Übung 1

- Logische Uhren

Die Relation ***a*** ereignet sich kausal vor ***b*** wird kurz als **a**$\rightarrow$ **b** notiert

Von zwei Ereignissen ***a*** und ***b*** sind logische Zeitstempel nach Lamport, ***C(a)*** und ***C(b)***, bekannt, und es gilt ***C(a) < C(b)***.

Kann man daraus Informationen über die Korrektheit von folgenden Aussagen gewinnen

(1) $a \rightarrow b$

(2) $b \rightarrow a$

(3) ***a*** ereignet sich in der Realzeit vor ***b***

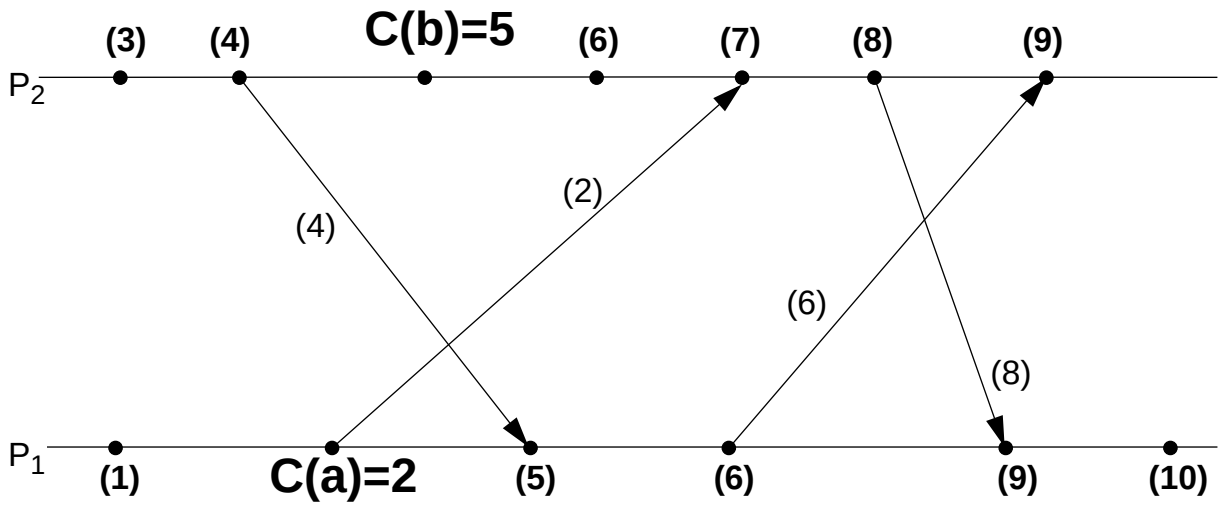
(4) ***b*** ereignet sich in der Realzeit vor ***a***

(5) ***a*** und ***b*** sind nebenläufig (d. h. es gilt weder $b \rightarrow a$ noch $a \rightarrow b$)

Übung 1

(1) ***a*** $\rightarrow$ ***b***

(2) ***b*** $\rightarrow$ ***a***

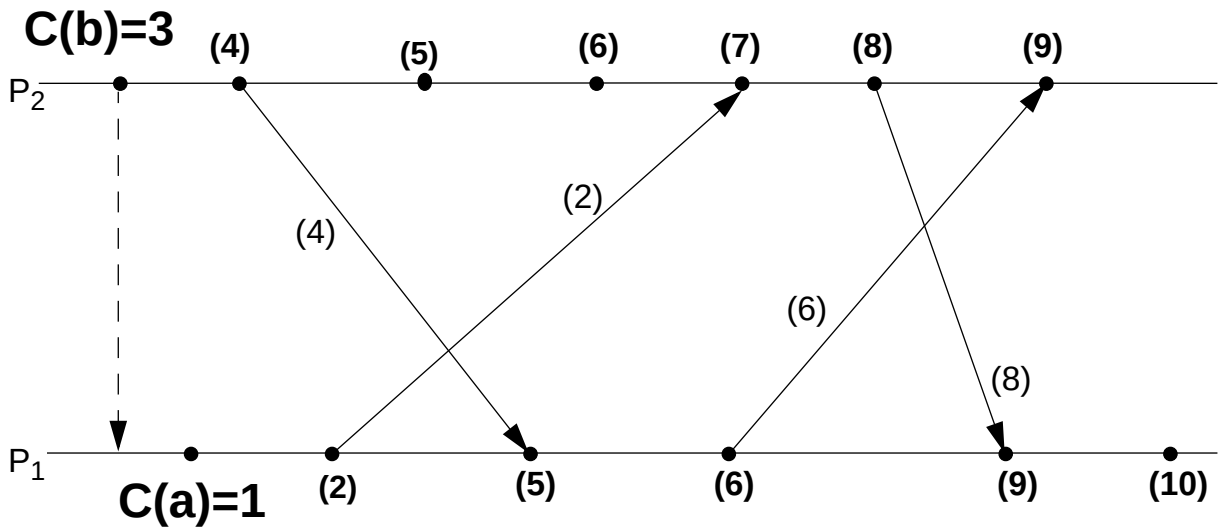


Übung 1

(4) ***a*** ereignet sich in der Realzeit vor ***b***

(5) ***b*** ereignet sich in der Realzeit vor ***a***

(6) ***a*** und ***b*** sind nebenläufig (d. h. es gilt weder ***b*** $\rightarrow$ ***a*** noch ***a***$\rightarrow$ ***b***)



Übung 1

- Vektoruhren

Was kann man über die Korrektheit von folgenden Aussagen sagen wenn von **a** und **b** zwei Vektor-Zeitstempel bekannt sind und für diese **V(a) < V(b)** gilt?

(1) $\boldsymbol{a} \rightarrow \boldsymbol{b}$

(2) $\boldsymbol{b} \rightarrow \boldsymbol{a}$

(3) ***a*** ereignet sich in der Realzeit vor ***b***

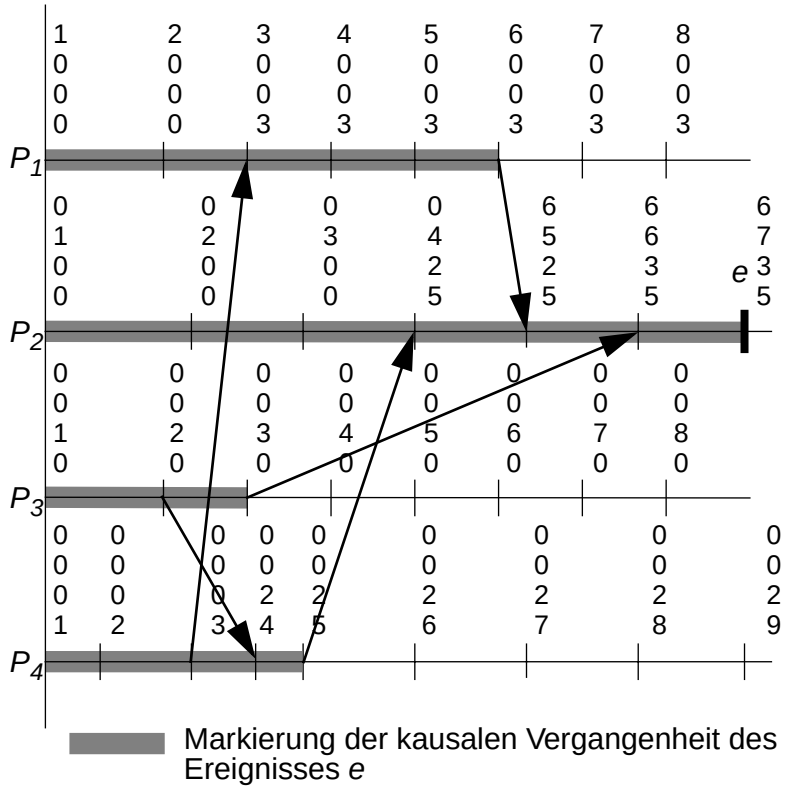
(4) ***b*** ereignet sich in der Realzeit vor ***a***

(5) ***a*** und ***b*** sind nebenläufig (d. h. es gilt weder $\boldsymbol{b} \rightarrow \boldsymbol{a}$ noch $\boldsymbol{a} \rightarrow \boldsymbol{b}$)

Übung 1

(1) $\boldsymbol{a} \rightarrow \boldsymbol{b}$

(2) $\boldsymbol{b} \rightarrow \boldsymbol{a}$

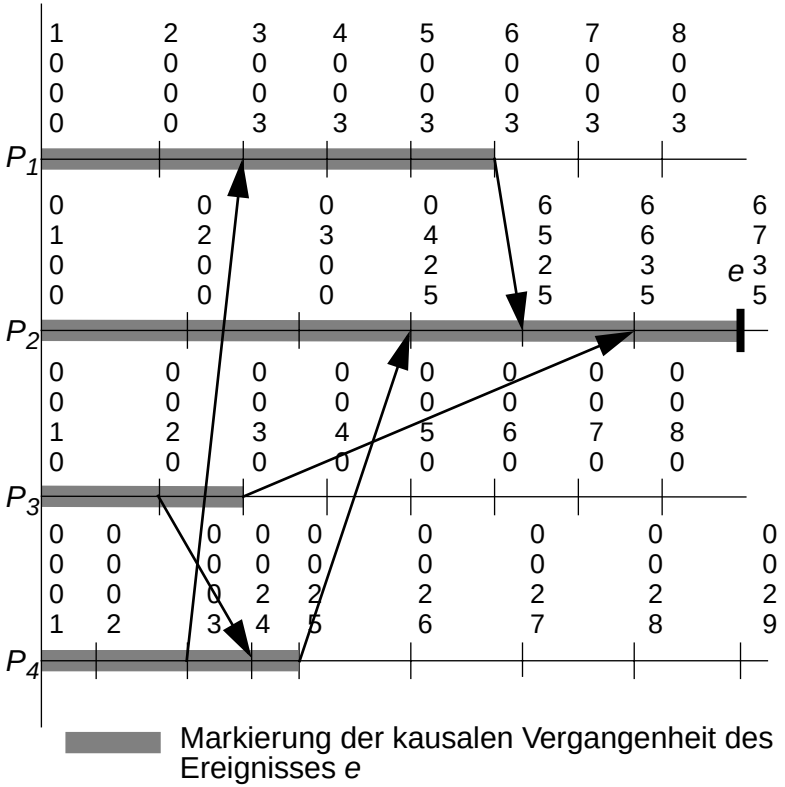


Übung 1

(4) ***a*** ereignet sich in der Realzeit vor ***b***

(5) ***b*** ereignet sich in der Realzeit vor ***a***

(6) ***a*** und ***b*** sind nebenläufig (d. h. es gilt weder ***b*** $\rightarrow$ ***a*** noch ***a***$\rightarrow$ ***b***)



Übung 1

- Logische Uhren und Fehlertoleranz

Wie wirkt sich der Ausfall eines Rechners bei Lamports logischen Uhren auf die garantierten Eigenschaften des Algorithmus aus? (D.h. können Ordnungseigenschaften bei einem Ausfall eines Rechners verletzt werden?)

- Es gibt keine Auswirkungen auf den laufenden Algorithmus
- Was passiert wenn der ausgefallene Rechner wieder aktiv wird?

Übung 1

- Totale Ordnung mit Lamports Uhren

  In der Vorlesung wurde ein Weg gezeigt, um aus Lamports Uhren eine totale Ordnung zu erzeugen. Dabei werden Ereignisse von Knoten mit kleiner ID immer bevorzugt. Lässt sich das Verfahren ohne großen Aufwand so modifizieren, dass dieser Nachteil vermieden wird?

Übung 1

Totale Ordnung mit Lamports Uhren

Lsg1:

$((C_i, i) < (C_k,k)$ oder $(C_i = C_k$ und $C_i+i \bmod N < C_k+k \bmod N))$

$\Leftrightarrow C_i < C_k$

Lsg2:

$((C_i, i) < (C_k,k)$ oder $(C_i = C_k$ und $C_k \bmod 2 = 0$ und $i<k)) \Leftrightarrow C_i < C_k$

Übung 1

- Kausale Ordnung von Nachrichten

  In einem verteilten System sei eine logische Uhr nach Lamport implementiert, die zu jedem Ereignis (lokal, senden, empfangen) einen Zeitstempel liefert. Wie bekannt gelte

  $a \rightarrow b \Rightarrow C(a) < C(b)$.

  Wie lässt sich in diesem System erreichen, dass alle Nachrichten gemäß einer kausalen Ordnung verarbeitet werden, wenn das System selbst nur einen einfachen Multicast mit FIFO-Semantik bereitstellt?

Übung 1

- Kausale Ordnung von Nachrichten

  Lsg: Ein Knoten kann eine Nachricht erst dann verarbeiten wenn er sicher ist das es **keine** Nachricht mehr mit einem älteren Zeitstempel gibt.

  Beispiel: Man wartet auf eine Nachricht von jedem Rechner und der Empfang einer Nachricht wir bestätigt durch eine weitere Nachricht an alle.

Übung 1

- Zusammenhang Lamports Uhren und Vektorzeit

  Wie ermittle ich aus einem Vektoruhr-Zeitstempel einen einfachen Skalaren Zeitstempel, der alle Eigenschaften von Lamports Uhren erfällt?

  Lsg: Alle Vektorkomponenten aufaddieren.

  Durch lokale Aktionen erhöht sich ein Zähler und damit die Gesamtsumme.

  Bei jedem Empfang einer Nachricht wir komponentenweise das Maximum gebildet. Durch das aufaddieren ergibt sich eine neue Gesamtsumme die größer als das Maximum der alten Summe ist.