

1 Conflict Graphs

Betrachten Sie die folgenden Schedules $S_1 - S_3$, die in einem kompakteren Format dargestellt sind als auf den Folien. Zum Beispiel:

- $r_x(A)$ bedeutet, dass Transaktion T_x das Datenelement A liest.
- Wir lassen lokale Variablen weg, um Informationen zu repräsentieren, die gelesen oder geschrieben werden, da diese Informationen für den Conflict Graph nicht relevant sind.
- $r_x(A) \rightarrow w_x(A)$ stellt die Reihenfolge der Operationen dar, d.h., die Leseoperation in diesem Beispiel wird zuerst ausgeführt, die Schreiboperation danach.
- c_x steht für die Commit-Operation der Transaktion T_x .

1. $S_1 := w_1(A) \rightarrow w_2(B) \rightarrow r_3(B) \rightarrow c_3 \rightarrow w_1(C) \rightarrow r_2(C) \rightarrow c_1 \rightarrow c_2$
2. $S_2 := r_1(A) \rightarrow r_2(C) \rightarrow w_1(A) \rightarrow r_1(C) \rightarrow r_3(A) \rightarrow c_1 \rightarrow w_3(C) \rightarrow w_2(A) \rightarrow w_3(B) \rightarrow r_2(B) \rightarrow w_2(B) \rightarrow c_3 \rightarrow c_2$
3. $S_3 := r_1(A) \rightarrow w_1(A) \rightarrow r_3(C) \rightarrow r_2(A) \rightarrow w_3(C) \rightarrow c_3 \rightarrow w_1(B) \rightarrow w_1(C) \rightarrow r_2(B) \rightarrow w_2(B) \rightarrow w_2(C) \rightarrow c_2 \rightarrow c_1$

Für jeden dieser Schedules erstellen Sie bitte einen Conflict Graph und entscheiden Sie, ob der Schedule conflict serializable ist. Wenn er conflict serializable ist, geben Sie ein Beispiel für einen conflict equivalent serial Schedule an.

2 Recoverable und Cascadeless Schedules

Gegeben sind die folgenden Schedules S_4 and S_5 . Bitte entscheiden und erklären Sie ob jeder dieser Schedules recoverable und/oder cascadeless sind.

1. $S_4 := w_1(A) \rightarrow w_1(C) \rightarrow r_2(C) \rightarrow r_2(A) \rightarrow c_1 \rightarrow c_2$.
2. $S_5 := r_1(A) \rightarrow w_1(B) \rightarrow r_2(C) \rightarrow w_2(C) \rightarrow r_3(A) \rightarrow r_3(B) \rightarrow c_3 \rightarrow c_2 \rightarrow c_1$.

3 Gleichzeitige Ausführung von Transaktionen

Begründe die folgende Aussage: Die gleichzeitige Ausführung von Transaktionen ist wichtiger, wenn Daten von (langsamen) Festplatten abgerufen werden müssen oder wenn Transaktionen lange dauern, und weniger wichtig, wenn Daten im Speicher sind und Transaktionen sehr kurz sind.

4 Zustände von Transaktionen

Während ihrer Ausführung durchläuft eine Transaktion mehrere Zustände, bis sie schließlich abgeschlossen (commit) oder abgebrochen (abort) wird. Listen Sie alle möglichen Zustandssequenzen auf, durch die eine Transaktion laufen kann. Erklären Sie, warum jeder Zustandswechsel auftreten kann.

5 ACID Implementation

Datenbanksystem-Implementierer:innen haben den ACID-Eigenschaften deutlich mehr Aufmerksamkeit gewidmet als Dateisystem-Implementierer:innen. Warum könnte das der Fall sein?