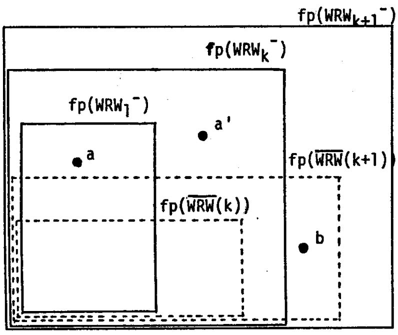


図4.1 無遅延クラスの包含関係

Fig. 4.1 Inclusion relationship of fixed point sets

4.2 取消し異常による評価

先読みスケジューラはトランザクション発行時にステップの集合を宣言するが，トランザクション実行中のステップ変更のうち，宣言したステップを取り消しても以後のスケジュールに支障を生じる(取消し異常：cancellation anomaly)かどうかにより，性能評価を行うことができる[(2),(3),(7)]. CS($\overline{WRW}(k)$)では取消し異常がないことが明らかにされている[(7)]が，我々のスケジューラでも以下のことが成り立つ.

〔定理4.2〕$CS_k(WRW^-)$は取消し異常を生じない.

（証明）$\langle PqQR_f[D], I_k^- I_q I_Q \rangle$にDITSが存在するような任意の$\langle P, I_k^- \rangle$, q, PEND（QはPEND上の系列）に対するG^-を考える. まずある$R_i[d] \in$ PENDが取り消されたときは，G^-において処理待ち読込み枝を削除するだけだから，結果のグラフでDITSが存在するのは明らかである. 次にある$W_i[d] \in$ PENDが取り消されたとき，$I_Q(R_j[d]) = W_i[d]$なる$R_j[d] \in$ PENDについて，もしDITSτにおいてT_jの直前に$W_r[d] \in$ PENDを持つT_rが直列化されていれば，新たに$I_Q'(R_j[d]) = W_r[d]$とすればよい. またそのようなT_rが存在しなければ，$T_{right(d)}$がT_jの直前のd書込みトランザクションであり，かつ補題3.2より$[d]_{right(d)} \in CV^-(d)$だから$I_Q(R_j[d]) = W_{right(d)}[d]$とすることが可能である. いずれの場合も$\langle PqQR_f[D]/W_i[d], I_k^- I_q I_Q' \rangle$にDITSが存在するので取消し異常は起こらない.　（証明終）

4.3 実行時間による評価

〔定理4.3〕手続き$CS_k(WRW^-)$は多項式時間で実行可能である.

（証明）手続き中最も実行時間が長いのは$DCOM_k$のDC1である. これは文献(7)の完成テストと本質的に同じ方法なので多項式時間で実行できることは明かである.　（証明終）

5. むすび

本稿では，必要と思われる版を優先的に選んで残すという文献(5)の方法を取り入れて，既に発表されているk版先読みスケジューラCS($\overline{WRW}(k)$)を拡張したスケジューラ$CS_k(WRW^-)$を提案した. 更に$CS_k(WRW^-)$は，CS($\overline{WRW}(k)$)と同様取消し異常がなく多項式時間で処理可能である上に，CS($\overline{WRW}(k)$)より並行処理性能の優れたスケジューラであることを示した.

謝辞　日頃ご指導をいただく京都大学長谷川利治教授に深謝いたします. また本研究に際し，温かいお励ましと貴重な御助言をいただいた，京都大学茨木俊秀教授，増山繁助手，大阪大学西尾章治郎助教授，カナダ国Simon Fraser大学亀田恒彦教授に深謝いたします.

文献

(1) Ibaraki, T., Kameda, T. and Minoura, T.: "Serializability with Constraints", ACM Trans. Database Syst., 12, 3, pp.429-452 (1987).

(2) Ibaraki, T., Kameda, T. and Katoh, N.: "Cautious Transaction Schedulers for Database Concurrency Control", IEEE Trans. Softw. Eng., 14, 7, pp.997-1009 (1988).

(3) Ibaraki, T., Kameda, T. and Katoh, N.: "Cautious Transaction Schedulers for Multiversion Database Systems", LCCR TR86-2, Dept. of Comput. Sci., Simon Fraser Univ., Canada(1986).

(4) Katoh, N., Ibaraki, T. and Kameda, T.: "Cautious Transaction Schedulers with Admission Control", ACM Trans. Database Syst., 10, 2, pp.205-229 (1985).

(5) 木庭, 加藤: "動的な版選択を行う1版先読みスケジューラ", 信学論(D)掲載予定.

(6) Sy, C.: "Efficient Schedulers in Multiversion Database Systems", Master Thesis, Faculty of Appl. Sci. Simon Fraser Univ., Canada (1986).

(7) 武田, 増山, 茨木: "版数制限をもつ先読みスケジューラ", 信学論(D), J70-D, 8, pp.1478-1486 (1986).