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 10 Abgabe: 17.6.1997

Aufgabe 1: Serialisierbarkeit, Äquivalenz

a) Gegeben ist der folgende Schedule der Transaktionen T_1, T_2, T_3, T_4 und T_5:

$$S_1 =< R_1(a), R_2(b), W_1(b), W_3(c), W_2(c), W_2(d), R_4(a), W_4(c), W_5(c), W_5(d), W_4(a) >$$

Ist der Schedule serialisierbar? Falls ja, geben Sie mindestens einen äquivalenten seriellen Schedule an. Falls nein, warum nicht?

b) Betrachten Sie den folgenden Schedule der Transaktionen T_1, T_2 und T_3:

$$S_2 =< R_1(a), R_2(b), W_1(a), R_3(b), R_2(c), W_3(d), W_2(c), W_3(e) >$$

Wie sieht der dazugehörige Abhängigkeitsgraph aus? Ist dieser Schedule serialisierbar? Falls ja, geben Sie mindestens einen äquivalenten seriellen Schedule an. Falls nein, warum nicht?

c) Gegeben ist der folgende Schedule der Transaktionen T_1, T_2, T_3:

$$S_3 =< R_1(a), R_2(a), W_1(a), R_3(b), W_3(b), R_2(b) >$$

Ist dieser Schedule serialisierbar und korrekt? Was fällt Ihnen bei diesem Schedule auf?

d) Gegeben seien die folgenden zwei Schedules der Transaktionen T_1, T_2, T_3 und T_4:

$$S_4 =< R_1(a), R_4(b), W_1(a), W_3(e), W_2(c), W_2(b), R_3(d), R_2(a), W_3(c), R_4(e), W_4(d) >$$
$$S_5 =< W_3(e), R_1(a), W_1(a), W_2(c), R_4(b), R_4(e), W_2(b), W_4(d), R_2(a), R_3(d), W_3(c) >$$

Sind die beiden Schedules äquivalent? Begründen Sie Ihre Antwort!

Aufgabe 2: Serialisierbarkeit

In der Vorlesung wurde der folgende Satz vorgestellt:

> Ein Schedule ist serialisierbar genau dann, wenn sein Abhängigkeitsgraph azyklisch ist.

Für jeden beliebigen Schedule, dessen Abhängigkeitsgraph azyklisch ist, wurde seine Zugehörigkeit zur Klasse der serialisierbaren Schedules durch topologische Sortierung des Abhängigkeitsgraphen nachgewiesen.

Beweisen Sie nun die umgekehrte Richtung des Satzes, d.h. dass jeder serialisierbare Schedule einen azyklischen Abhängigkeitsgraphen besitzt.

Aufgabe 3: Transaktionsverwaltung

Gegeben sei der folgende Schedule:

$$S =< R_1(x), W_1(x), R_2(x), W_2(x), C_2, W_1(y), C_1 >$$

C_1 und C_2 markieren den erfolgreichen Abschluss der Transaktionen T_1 und T_2 (commit).

a) Ist der Schedule serialisierbar? Begründen Sie Ihre Antwort.

b) Kann der Schedule durch das strikte 2PL-Protokoll entstanden sein? Begründen Sie Ihre Antwort.

c) Kann dieser Schedule durch das allgemeine 2PL-Protokoll entstanden sein? Begründen Sie Ihre Antwort.

d) Welches Problem gibt es, wenn anstatt der Aktion C_1 nun die Abort-Aktion A_1 ausgeführt werden soll? Wie kann das Problem verhindert werden?

Aufgabe 4: Sperrprotokoll

Betrachten Sie das folgende Sperrprotokoll:

- Für Transaktionen, die mindestens eine Änderungsoperation beinhalten, wird das strikte 2PL verwendet.
- Für Transaktionen, die nur Leseoperationen beinhalten, wird ein nichtzweiphasiges Sperrprotokoll wie folgt verwendet[1]:
  - Vor dem Zugriff auf ein Objekt muss dieses gesperrt werden.
  - Alle Sperren einer Transaktion werden jeweils bei Beendigung einer SQL–Anweisung freigegeben. (Bei der nächsten SQL–Anweisung werden wieder neue Sperren angefordert.)

a) Zeigen Sie, dass bei Verwendung dieses Protokolls nichtserialisierbare Schedules entstehen können.

b) Kann bei Verwendung des angegebenen Sperrprotokolls eine Integritätsbedingung der Datenbank verletzt werden? Falls nein, ist der „Verlust“ der Serialisierbarkeit überhaupt gravierend? Falls ja, geben Sie ein Beispiel an.

[1] Dieses Protokoll entspricht dem TRANSACTION ISOLATION LEVEL READ COMMITTED des SQL–Standards.