Eidgenössische Technische Hochschule Zürich
Ecole polytechnique fédérale de Zurich
Politecnico federale di Zurigo
Swiss Federal Institute of Technology Zurich

Prof. H.-J. Schek *A. Feßler, U. Röhm, H. Schuldt, R. Weber*

Informationssysteme IS-G SS 1997

Übung 11 Beispiellösung

Aufgabe 1: Konfliktpaare, Liest-von-Paare

Bestimmen Sie die Konfliktpaare und RF-Paare (Liest-von-Paare) des folgenden Schedules der Transaktionen T_1, T_2, T_3:

$$S = < R_1(a), W_1(a), W_1(b), W_2(b), R_3(a), R_3(b), W_3(c), R_1(c), W_1(c), R_2(b), R_1(b) >$$

Hinweis: Berücksichtigen Sie die künstliche Transaktion T_0, welche alle Objekte schreibt, die in S gelesen werden, nur für die Bestimmung der RF-Paare, nicht aber für die Konfliktpaare.

Die Konfliktpaare (CON) und RF-Paare sind:

$$\begin{aligned} CON &= \{(W_1(a), R_3(a)),\ (W_1(b), W_2(b)),\ (W_1(b), R_3(b)),\ (W_1(b), R_2(b)), \\ &\quad (W_2(b), R_3(b)),\ (W_2(b), R_1(b)),\ (W_3(c), R_1(c)),\ (W_3(c), W_1(c))\} \\ RF &= \{(T_1, a, T_0),\ (T_3, a, T_1),\ (T_3, b, T_2),\ (T_1, c, T_3),\ (T_1, b, T_2)\} \end{aligned}$$

Aufgabe 2: Rücksetzbarkeit

Die Rücksetzbarkeit von Transaktionen wird zu jedem Zeitpunkt der Bearbeitung gefordert.

a) Warum wird diese Forderung aufgestellt? Begründen Sie Ihre Antwort!
Von einer Transaktion wird unter anderem gefordert, dass sie als atomare Operation erscheint. Die Frage ist nun, ob zu einem **beliebigen** *Zeitpunkt diese Operation abgebrochen werden kann, so dass bereits getätigte Teiloperationen rückgängig gemacht werden müssen. Dies ist aber sicherlich der Fall, da allein schon der (das) Anwender(-programm) jederzeit ein Abbruch auslösen kann.*

b) Um Transaktionen möglichst parallel ablaufen zu lassen, sollen Sperren so kurz wie möglich gehalten werden. Wir betrachten Transaktionen mit Lese- und Schreibaktionen, die Lesesperren (S) und Schreibsperren (X) anfordern. Wir wollen die beiden Fälle untersuchen, in welchen diese Sperren entweder i) bereits *vor Ende* oder ii) erst *am Schluss* der Transaktion freigegeben werden können.
Wann ergeben sich für Lese- oder Schreibsperren bei den Alternativen i) oder ii) Probleme bei der Rücksetzbarkeit? Beschreiben Sie die möglichen Fehlersituationen.
Mögliche Probleme bei der Freigabe von Sperren:

	Sperrfreigabe vor Ende *der Transaktion*	*Sperrfreigabe* am Schluß *der Transaktion*
S-Sperre	*kein Problem*	*kein Problem*
X-Sperre	**Problem!**	*kein Problem*

Die einzige Situation, in der es Probleme geben kann, ist die vorzeitige Freigabe von X-Sperren. In diesem Fall würden geschriebene/veränderte Objekte vor dem Commit *einer Transaktion anderen Transaktionen zugänglich gemacht! Dies ist unannehmbar, sei es für den Fall, daß eine andere Transaktion diese Werte 'nur' liest (und aufgrund dessen Entscheidungen fällt) oder sogar verändert.*

c) Sind Schedules des 2PL-Protokolls rücksetzbar (im Sinne der Definition auf Seite 10-5)?
Nein, da Sperren vor dem Commit resp. Abort freigegeben werden, können andere Transaktionen Änderungen bereits lesen. Z.B. kann der Schedule:

$$S = < W_1(a), R_2(a), W_2(a), C_2, C_1 >$$

mit dem 2PL-Protokoll entstanden sein. Es gilt $RF(T_2, a, T_1)$*, nicht aber* $C_1 < C_2$.

d) Sind Schedules des strikten 2PL-Protokolls rücksetzbar (im Sinne der Definition auf Seite 10-5)?
Ja. Es gilt sogar, dass striktes 2PL rigorose Schedules erzeugt. Falls ein Konfliktpaar $o^i < o^k$ *existiert, muss die Transaktion* T_k *warten, bis* T_i *die Sperre freigegeben hat. Dies kann aber erst nach einem Commit oder Abort geschehen, so dass die Forderung der Rigorosität erfüllt ist (siehe Definition auf Seite 10-6).*

Aufgabe 3: Rücksetzbarkeit

Gegeben seien die folgenden zwei Transaktionen:

$$\begin{aligned} T_1 &= < R_1(p), W_1(p), W_1(s) > \\ T_2 &= < W_2(s), R_2(p), W_2(p) > \end{aligned}$$

a) Im folgenden Schedule repräsentieren die Operationen X und Y Commit- oder Abortoperationen:

$$S_1 = < R_1(p), W_1(p), W_1(s), W_2(s), R_2(p), W_2(p), X, Y >$$

Überlegen Sie sich, in welcher Klasse (RC, ACA, ST und RG) der Schedule S_1 für folgende Belegungen von X und Y liegt.
Für Kaskadenfreiheit müsste $C_1 < R_2(p)$ *gelten, da* T_2 *von* T_1 *liest. Dies kann aber mit keiner Belegung von* X *und* Y *erfüllt werden, so dass der Schedule nicht kaskadenfrei ist und damit auch nicht strikt und rigoros sein kann. Damit der Schedule rücksetzbar ist, muss* $C_1 < C_2$ *gelten, sofern* C_2 *in* S_1 *enthalten ist.*

- $X = C_1,\ Y = C_2$
  $S_1 \in RC,\ S_1 \notin ACA,\ S_1 \notin ST,\ S_1 \notin RG$
- $X = C_2,\ Y = C_1$
  $S_1 \notin RC,\ S_1 \notin ACA,\ S_1 \notin ST,\ S_1 \notin RG$
- $X = A_2,\ Y = C_1$
  $S_1 \in RC,\ S_1 \notin ACA,\ S_1 \notin ST,\ S_1 \notin RG$
- $X = C_2,\ Y = A_1$. Was fällt Ihnen hier speziell auf?
  $S \notin RC,\ S_1 \notin ACA,\ S_1 \notin ST,\ S_1 \notin RG$
  Transaktion T_2 müsste wegen dem Abort von T_1 trotz des Commits zurückgesetzt werden, da der ausgelesene Wert des Datenobjektes p nicht richtig war (Domino-Effekt).
- $X = A_1,\ Y = A_2$. Was fällt Ihnen hier speziell auf?
  $S_1 \notin RC,\ S_1 \notin ACA,\ S_1 \notin ST,\ S_1 \notin RG$
  In der Regel wird für das Rücksetzen einer Transaktion das before image *der betroffenen Datenobjekte geschrieben. Bei obiger Belegung von X und Y bedeutet dies, dass zuerst die before images von p und s der Transaktion T_1 geschrieben werden und danach jene von Transaktion T_2. Die before images von T_2 sind aber genau die after images von T_1, so dass am Schluss die Änderungen von T_1 sichtbar bleiben!*

Betrachten Sie den folgenden Schedule, in welchem die Commit- resp. Abortoperationen der Transaktionen noch nicht enthalten sind:

$$S_2 \quad = \quad < R_1(p), W_1(p), W_2(s), W_1(s), R_2(p), W_2(p) >$$

b) In den Schedule S_2 sollen an beliebiger Stelle Commit- resp. Abortoperationen für die Transaktionen T_1 und T_2 eingefügt werden. Formulieren Sie exakt, welche Bedingungen erfüllt sein müssen, damit der ergänzte Schedule in RC, ACA, ST resp. RG zu liegen kommt.

Grundsätzlich steht die Commit- resp. Abortoperation am Schluss der Transaktionen, d.h. alle Operationen der Transaktion müssen vor dem Commit resp. Abort stehen. Also muss

$$C_1 > W_1(s) \ \textit{resp.}\ A_1 > W_1(s) C_2 > W_2(p) \ \textit{resp.}\ A_2 > W_2(p)$$

gelten. Im weitern bestimmt man alle Liest-von-Paare (RF) und alle Konfliktpaare (CON); die künstliche Anfangstransaktion T_0 wurde zur Vereinfachung weggelassen:

$$\begin{aligned} RF &= \{(T_2, p, T_1)\} \\ CON &= \{(R_1(p), W_2(p)),\ (W_1(p), R_2(p)),\ (W_1(p), W_2(p)),\ (W_2(s), W_1(s))\} \end{aligned}$$

Damit S_2 in RC zu liegen kommt, muss

$$(T_2, p, T_1) \wedge C_2 \in S_2 \Longrightarrow C_1 < C_2$$

gelten, d.h. falls T_2 committed, muss T_1 vorher committen (siehe Definition von Rücksetzbarkeit auf Seite 10-5). Für die Zugehörigkeit zu ACA, muss

$$(T_2, p, T_1) \Longrightarrow C_1 < R_2(p)$$

erfüllt sein, d.h. T_1 muss vor dem Auslesen von p durch T_2 committen (siehe Definition von Kaskadenfreiheit auf Seite 10-5). Die Definition von Striktheit (Seite 10-6) bedeutet, dass

$$\begin{aligned} (W_1(p), R_2(p)) \in CON &\Longrightarrow C_1 < R_2(p) \ \textit{resp.}\ A_1 < R_2(p) \\ (W_1(p), W_2(p)) \in CON &\Longrightarrow C_1 < W_2(p) \ \textit{resp.}\ A_1 < W_2(p) \\ (W_2(s), W_1(s)) \in CON &\Longrightarrow C_2 < W_1(s) \ \textit{resp.}\ A_1 < W_1(s) \end{aligned}$$

gelten muss. Mit der gegebenen Ordnung in S_2 folgt:

$$C_2 < W_1(s) < R_2(p) \ \textit{resp.}\ A_2 < W_1(s) < R_2(p).$$

Die Transaktion T_2 müsste also vor $R_2(p)$ beendet werden, was aber nicht möglich ist. Daraus folgt, dass der Schedule S_2 nicht strikt und damit auch nicht rigoros gemacht werden kann.

c) Erweitern Sie den Schedule S_2 um Commit-/Abortoperationen zu einem Schedule S_3, so dass $S_3 \in RC$ und $S_3 \notin ACA$ gelten.

Wir nehmen an, dass beide Transaktionen committen. Aus den obigen Überlegungen geht hervor, dass S_3 in RC liegt, wenn

$$C_1 > W_1(s) \ \wedge \ C_2 > W_2(p) \ \wedge \ C_1 < C_2$$

erfüllt ist, und dass S_3 nicht in ACA liegt, wenn

$$C_1 \not< R_2(p)$$

gilt. Also erfüllt der folgende Schedule die Bedingungen:

$$S_3 \quad = \quad < R_1(p), W_1(p), W_2(s), W_1(s), R_2(p), W_2(p), C_1, C_2 >$$

Hinweis: Es gibt noch mehr Lösungen.

d) Erweitern Sie den Schedule S_2 um Commit-/Abortoperationen zu einem Schedule S_4, so dass $S_4 \in ACA$ und $S_4 \notin ST$ gelten.

Wiederum betrachten wir einen Schedule, bei dem beide Transaktionen committen. Für die Zugehörigkeit zu ACA muss

$$C_1 > W_1(s) \ \wedge \ C_2 > W_2(p) \ \wedge \ C_1 < C_2 \ \wedge \ C_1 < R_2(p)$$

erfüllt sein. Da Striktheit nicht erreichbar ist, genügt der folgende Schedule den Anforderungen:

$$S_4 \quad = \quad < R_1(p), W_1(p), W_2(s), W_1(s), C_1, R_2(p), W_2(p), C_2 >$$

Hinweis: Es gibt noch mehr Lösungen.

e) Erweitern Sie den Schedule S_2 um Commit-/Abortoperationen zu einem Schedule S_5, so dass $S_5 \in ST$ gilt.

Wie oben gezeigt wurde, kann S_2 nicht zu einem strikten Schedule ergänzt werden.

f) Geben Sie einen Schedule S_6 für die Transatkionen T_1 und T_2 an, der die Rigorositätsbedingung erfüllt. Mit welchem Protokoll könnten Sie einen solchen Schedule erzeugen?

Das strikte 2PL-Protokoll erzeugt rigorose, konflikt-serialisierbare Schedules, sofern keine Verklemmung (dead lock) auftritt:

$$S_6 = \; < R_1(p), W_1(p), W_1(s), C_1, W_2(s), R_2(p), W_2(p), C_2 >$$

Aufgabe 4: Funktionalabhängigkeiten

a) Beweisen Sie die Ableitungsregeln Erweiterung (A2) und Transitivität (A3) von Armstrong unter Benutzung der Definition der Funktionalabhängigkeit.

Es sei $X \subseteq sch(R), Y \subseteq sch(R), Z \subseteq sch(R)$.

Definition FA:

$X \rightarrow Y := \forall t_1, t_2 \in val(R) : t_1.X = t_2.X \Rightarrow t_1.Y = t_2.Y$

A2: $X \rightarrow Y \Rightarrow XZ \rightarrow YZ$

$$\begin{aligned} \forall t_1, t_2 \in val(R) : t_1.XZ = t_2.XZ &\stackrel{(1)}{\Longrightarrow} t_1.X = t_2.X \wedge t_1.Z = t_2.Z \\ &\stackrel{X\rightarrow Y}{\Longrightarrow} t_1.Y = t_2.Y \wedge t_1.Z = t_2.Z \\ &\stackrel{(1)}{\Longrightarrow} t_1.YZ = t_2.YZ \end{aligned}$$

□

A3: $X \rightarrow Y \wedge Y \rightarrow Z \Rightarrow X \rightarrow Z$

$$\begin{aligned} \forall t_1, t_2 \in val(R) : \quad & t_1.X = t_2.X \Rightarrow t_1.Y = t_2.Y \\ \wedge \quad & t_1.Y = t_2.Y \Rightarrow t_1.Z = t_2.Z \\ \stackrel{(2)}{\Longrightarrow} \quad & X \rightarrow Y \wedge Y \rightarrow Z \Rightarrow X \rightarrow Z \end{aligned}$$

□

(1) Die Gleichheit von Tupeln ist definiert über die Gleichheit der Attribute.

(2) Transitivität der Implikation.

b) Beweisen Sie weiter die abgeleitete Zerlegungsregel (A6) unter Benutzung der Ableitungsregeln A1 – A3.

A6: $X \rightarrow Y \wedge Z \subseteq Y \Rightarrow X \rightarrow Z$

$$\begin{aligned} Z \subseteq Y &\stackrel{A1}{\Longrightarrow} Y \rightarrow Z \\ X \rightarrow Y \wedge Y \rightarrow Z &\stackrel{A3}{\Longrightarrow} X \rightarrow Z \end{aligned}$$

□

Welche der folgenden Aussagen sind wahr?

c) $A \rightarrow B,\ B \rightarrow C \implies A \rightarrow C$

$$A \rightarrow B \wedge B \rightarrow C \stackrel{A3}{\Longrightarrow} A \rightarrow C$$

□

d) $A \rightarrow B,\ A \rightarrow C \iff A \rightarrow BC$

$$\begin{aligned} A \rightarrow B \wedge A \rightarrow C &\stackrel{A4}{\Longrightarrow} A \rightarrow BC \\ A \rightarrow BC &\stackrel{A6,A6}{\Longrightarrow} A \rightarrow B \wedge A \rightarrow C \end{aligned}$$

□

e) $A \rightarrow B,\ C \rightarrow B \iff AC \rightarrow B$

$$\begin{aligned} A \rightarrow B &\stackrel{A2}{\Longrightarrow} AC \rightarrow BC \\ &\stackrel{A6}{\Longrightarrow} AC \rightarrow B \end{aligned}$$

oder :

$$\left.\begin{aligned} A \rightarrow B &\stackrel{A2}{\Longrightarrow} AC \rightarrow BC \\ C \rightarrow B &\stackrel{A2}{\Longrightarrow} BC \rightarrow B \end{aligned}\right\} \stackrel{A3}{\Longrightarrow} AC \rightarrow B$$

$$AC \rightarrow B \stackrel{(1)}{\not\Longrightarrow} A \rightarrow B \wedge C \rightarrow B$$

(1) Gegenbeispiele:

R	*A*	*B*	*C*
	67	*7*	*otto*
	67	*3*	*ralph*
	45	*5*	*otto*

Da keine Voraussetzungen über A und C gemacht wurden, könnte für C auch die leere Attributsmenge gewählt werden. Die Implikation $C \rightarrow B$ wäre dann nicht erfüllt!

□

f) Für alle R(A,B,C) gilt: $R = \pi_{A,C}(R) \bowtie \pi_{A,B}(R) \implies A \rightarrow B$ *in R*

Die Implikation gilt nicht! Ein Gegenbeispiel wäre:

R	*A*	*B*	*C*
	1	*1*	*1*
	1	*2*	*1*

□

Aufgabe 5: Funktionalabhängigkeiten, Schlüsselkandidaten

Gegeben sei das relationale Schema `EXAM(S,F,R,P)` mit

`EXAM`	Examen
`S`	Student
`F`	Fach
`R`	Rang
`P`	Prüfer

und den Funktionalabhängigkeiten

$$\{ \texttt{FS} \longrightarrow \texttt{P},\ \texttt{FR} \longrightarrow \texttt{S},\ \texttt{FS} \longrightarrow \texttt{R} \}$$

Wir setzen hier voraus, dass jeder Student in einer Prüfung eine Platzziffer (Rang) bekommt, d.h. ausschliesslich relativ bewertet wird.

a) Beschreiben Sie die angegebenen Funktionalabhängigkeiten mit eigenen Worten.

F,S → P: *Ein Student hat in einem Fach genau einen Prüfer.*

F,R → S: *In einem Fach kann eine Platzziffer (Rang) nur einmal an einen Student vergeben werden.*

F,S → R: *Ein Student bekommt in einem Fach genau einen Rang.*

b) Beweisen Sie, dass $\{F, R\}$ ein Schlüsselkandidat der Relation EXAM ist. Gibt es noch weitere Schlüsselkandidaten?

Hinweis: X ist ein Schlüsselkandidat, wenn sich $X \longrightarrow sch(R)$ in F^+ befindet, d.h. wenn sich mit den gegebenen Funktionalabhängigkeiten und den Regeln im Skript die Funktionalabhängigkeit $X \longrightarrow FRSP$ herleiten lässt.

Zu zeigen ist, dass $X \longrightarrow FRSP$ gilt:

1. $FR \longrightarrow S$, *da als Funktionalabhängigkeit gegeben*
2. $FR \longrightarrow FR$, *Anwendung der Regel 1 mit* $X = FR$, $Y = FR$
3. $FR \longrightarrow FRS$, *Anwendung der Regel 4 mit* $X = FR$, $Y = S$, $Z = FR$
4. $FR \longrightarrow FS$, *Anwendung der Regel 6 mit* $X = FR$, $Y = FRS$, $Z = FS$
5. $FS \longrightarrow P$, *da als Funktionalabhängigkeit gegeben*
6. $FR \longrightarrow P$, *Anwendung der Regel 3 mit* $X = FR$, $Y = FS$, $Z = P$
7. $FR \longrightarrow FRSP$, *Anwendung der Regel 4 mit* $X = FR$, $Y = FRS$, $Z = P$

□

Ein weiterer Schlüsselkandidat ist $\{S, F\}$.