

## 7. Übungsblatt

Ausgabe: 29. November 2016 · Besprechung: Ab 05. Dezember 2016

### 1 Serialisierbarkeit

	read(x)		read(d)		read(a)
$T_1$	write(b)	$T_2$	read(a)	$T_3$	read(b)
	read(c)		write(y)		write(a)
	write(x)				write(b)
					write(x)

Die Transaktionen  $T_1, T_2$  und  $T_3$  sollen nebenläufig ausgeführt werden. Dazu wird ein Datenbanksystem eingesetzt, das sich an das 2-Phasen-Sperrprotokolls (2PL) hält. Die drei Transaktionen werden nach einer Round-Robin Strategie bearbeitet ( $T_1, T_2, T_3, T_1, \dots$ ), die jeweils einen Transaktionsschritt einer Transaktion  $T_i$  ausführt.

#### Transaktionsschritt

1. Beziehe die nächste read/write Operation  $op(X)$  von  $T_i$ .
2. Wenn  $T_i$  das Lock für  $X$  noch nicht hält:  $lock(X)$ .
3. Führe  $op(X)$  aus.
4. Tritt so früh wie möglich in die Release-Phase ein und führe für jedes Objekt  $Y$ , das von  $T_i$  nicht mehr benutzt wird  $unlock(Y)$  aus.

Wenn ein Lock für eine Transaktion nicht gewährt werden kann, wird der Transaktionsschritt beendet und die Transaktion akquiriert das Lock im nächsten regulären Transaktionsschritt, in dem das Lock frei ist.

#### Aufgaben

1. Bestimme den Schedule  $S$  mit dem das Datenbanksystem die Transaktionen ausführt.
2. Bestimme alle Konflikte in der Konfliktrelation von  $S$ .
3. Zu welchem seriellen Plan ist  $S$  konfliktäquivalent?

## 2 Klassen von Schedules

Es bezeichnen

- $S_{ser}$  die Menge aller seriellen Schedules,
- $S_{csb}$  die Menge aller konflikt-serialisierbaren Schedules,
- $S_{2PL}$  die Menge aller Schedules die durch einen 2PL Scheduler erzeugt werden können.

Zwei Mengen können in unterschiedlichen Beziehungen zueinander stehen, zum Beispiel durch Inklusion ( $S_x \subset S_y$ ), nicht-leeren Schnitt ( $S_x \cap S_y \neq \emptyset$ ) oder Disjunktheit ( $S_x \cap S_y = \emptyset$ ).

### Aufgaben

Zeige für die folgenden Paare von Mengen, in welcher Beziehung sie zueinander stehen.

1.  $S_{ser}$  und  $S_{csb}$
2.  $S_{csb}$  und  $S_{2PL}$