Transaktionsorientierte Informationssysteme SS 2000

Übung 1 Beispiellösung

Aufgabe 1:

Consider the following histories:

$s = r_1(x)r_2(y)w_1(y)r_3(z)w_3(z)r_2(x)w_2(z)w_1(x)c_1c_2c_3$
$s' = r_3(z)w_3(z)r_2(y)r_2(x)w_2(z)r_1(x)w_1(y)w_1(x)c_3c_2c_1$

Compute $H[s]$ and $H[s']$ as well as the respective RF and LRF relations. Show the step graph of these histories.

a) Lösung für s:

$H[s](x) = H_s(w_1(x)) = f_{1x}(H_s(r_1(x))) = f_{1x}(H_s(w_0(x))) = f_{1x}(f_{0x}())$
$H[s](y) = H_s(w_1(y)) = f_{1y}(H_s(r_1(x))) = f_{1y}(H_s(w_0(x))) = f_{1y}(f_{0x}())$
$H[s](z) = H_s(w_2(z)) = f_{2z}(H_s(r_2(x)), H_s(r_2(y))) = f_{2z}(H_s(w_0(x)), H_s(w_0(y))) = f_{2z}(f_{0x}(), f_{0y}())$
$RF(s) = \{(t_0, x, t_1), (t_0, y, t_2), (t_0, z, t_3), (t_0, x, t_2), (t_1, x, t_\infty), (t_1, y, t_\infty), (t_2, z, t_\infty)\}$
$LRF(s) = \{(t_0, x, t_1), (t_0, y, t_2), (t_0, x, t_2), (t_1, x, t_\infty), (t_1, y, t_\infty), (t_2, z, t_\infty)\}$

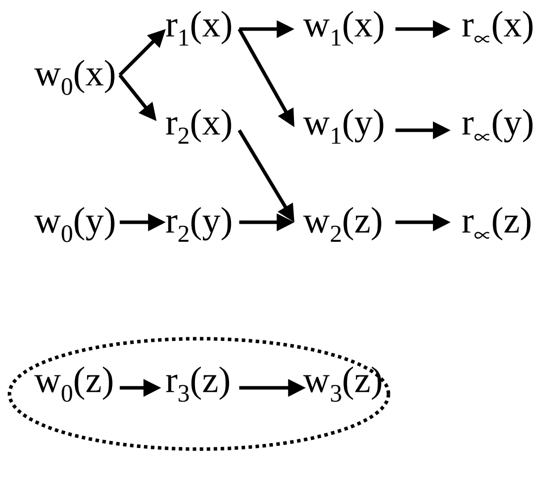

Abbildung 1: Schrittgraph des Schedules s

b) Lösung für s':
Diese Lösung ist identisch zu der für s.

Aufgabe 2:

Suppose that in a given schedule the functions corresponding to the write steps represent increments of a counter, i.e., $f(x) = x+1$. Compute the (Herbrand) semantics of the following schedules using this semantic information:

$s = r_3(z)r_1(y)w_3(z)w_1(y)r_1(x)r_2(y)w_2(y)w_1(x)r_2(x)w_2(x)c_1c_2c_3$
$s' = r_3(z)w_3(z)r_2(y)w_2(y)r_1(y)w_1(y)r_2(x)w_2(x)r_1(x)w_1(x)c_3c_2c_1$

ACHTUNG: Es gibt, so wie die Aufgabe ursprünglich gestellt war, Unklarheiten über die gewünschte Semantik. Offensichtlich ist es notwendig, dass eine Transaktion einen zu erhöhenden Zähler zunächst liest, dies ist aber hier nicht immer der Fall. Wir verwenden dann einfach die übliche Herbrand-Semantik für solche Schreibschritte. Die Musterlösung enthält die aktualisierte Aufgabenstellung, in der dieses Problem beseitigt wurde.

a) Lösung für s:
$H[s](x) = H_s(w_2(x)) = f_{2x}(H_s(r_2(x))) = f_{2x}(H_s(w_1(x))) = 1+f_{1x}(H_s(r_1(x))) = 1 + f_{1x}(H_s(w_0(x))) = 2 + f_{0x}()$
$H[s](y) = H_s(w_2(y)) = f_{2y}(H_s(r_2(y))) = f_{2y}(H_s(w_1(y))) = 1+f_1y(H_s(r_1(y))) = 1 + f_{1y}(H_s(w_0(y))) = 2 + f_{0y}()$
$H[s](z) = H_s(w_3(z)) = f_{3z}(H_s(r_3(z))) = f_{3z}(H_s(w_0(z))) = 1 + f_{0z}()$

b) Lösung für s':
$H[s'](x) = H_{s'}(w_1(x)) = f_{1x}(H_{s'}(r_1(x))) = f_{1x}(H_{s'}(w_2(x))) = 1+f_{2x}(H_{s'}(r_2(x))) = 1 + f_{2x}(H_{s'}(w_0(x))) = 2 + f_{0x}()$
$H[s'](y) = H_{s'}(w_1(y)) = f_{1y}(H_{s'}(r_1(y))) = f_{1y}(H_{s'}(w_2(y))) = 1+f_2y(H_{s'}(r_2(y))) = 1 + f_{22y}(H_{s'}(w_0(y))) = 2 + f_{0y}()$
$H[s'](z) = H_{s'}(w_3(z)) = f_{3z}(H_{s'}(r_3(z))) = f_{3z}(H_{s'}(w_0(z))) = 1 + f_{0z}()$

Aufgabe 3:

Consider $s = r_1(x)w_1(x)r_2(x)r_2(y)w_2(y)c_2w_1(y)c_1$.

Show that $s \in FSR \setminus VSR$.

a) Zunächst zeigen wir $s \in FSR$:
$RF(s) = \{(t_0, x, t_1), (t_1, x, t_2), (t_0, y, t_2), (t_1, x, t_\infty), (t_1, y, t_\infty)\}$
$LRF(s) = \{(t_0, x, t_1), (t_1, x, t_\infty), (t_1, y, t_\infty)\}$
Betrachte $s' = t_2t_1$, dann gilt:
$LRF(s') = \{(t_0, x, t_1), (t_1, x, t_\infty), (t_1, y, t_\infty)\} = LRF(s)$
also ist $s =_f s'$ und somit $s \in FSR$.

b) Wir prüfen nun die möglichen seriellen Kombinationen von t_1 und t_2, ob der entstehende Schedule view-äquivalent zu s ist.

   $s_{12} := t_1 t_2$: Nein, da $(t_1, y, t_2) \in RF(s_{12})$.

   $s_{21} := t_2 t_1$: Nein, da $(t_0, x, t_2) \in RF(s_{21})$.

   Also gilt $s \notin VSR$.

Aufgabe 4:

Consider the following history:

$s = r_1(x) r_3(x) w_3(y) w_2(y) r_4(y) c_2 w_4(x) c_4 r_5(x) c_3 w_5(z) c_5 w_1(z) c_1$

Show that $s \in FSR$.

Um zu zeigen, dass $s \in FSR$ gilt, müssen wir einen seriellen Schedule angeben, der final-state-äquivalent zu s ist. Ein solcher Schedule ist z.B. der Schedule

$$s' = t_5 t_3 t_1 t_2 t_4 = r_5(x) w_5(z) c_5 r_3(x) w_3(y) c_3 r_1(x) w_1(z) c_1 w_2(y) c_2 r_4(y) w_4(x) c_4,$$

denn:

$RF(s) = \{(t_0, x, t_3), (t_0, x, t_1), (t_3, y, t_4), (t_4, x, t_5), (t_1, z, t_\infty), (t_4, x, t_\infty), (t_3, y, t_\infty)\}$
$LRF(s) = \{(t_0, x, t_3), (t_0, x, t_1), (t_3, y, t_4), (t_1, z, t_\infty), (t_4, x, t_\infty), (t_3, y, t_\infty)\} = LRF(s')$

Aufgabe 5:

Show that $VSR = CSR$ in the absence of "blind writes", i.e., if each write step of object x is preceded by a read step on object x by the same transaction.

Sei s ein Schedule.

a) Wir müssen zunächst zeigen, dass $s \in CSR \Rightarrow s \in VSR$. Dies wurde bereits in der Vorlesung gezeigt.

b) Nun zeigen wir, dass $s \in VSR \Rightarrow s \in CSR$. Dazu nehmen wir an, s sei in VSR. Dann gibt es einen seriellen Schedule s' mit $s =_v s'$. Da s' seriell ist, ist s' trivialerweise konfliktserialisierbar.

   Um einen Widerspruch herbeizuführen, nehmen wir an, s sei nicht konfliktserialisierbar.

   Offensichtlich müssen sich dann s und s' in der Reihenfolge der Operationen mindestens eines Konfliktpärchens unterscheiden, andernfalls hätten sie den gleichen Konfliktgraphen.

   Fall 1: Es gibt eine unterschiedliche Reihenfolge von Schreiboperationen $w_i(x)$ und $w_j(x)$. o.B.d.A. sei t_i vor t_j in s'. Dann sieht der Schedule s' folgendermaßen aus:

   $s' = w_0(x) \dots r_i(x) \dots w_i(x) \dots r_{i+1}(x) \dots w_{i+1}(x) \dots r_j(x) \dots w_j(x) \dots r_\infty(x)$

In jedem view-äquivalenten Schedule muss daher gelten: $r_{i+1}(x)$ liest von $w_i(x)$. Die Reihenfolge von Operationen innerhalb einer Transaktion bleibt erhalten, also muss in jedem zu s' view-äquivalenten Schedule $w_i(x) < w_{i+1}(x)$ gelten.

Das gilt für jeweils zwei direkt aufeinander folgende Schreibzugriffe. Wegen der Transitivität von $<$ ist eine unterschiedliche Reihenfolge zweier Schreibzugriffe in s und s' nicht möglich. $\Rightarrow$ Widerspruch.

Fall 2: Es gibt eine unterschiedliche Reihenfolge einer Leseoperation $r_i(x)$ und $w_j(x)$. Da s und s' view-äquivalent sind, darf sich die Liest-von-Relation nicht ändern.

Die Operation $r_i(x)$ lese x von $w_k(x)$ im Schedule s. Dann kann nicht $j = k$ gelten, da diese Liest-von-Beziehung sonst in s' nicht gelten kann. Also muss $j \neq k$ gelten.

Fall 2a: Wir nehmen an, Schedule s sehe folgendermaßen aus:

$\ldots w_k(x) \ldots r_i(x) \ldots w_j(x) \ldots$

Soll die Reihenfolge von $r_i(x)$ und $w_j(x)$ in s' umgedreht werden, aber die Liest-von-Relation erhalten bleiben, so muss s' folgendermassen aussehen:

$\ldots \ldots w_j(x) w_k(x) \ldots r_i(x) \ldots$

Dann hat sich aber die Reihenfolge von $w_j(x)$ und $w_k(x)$ geändert, und wir können Fall 1 anwenden.

Fall 2b: Die andere Möglichkeit für Schedule s sieht folgendermaßen aus:

$\ldots \ldots w_j(x) w_k(x) \ldots r_i(x) \ldots$

Dieser Fall läßt sich analog zu Fall 2a zum Widerspruch führen.

Aufgabe 6:

Let $s = r_1(z)r_3(x)r_2(z)w_1(z)w_1(y)c_1w_2(y)w_2(u)c_2w_3(y)c_3$.

Show that $s \in VSR \setminus CSR$.

Um zu zeigen, dass $s \notin CSR$ gilt, betrachten wir den Konfliktgraphen von s, der einen Zyklus hat:

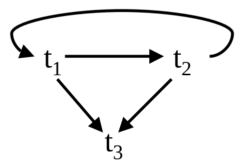

Abbildung 2: Konfliktgraph des Schedules s

Um zu zeigen, dass $s \in VSR$ gilt, müssen wir einen seriellen Schedule angeben, der view-äquivalent zu s ist. Ein solcher Schedule ist z.B. der Schedule

$$s' = t_2t_1t_3 = r_2(z)w_2(y)w_2(u)c_2r_1(z)w_1(z)w_1(y)c_1r_3(x)w_3(y)c_3,$$

denn es gilt:

$RF(s) = \{(t_0, z, t_1), (t_0, x, t_3), (t_0, z, t_2), (t_0, x, t_\infty), (t_3, y, t_\infty), (t_1, z, t_\infty), (t_2, u, t_\infty)\} = RF(s')$

Aufgabe 7:

Consider $s = r_1(x)w)1(z)w_2(z)w_1(y)c_1r_3(y)w_2(z)c_2w_3(x)w_3(y)c_3$. Using the conflict graph of s as an argument, show that $s \in CSR$. Does $s \in OCSR$ also hold?

Da der Konfliktgraph des Schedules azyklisch ist, gilt $s \in CSR$.

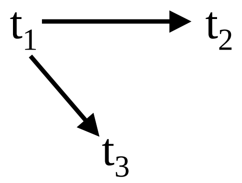

Abbildung 3: Konfliktgraphgraph des Schedules s

Durch topologische Sortierung des Konfliktgraphen ergibt sich der äquivalente serielle Schedule $s' = t_1t_2t_3$. In s sind lediglich die Transaktionen t_1 und t_3 vollständig geordnet, ihre Reihenfolgen bleibt in s' erhalten, also gilt $s \in OCSR$. Außerdem gilt $s \in COCSR$, da die Reihenfolge der Commitoperationen in s mit der Reihenfolge der Transaktionen in s' übereinstimmt.