total: 23/25

Distributed systems I
Winter Term 2019/20

G2T1 – Assignment 4 (theoretical part)

Felix Bühler 2973410 | Clemens Lieb 3130838 | Steffen Wonner 2862123 | Fabian Bühler 2953320

December 15, 2019

11

1 - Global State

a)

C_1: e_{11} e_{21} e_{31} e_{12} e_{22}
1
C_2: e_{11} e_{21} e_{31} e_{12} e_{22} e_{41} are they consistent ? justification missing

b)

(i) The inequation gives true for all states that can follow the first one given, independent of
2 how many states lie between them. To make sure that the other state lies on level $L+1$ you can either check if any other state lies between them or check the timestamps. The timestamp of the process with that last event has to be 1 higher than in the other state.

(ii) $S_{30} \rightarrow S_{31} = \text{true}$
2 $S_{30} \rightarrow S_{32} = \text{true}$
$S_{30} \rightarrow S_{42} = \text{true}$? justification?

$S_{31} \rightarrow S_{32} = \text{true}$
$S_{32} \rightarrow S_{42} = \text{true}$ not asked

So S_{32} and S_{42} are not on level $L+1$, because S_{31} lies between them and S_{30}.
You either have to check the timestamps (not given) or check all other states if they lie between S_{30} and S_{31}. None of them do, so S_{31} is on level $L+1$ of S_{30}.
it would be way better if you argument with the difference for each event instead of what lies where assuming you meant the same thing as mentioned in part i

2
c)

See Figure 1

d)

- $\phi_1 = (x_2 - x_1) = 5$
  $\Rightarrow (\neg\phi_1)$ marked in blue
  white?

Felix Bühler
Clemens Lieb
Steffen Wonner
Fabian Bühler

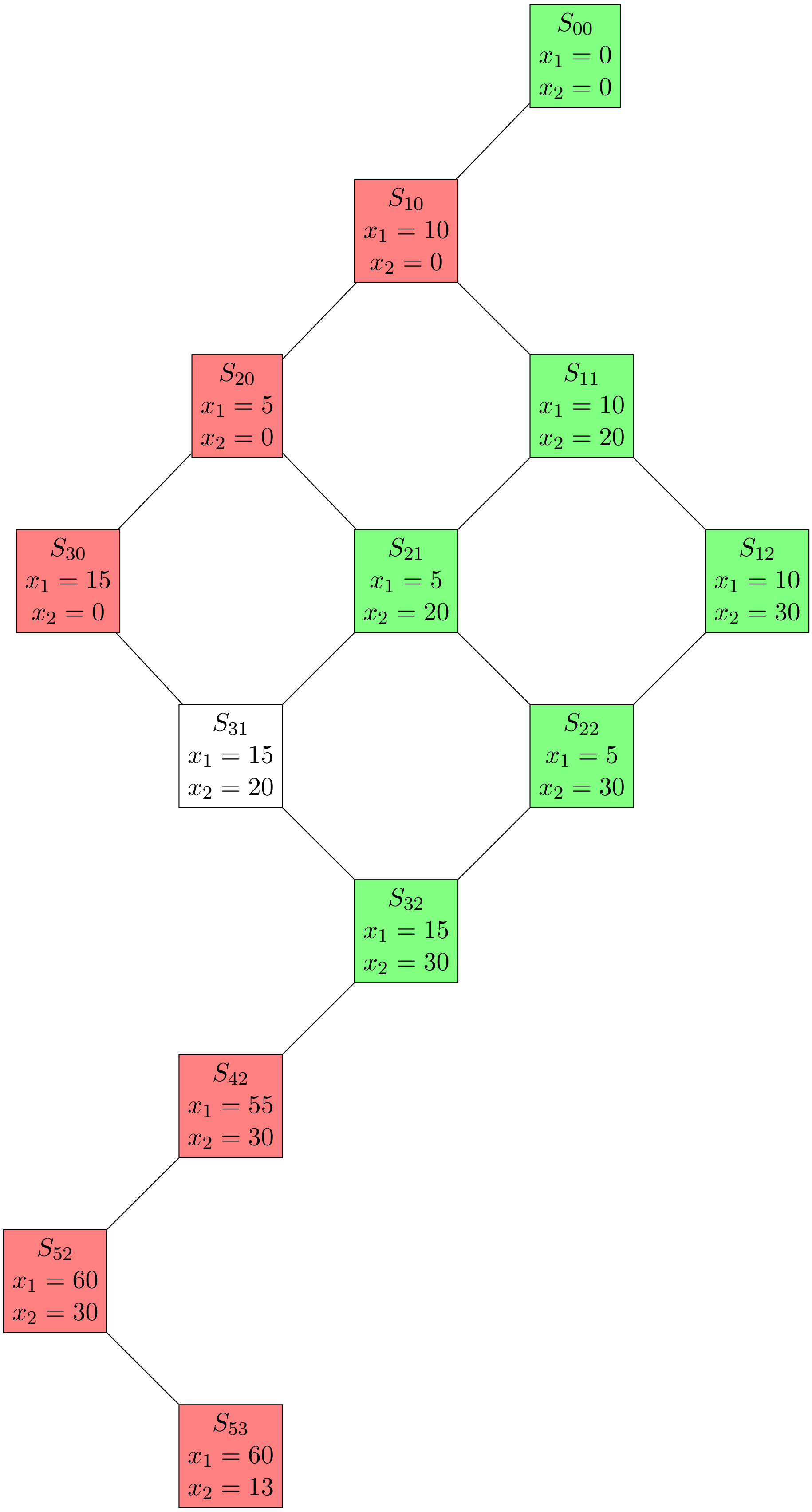

Figure 1: solution for 1.c)

Felix Bühler
Clemens Lieb
Steffen Wonner
Fabian Bühler

- $\phi_2 = (x_2 - x_1) \leq 5$ and $(x_2 - x_1) > 0$
  $\Rightarrow (\phi_2)$ marked in green so in S00 x2-x1>0?!
- if they fulfill $(\neg\phi_1)$ and (ϕ_2) marked in red

2 1. safety condition $\neg\phi_1$ is not fulfilled in the state S_{31} and thereby not fulfilled.

2 2. liveness condition ϕ_2 is fulfilled in the end and thereby fulfilled.
(therefore in any linearization)

6

2 - Transaction Processing

2 a)

The conflicting operation pairs are:

$$\{(w_1[u], w_3[u]), (r_1[y], w_2[y]), (r_3[y], w_2[y]), (r_2[x], w_3[x])\}$$

b)

4

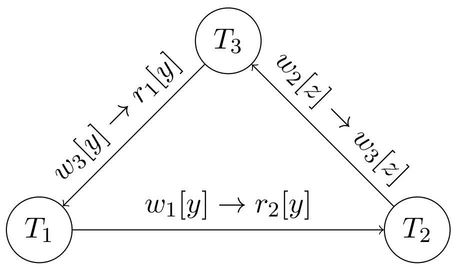

Figure 2: Serialization Graph for History H_1

It's clear that the serialization graph shown in Figure 2 contains a cycle, as such H_1 is not serializable.

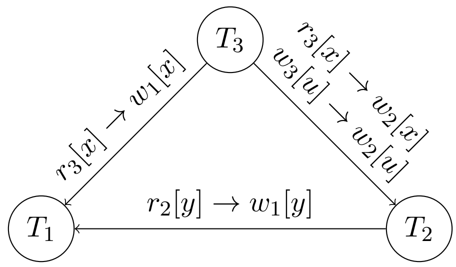

Figure 3: Serialization Graph for History H_2

The serialization graph shown in Figure 3 does not contain a cycle, which would imply that H_2 is serializable.
Unfortunately H_2 is not a correct history as it does not specify the order or the conflicting operations $w_1[x], w_2[x]$.

Felix Bühler
Clemens Lieb
Steffen Wonner
Fabian Bühler

Distributed systems I
Winter Term 2019/20 G2T1 – Assignment 4 (theoretical part)

6

3 - Two-Phase Locking

H_1: could be generated by 2-phase-locking. The shrinking phase for each transaction begins right after the last operation before the commit.

Execution split for each resource:

y: $rL_1 \to r_1 \to wL_1 \to w_1 \to wU_1 \to rL_2 \to r_2 \to c_1 \to rU_2 \to c_2 \to c_3$

x: only read operations $\Rightarrow$ no conflicting locks possible

z: only read operations $\Rightarrow$ no conflicting locks possible

w: $c_1 \to wL_2 \to w_2 \to wU_2 \to c_2 \to wL_3 \to w_3 \to wU_3 \to c_3$

H_2: is not possible with 2-phase-locking!

Conflicting excerpt: $r_1[x] \to w_2[x] \to r_1[y]$

Either T_2 cannot aquire the write lock for $w_2[x]$ as T_1 still holds the read lock or T_1 is already in the lock shrinking phase when it wants to execute $r_1[y]$ (this is the first operation of T_1on y) but both transactions commit.

H_3: is not possible with 2-phase-locking! is it even serializable?

Conflicting excerpt: $r_1[y] \to w_3[y] \to w_1[z]$

Analogous situation to H_2. Shrinking for T_1 can only happen after $w_1[z]$ but $r_1[y]$ and $w_3[y]$ are conflicting locks.