Theory Exercise 9

Tarek Auel, Markus Schanz

Task 1: ‘Snapshot’-Algorithm of Chandy and Lamport

a)

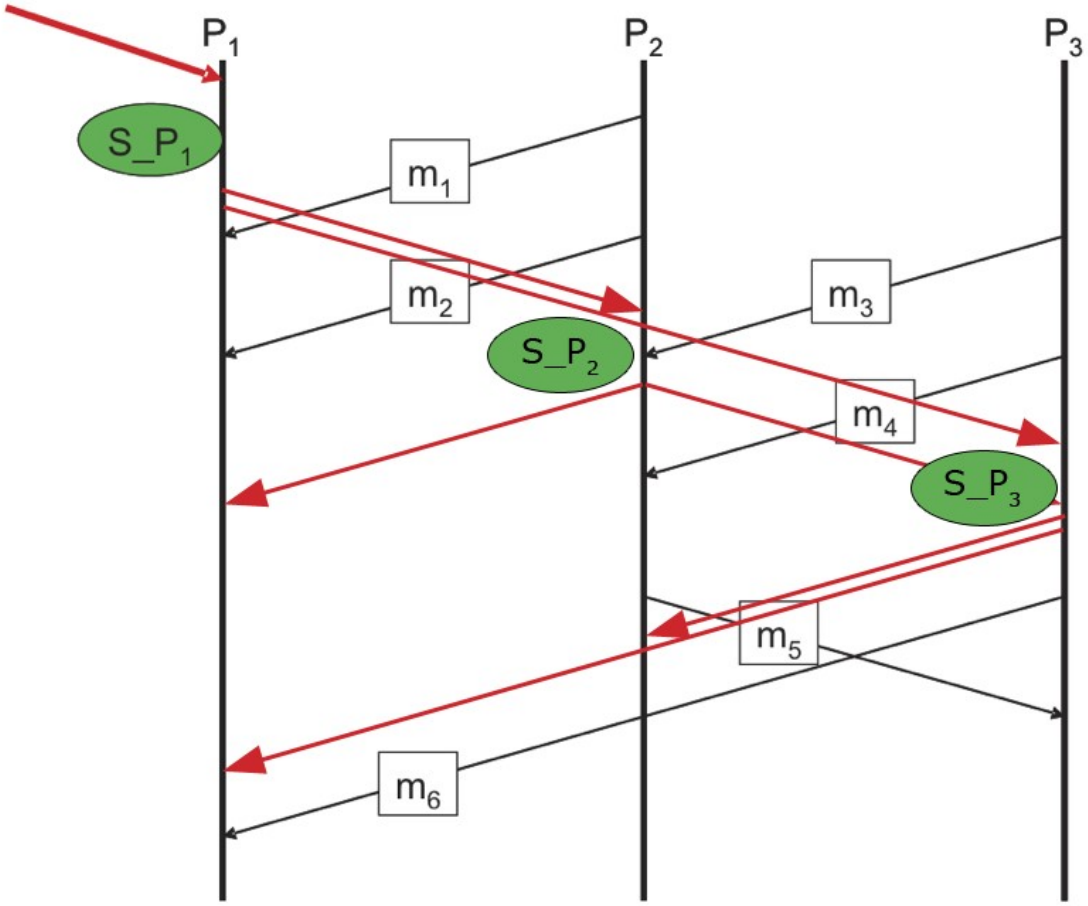
Illustration 1.I is correct, as P_1 forwards the marker message to P_2 after recording its state and before sending any other message.

Example for figure 1.I:
P_1 receives the marker message and saves its local state immediately. It forwards the marker message to process P_2, followed by a regular message. P_2 receives the marker message first and creates a local snapshot. It then receives the regular message which is not part of the local snapshot, as it is received by P_1 after the receiving the marker message. For both processes the message is not part of the snapshot, so the snapshot itself is consistent.

Example for figure 1.II:
P_1 receives the marker message and saves its local state immediately. It then sends a regular message to P_2, followed by the marker message. The regular message is not part of the local snapshot, as it happens after the receiving of the marker message. Now P_2 receives the regular message first, followed by the marker message. It creates a local snapshot, consisting of all previous events, including the receiving of the regular message from P_1. After the algorithm finishes, the global snapshot that is created out of the local snapshots is inconsistent because there exists an receiving event for a message without a corresponding sending event.

b)

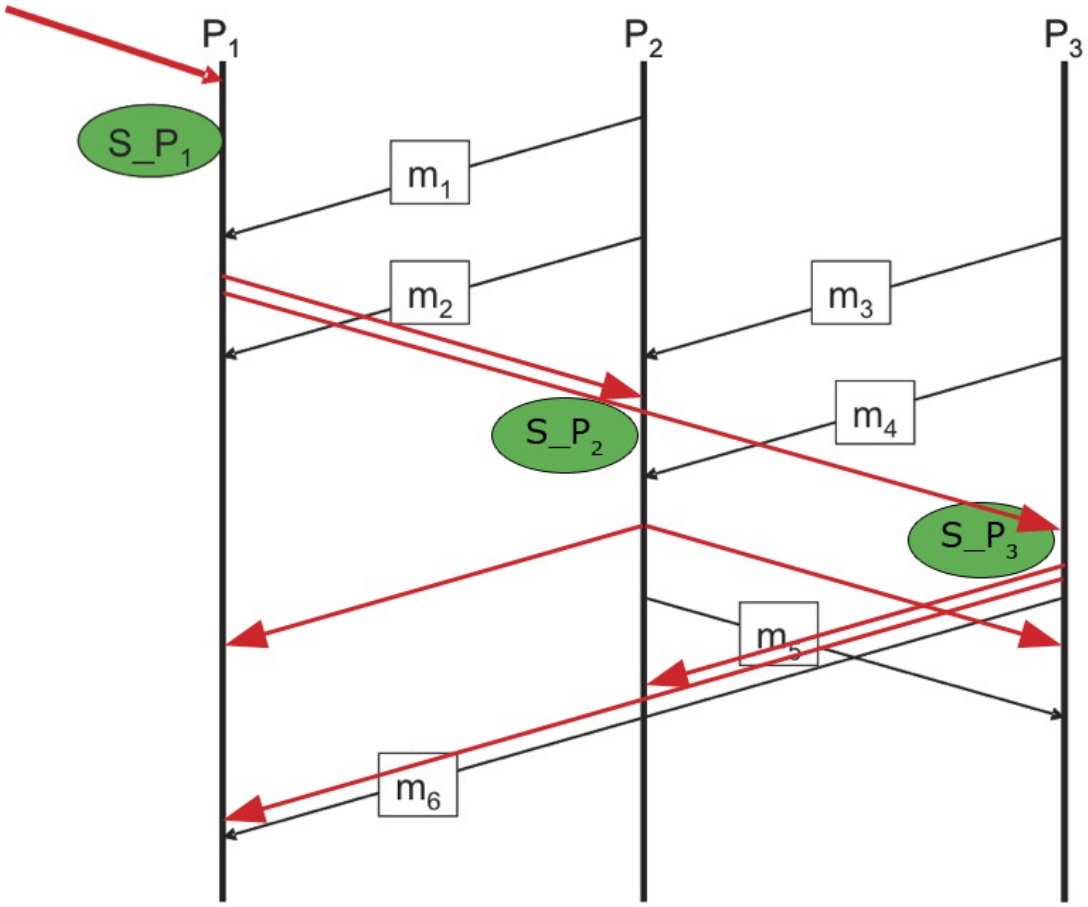
Variant 1:



States:

S_P_1: <>	$C(P_2, P_1)$: <rec_m1, rec_m2>	$C(P_3, P_1)$: <>
S_P_2: <send_m_1, send_m_2>	$C(P_1, P_2)$: <>	$C(P_3, P_2)$: <rec_m_3, rec_m_4>
S_P_3: <send_m_3, send_m_4>	$C(P_1, P_3)$: <>	$C(P_2, P_3)$: <>

Variant 2:



States:

S_P_1: <>	$C(P_2, P_1)$: <rec_m_1, rec_m_2>	$C(P_3, P_1)$: <>
S_P_2: <$send\_m_1$, $send\_m_2$, rec_m_3>	$C(P_1, P_2)$: <>	$C(P_3, P_2)$: <rec_m_4>
S_P_3: <$send\_m_3$, $send\_m_4$>	$C(P_1, P_3)$: <>	$C(P_2, P_3)$: <>

Task 2: Snapshot vs. Actual Program Flow

a)

Sys' = s_{m1}, s_{m2}, r_{m1}, r_{m2}
pre-snap: s_{m1}
post-snap: s_{m2}, r_{m1}, r_{m2}

b)