

---

# Transaktionsverwaltung in modernen IS

## Theoretische Übung 8

Besprechung: 17.01.2002

---

### Aufgabe 1: Rücksetzbarkeit

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

- a) Warum wird diese Forderung aufgestellt? Begründen Sie Ihre Antwort!

Um Transaktionen möglichst parallel ablaufen zu lassen, sollen Sperren so kurz wie möglich gesetzt bleiben. Wir betrachten Transaktionen mit Lese- und Schreibaktionen, die Lesesperren (S) und Schreibsperrern (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.

- b) Wann ergeben sich für Lese- oder Schreibsperrern bei den Alternativen i) oder ii) Probleme bei der Rücksetzbarkeit? Beschreiben Sie die möglichen Fehler-situationen.
- c) In welchen Fällen führt das seitenweise Wiederherstellen des Anfangszustandes einer Transaktion zu Problemen?

### Aufgabe 2: Rücksetzbarkeit/Striktheit

- a) Zeigen Sie, dass die strikten Schedules eine echte Teilmenge der kaskadenfreien Schedules sind. Finden Sie dazu ein Beispiel für einen Schedule, der kaskadenfrei, aber nicht strikt ist.
- b) Striktheit ist unabhängig von Serialisierbarkeit: Konstruieren Sie einen Schedule, der strikt ist, aber nicht konflikt-serialisierbar.

### Aufgabe 3: 2PL und Rücksetzbarkeit

- a) Konstruieren Sie einen 2PL-Schedule, der nicht zur Klasse ACA gehört. Finden Sie einen 2PL-Schedule, der nicht zu RC gehört?
- b) Finden Sie auch einen *strikten* 2PL-Schedule (S2PL), der nicht zu ACA gehört? Was schliessen Sie daraus?

## Aufgabe 4: Analyse eines Schedules

Gegeben seien die folgenden drei Transaktionen  $T^1$ ,  $T^2$ , und  $T^3$  und der unvollständige Schedule  $S$ :

$$\begin{aligned}T^1 &= \langle w_1^1(y) w_2^1(x) r_3^1(z) C^1 \rangle \\T^2 &= \langle r_1^2(z) w_2^2(x) C^2 \rangle \\T^3 &= \langle r_1^3(x) w_2^3(y) C^3 \rangle \\S &= \langle r_1^2(z) w_1^1(y) w_2^1(x) w_2^2(x) r_1^3(x) r_3^1(z) w_2^3(y) \rangle\end{aligned}$$

Fügen Sie die fehlenden Commit-Aktionen so in  $S$  ein, dass er

1. rücksetzbar, aber nicht kaskadenfrei ist.
2. kaskadenfrei ist.

## Aufgabe 5: Beispielschedules

Geben Sie für jede der folgenden Anforderungen einen Beispielschedule an oder kennzeichnen Sie die Anforderung als nicht erfüllbar.

1. Der Schedule ist konflikt-serialisierbar, aber nicht strikt.
2. Der Schedule ist strikt, aber nicht rücksetzbar.
3. Der Schedule ist durch 2PL entstanden, aber nicht strikt.