

46116 Frühjahr 2016

Softwaretechnologie / Datenbanksysteme (nicht vertieft)

Aufgabenstellungen mit Lösungsvorschlägen



Die Bschlangaul-Sammlung

Hermine Bschlangaul and Friends

Aufgabenübersicht

Thema Nr. 1	3
Teilaufgabe Nr. 1	3
Aufgabe 5 [Transaktionen]	3
Thema Nr. 2	5
Teilaufgabe Nr. 1	5
Aufgabe 2 [Petri-Netz]	5



Die Bschlangaul-Sammlung

Hermine Bschlangaul and Friends

Eine freie Aufgabensammlung mit Lösungen von Studierenden für Studierende zur Vorbereitung auf die 1. Staatsexamensprüfungen des Lehramts Informatik in Bayern.



Diese Materialsammlung unterliegt den Bestimmungen der Creative Commons Namensnennung-Nicht kommerziell-Share Alike 4.0 International-Lizenz.

Thema Nr. 1

Teilaufgabe Nr. 1

Aufgabe 5 [Transaktionen]

- (a) Nennen Sie die vier wesentlichen Eigenschaften einer Transaktion und erläutern Sie jede Eigenschaft kurz (ein Satz pro Eigenschaft).

Lösungsvorschlag

Atomicity Eine Transaktion ist atomar, d.h. von den vorgesehenen Änderungsoperationen auf die Datenbank haben entweder alle oder keine eine Wirkung auf die Datenbank.

Consistency Eine Transaktion überführt einen korrekten (konsistenten) Datenbankzustand wieder in einen korrekten (konsistenten) Datenbankzustand.

Isolation Eine Transaktion bemerkt das Vorhandensein anderer (parallel ablaufender) Transaktionen nicht und beeinflusst auch andere Transaktionen nicht.

Durability Die durch eine erfolgreiche Transaktion vorgenommenen Änderungen sind dauerhaft (persistent).

- (b) Gegeben ist die folgende Historie (Schedule) dreier Transaktionen:

$r_1(B) \rightarrow w_1(C) \rightarrow r_3(C) \rightarrow r_1(A) \rightarrow c_1 \rightarrow r_2(C) \rightarrow r_3(C) \rightarrow r_2(C) \rightarrow w_2(B) \rightarrow c_2 \rightarrow c_3$

Zeichnen Sie den Serialisierbarkeitsgraphen zu dieser Historie und begründen Sie, warum die Historie serialisierbar ist oder nicht.

Exkurs: Historie

In Transaktionssystemen existiert ein Ausführungsplan für die parallele Ausführung mehrerer Transaktionen. Der Plan wird auch Historie genannt und gibt an, in welcher Reihenfolge die einzelnen Operationen der Transaktion ausgeführt werden. Als serialisierbar bezeichnet man eine Historie, wenn sie zum selben Ergebnis führt wie eine nacheinander (seriell) ausgeführte Historie über dieselben Transaktionen.

Lösungsvorschlag

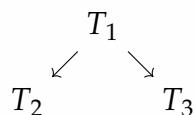
Der Algorithmus geht schrittweise durch den Ablaufplan unten und hebt die Abhängigkeiten der aktiven Transaktion zu allen anderen Transaktionen hervor. Hierfür werden in allen nachfolgenden Schritten solche Operationen gesucht, die einen Konflikt mit der aktuellen Operation hervorrufen. Konflikt-Operationen sind: read after write, write after read und write after write auf denselben Datenobjekt.

T_1	T_2	T_3
$r_1(B)$		
$w_1(C)$		
		$r_3(C)$
$r_1(A)$		
c_1		
	$r_2(C)$	
		$r_3(C)$
	$r_2(C)$	
	$w_2(B)$	
	c_2	
		c_3

Konfliktoperation

- $r_1(B) < w_2(B)$: Kante von T_1 nach T_2
- $w_1(C) < r_3(C)$: Kante von T_1 nach T_3
- $w_1(C) < r_2(C)$: Kante von T_1 nach T_2

Serialisierbarkeitsgraph



Es gibt keinen Zyklus im Graph. Er ist deshalb serialisierbar. Wenn ein Zyklus auftreten würde, dann wäre er nicht serialisierbar.

- (c) Geben Sie an, wodurch die erste und die zweite Phase des Zwei-Phasen-Sperrprotokolls jeweils charakterisiert sind (ein Satz pro Phase).

Lösungsvorschlag

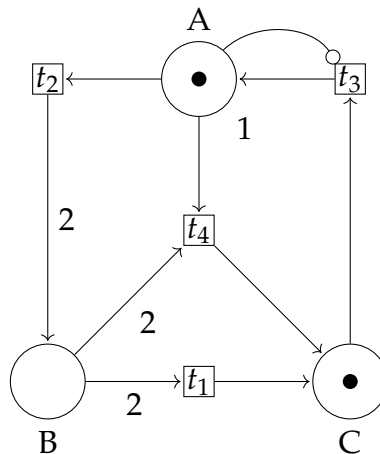
Die zwei Phasen des Protokolls bestehen aus einer *Sperrphase*, in der alle benötigten Objekte für die Transaktion gesperrt werden. In der zweiten Phase werden die *Sperren wieder freigegeben*, sodass die Objekte von anderen Transaktionen genutzt werden können.

Thema Nr. 2

Teilaufgabe Nr. 1

Aufgabe 2 [Petri-Netz]

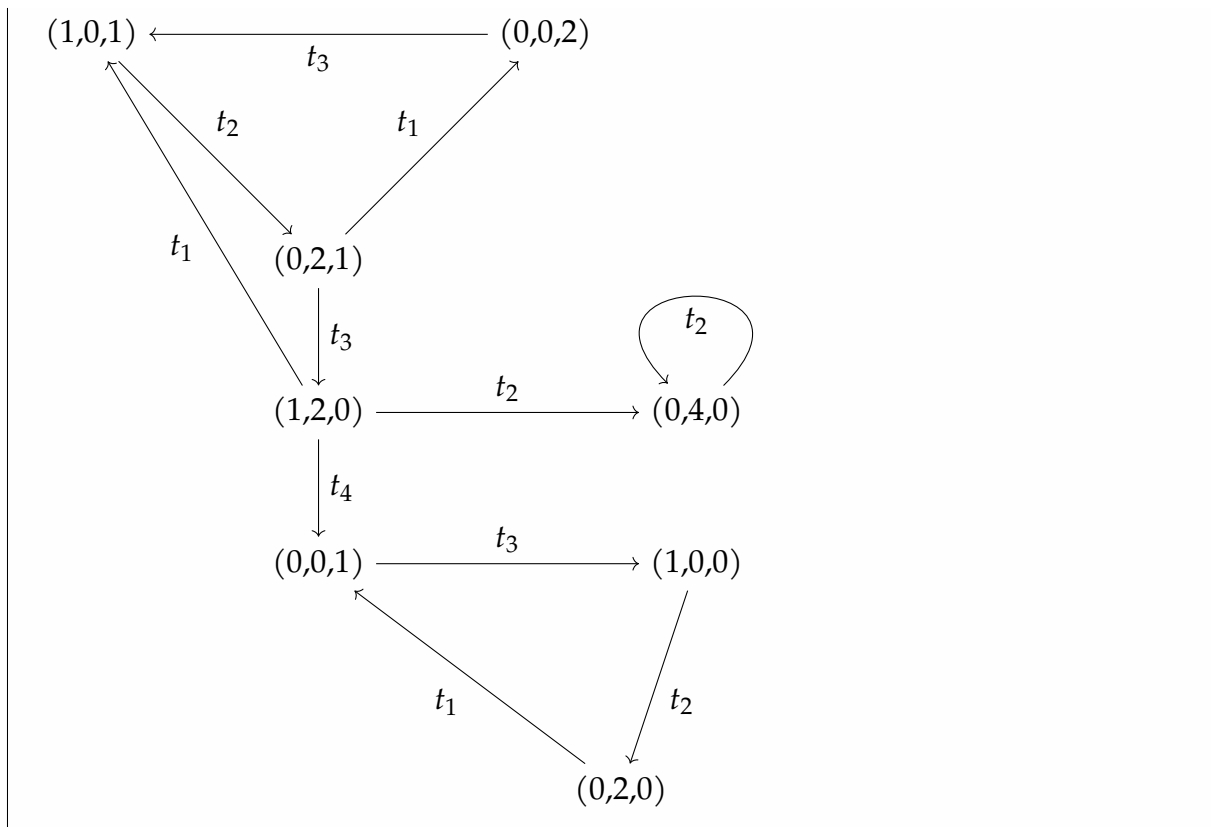
Gegeben sei das folgende Petri-Netz:



- (a) Erstellen Sie den zum Petri-Netz gehörenden Erreichbarkeitsgraphen. Die Belegungen sind jeweils in der Form $[A, B, C]$ anzugeben. Beschriften Sie auch jede Kante mit der zugehörigen Transition. Beachten Sie die auf 1 beschränkte Kapazität von Stelle A oder alternativ die Inhibitor-Kante von A zu t_3 (beides ist hier semantisch äquivalent).

Lösungsvorschlag

$(0,0,1) \rightarrow t_3 \rightarrow (1,0,0)$
 $(0,0,2) \rightarrow t_3 \rightarrow (1,0,1)$
 $(0,2,0) \rightarrow t_1 \rightarrow (0,0,1)$
 $(0,2,1) \rightarrow t_1 \rightarrow (0,0,2)$
 $(0,2,1) \rightarrow t_3 \rightarrow (1,2,0)$
 $(1,0,0) \rightarrow t_2 \rightarrow (0,2,0)$
 $(1,0,1) \rightarrow t_2 \rightarrow (0,2,1)$
 $(1,2,0) \rightarrow t_1 \rightarrow (1,0,1)$
 $(1,2,0) \rightarrow t_2 \rightarrow (0,4,0)$
 $(1,2,0) \rightarrow t_4 \rightarrow (0,0,1)$
 $(0,4,0) \rightarrow t_2 \rightarrow (0,4,0)$



- (b) Wie kann man mit Hilfe des Erreichbarkeitsgraphen feststellen, ob ein Petri-Netz lebendig ist?
- (c) Aufgrund von Transition t_4 ist das gegebene Petri-Netz nicht stark lebendig. Wie müssten die Pfeilgewichte der Transition t_4 verändert werden, damit das Petri-Netz mit der gegebenen Startmarkierung beschränkt bleibt und lebendig wird?

Lösungsvorschlag

t_4 nach C mit Gewicht 2 versehen