

Modell einer Datenbank

- Datenobjekte $x, y, z \dots$
- Aktionen
 - $r(x)$ – lese Datenobjekt x (read)
 - $w(x)$ – schreibe Datenobjekt x (write)
 - c – bestätige Transaktion (commit)
 - a – breche Transaktion ab (abort)
- Transaktion ist Folge solcher Aktionen:
 $r(x); w(x); r(y); w(z); c$

Aktionen mehrerer Transaktionen

- $T_1 \triangleq r_1(x); w_1(x); r_1(y); w_1(y)$
- $T_2 \triangleq r_2(x); w_2(x)$

Der Index gibt an, welche Transaktion die Aktion durchführt.

Verschränkung von Transaktionen

Drei Abläufe S_1 , S_2 und S_3 mit den beiden Transaktionen:

- $S_1 \triangleq r_1(x); w_1(x); r_1(y); w_1(y); r_2(x); w_2(x)$
- $S_2 \triangleq r_1(x); r_2(x); w_1(x); r_1(y); w_2(x); w_1(y)$
- $S_3 \triangleq r_1(x); w_1(x); r_2(x); w_2(x); r_1(y); w_1(y)$

Serielle Abläufe

Definition

Ein Ablauf heißt **seriell**, wenn alle Schritte einer Transaktion vollständig ausgeführt werden, ehe die der nächsten Transaktion beginnen.

Beispiel

Der Ablauf S_1 ist seriell, die beiden Transaktionen folgen zeitlich aufeinander.

Konflikt zwischen Aktionen

Definition

Zwei Aktionen in einem Ablauf **stehen in Konflikt**, wenn gilt

- ① sie gehören zu unterschiedlichen Transaktionen
- ② sie greifen auf dasselbe Datenobjekt zu
- ③ mindestens einer der Aktionen ist **write**

Beispiel

- $r_1(x)$ und $w_2(x)$ stehen in Konflikt
- $r_1(x)$ und $r_2(x)$ stehen nicht in Konflikt (beide lesend)
- $w_1(y)$ und $w_2(x)$ stehen nicht in Konflikt (unterschiedliche Datenobjekte)

Konfliktäquivalenz

Definition

Zwei Abläufe heißen **konfliktäquivalent**, wenn die Reihenfolge von allen Paaren von in Konflikt stehenden Aktionen in beiden Abläufen gleich ist.

Beispiel

- S_1 und S_2 sind **nicht** konfliktäquivalent
- S_1 und S_3 sind konfliktäquivalent

Konfliktserialisierbarkeit

Definition

Ein Ablauf heißt **konfliktserialisierbar**, wenn er zu einem seriellen Ablauf konfliktäquivalent ist.

Beispiel

- S_1 ist seriell, also konfliktserialisierbar
- S_2 ist **nicht** serialisierbar
- S_3 ist serialisierbar, weil konfliktäquivalent zu S_1

Präzedenzgraph

Fragestellung

Kriterium für Serialisierbarkeit

Definition

Der **Präzedenzgraph** zu einem Ablauf S ist ein gerichteter Graph G_S mit

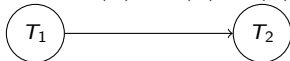
- ① den Knoten $T_1, T_2, \dots$ für jede Transaktion T_i in S
- ② den Kanten $T_i \rightarrow T_j$ falls T_i und T_j konfligierende Aktionen haben, bei denen die Aktion in T_i **vor** der in T_j in S vorkommt.

Beispiele für den Präzedenzgraphen

Beispiel

Präzedenzgraphen zu unseren Beispielabläufen:

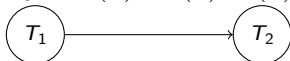
- $S_1 \triangleq r_1(x); w_1(x); r_1(y); w_1(y); r_2(x); w_2(x)$



- $S_2 \triangleq r_1(x); r_2(x); w_1(x); r_1(y); w_2(x); w_1(y)$



- $S_3 \triangleq r_1(x); w_1(x); r_2(x); w_2(x); r_1(y); w_1(y)$



Kriterium für Serialisierbarkeit

Satz (Kriterium für Serialisierbarkeit)

Ein Ablauf S ist genau dann (konflikt-)serialisierbar, wenn sein Präzedenzgraph G_S keinen Zyklus hat.

Beweis.

$\Rightarrow$) Widerspruchsbeweis

$\Leftarrow$) Induktion über die Zahl der Transaktionen

