

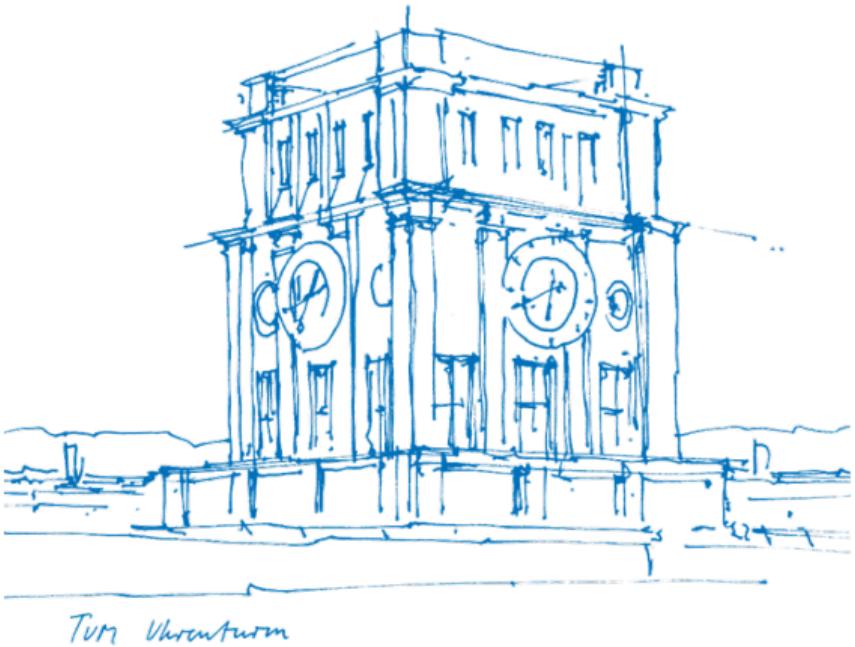
Grundlagen: Betriebssysteme und Systemsoftware

Tutorübung

Mario Delic

Lehrstuhl für Connected Mobility
School of Computation, Information and Technology
Technische Universität München

Übungswoche 8



Aufgabe 2

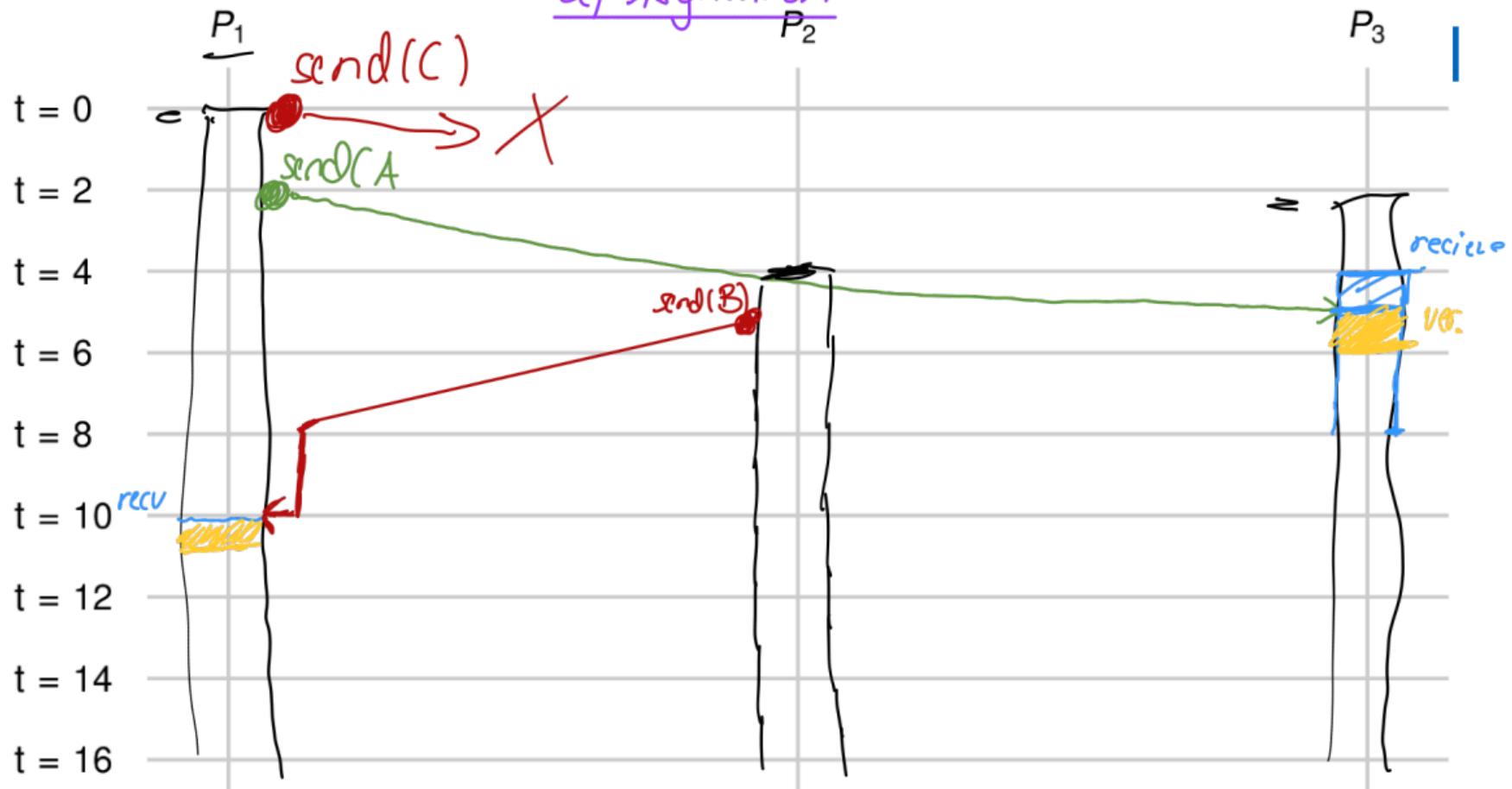
Modellierung von Kommunikation

- Eine Nachricht kann nur abgeschickt werden, wenn der Empfänger existiert.
- Nachrichtenlaufzeiten = 3 ZE (Mindestzeit zw. erfolgreichem send und receive).
- Bei erfolgreichem receive wird 1 ZE verarbeitet.
- Sonstige Operationen sind overheadfrei (send, Verarbeitung von Bestätigungen...).

a) **Asynchron:**

- send wartet nicht auf eine Bestätigungsmessage vom Empfänger.
- Der Sender ist nach erfolgreichem send 'fertig'.
- Der Empfänger ist nach Verarbeiten der Nachricht 'fertig'.

a) Asynchron



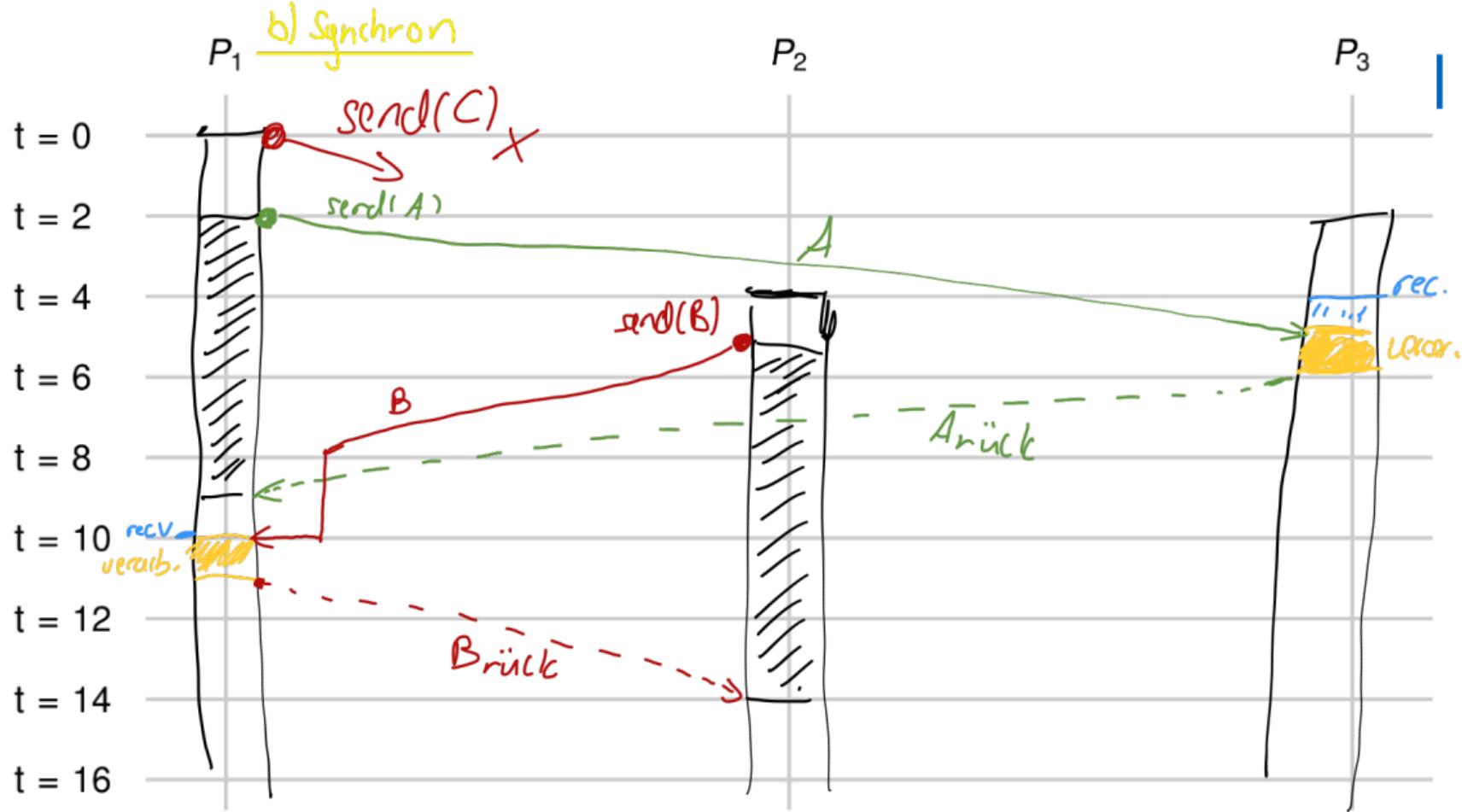
Aufgabe 2

Modellierung von Kommunikation

- Eine Nachricht kann nur abgeschickt werden, wenn der Empfänger existiert.
- Nachrichtenlaufzeiten = 3 ZE (Mindestzeit zw. erfolgreichem send und receive).
- Bei erfolgreichem receive wird 1 ZE verarbeitet.
- Sonstige Operationen sind overheadfrei (send, Verarbeitung von Bestätigungen...).

b) **Synchron:**

- send wartet auf eine Bestätigungsmessage vom Empfänger.
- send blockiert nach erfolgreichem senden bis zum Erhalt dieser Nachricht.
- Der Empfänger sendet nach der Verarbeitung die Bestätigungsmessage.
- Der Empfänger ist nach Senden dieser Nachricht 'fertig'.
- Der Sender ist nach Erhalt der Bestätigungsmessage 'fertig'.



Erreichbarkeitsgraph

Erinnerung

- Knoten (Zustände): $(M(s_0), \dots, M(s_n))$ für alle n Stellen des Petrinetzes (M: engl. **Marking** = Belegung der Stelle)
- Übergangsfunktion: Kanten $t_0, t_1, t_2 \dots$ welche Transitionen darstellen (nur die, die tatsächlich schalten können!).
- Ein Blatt/Fangzustand ist ein Zustand ohne ausgehende Kanten ist ein Blatt/Fangzustand, und somit **verklemmt** das Petrinetz dort.
- Ist der Erreichbarkeitsgraph verklemmungsfrei, lebendig oder fair, so auch das zugrundeliegende (standard) Petrinetz.

Erreichbarkeitsgraph

Konstruktion

⇒ Konstruktion $P \rightarrow E$

Siehe letzte Woche (Blatt 7).

⇐ Konstruktion $P \leftarrow E$

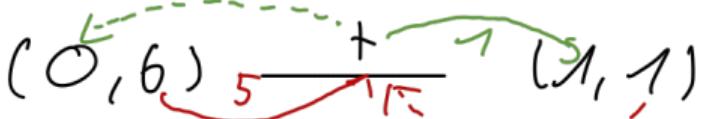
Alle Stellen notieren.

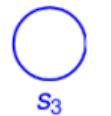
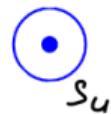
Wiederhole für jede Transition:

Transition t_x und Zustände Z_{vor} und Z_{nach} betrachten.

- Falls Belegung der Stelle s_i im Zustand Z_{vor} größer ist als im Zustand Z_{nach} , also:
Falls $Z_{\text{vor}}(M(s_i)) > Z_{\text{nach}}(M(s_i))$, dann Kante von Stelle s_i zu t_x .
- Falls $Z_{\text{nach}}(M(s_i)) > Z_{\text{vor}}(M(s_i))$, dann Kante nach Stelle s_i aus t_x .
- Falls $Z_{\text{vor}}(M(s_i)) = Z_{\text{nach}}(M(s_i))$, dann keine Kante zwischen t_x und s_i

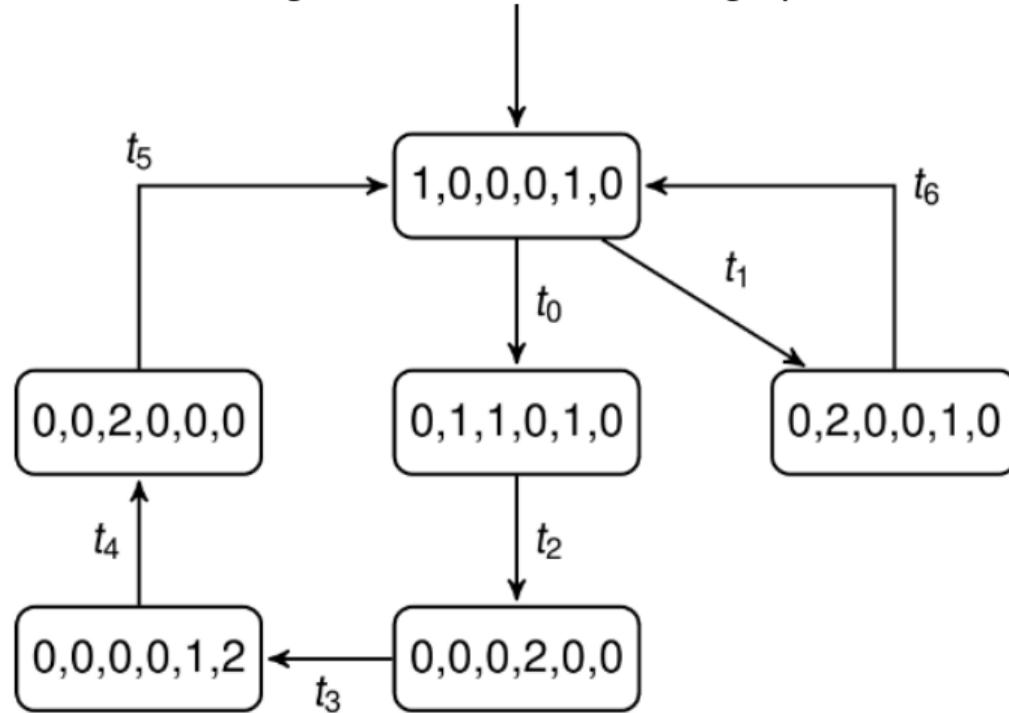
Das Gewicht der Kanten ist die Differenz der Belegung der jeweiligen Zustände.

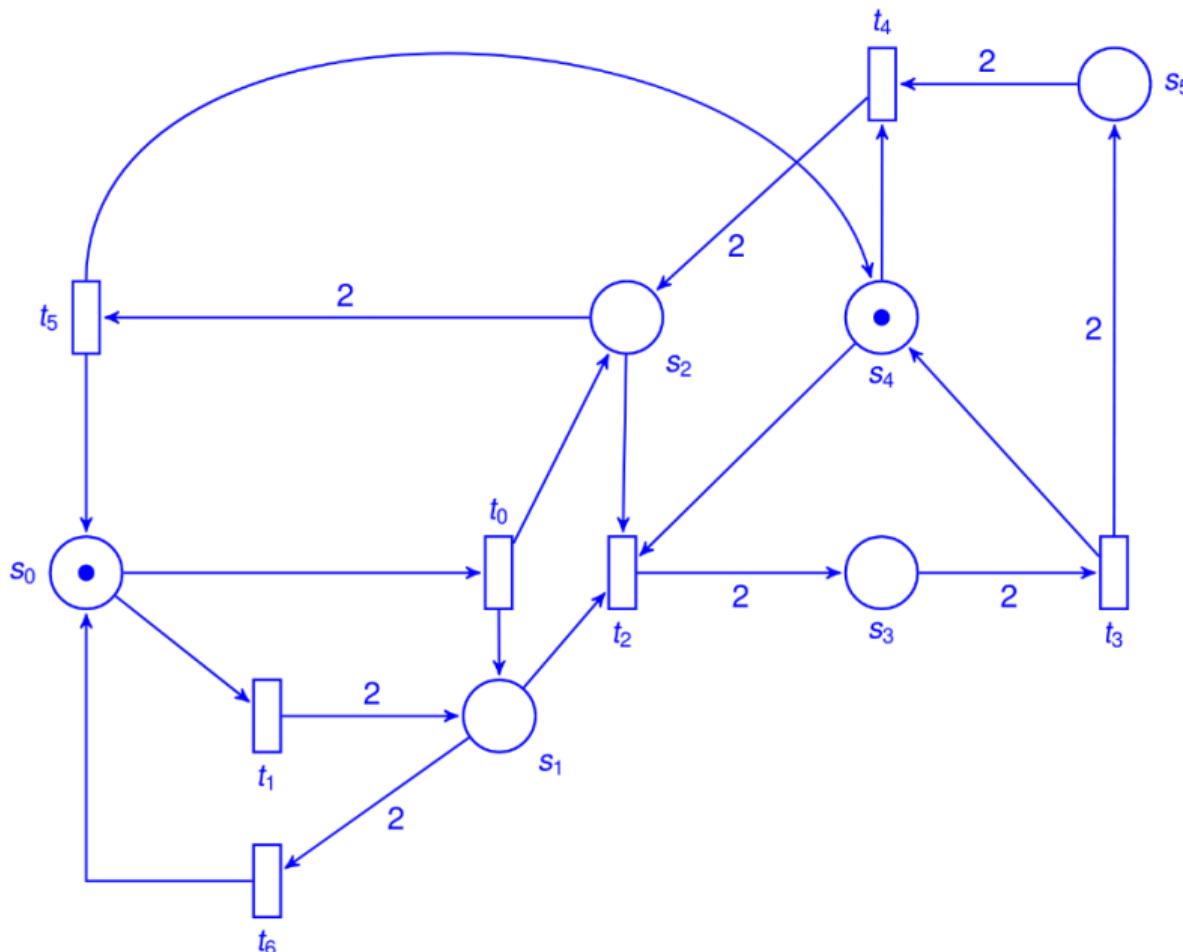




Aufgabe 3

a) Geben Sie ein Petrinetz zu folgendem Erreichbarkeitsgraphen an.





Aufgabe 3

- b) Argumentieren Sie anhand des Erreichbarkeitsgraphen, welche der (drei) Ihnen bekannten Eigenschaften das dazugehörige Petrinetz besitzt.

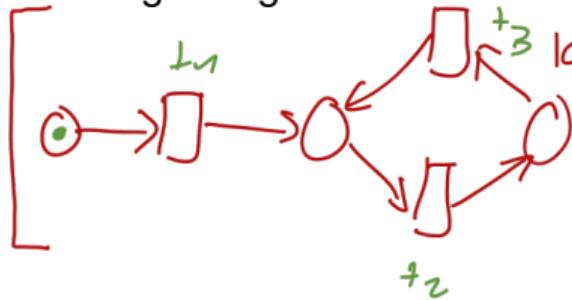
Aufgabe 3

- b) Argumentieren Sie anhand des Erreichbarkeitsgraphen, welche der (drei) Ihnen bekannten Eigenschaften das dazugehörige Petrinetz besitzt.

Verklemmungsfrei:

Lebendig:

Fair:



Klassisches Bsp. für verklemmungsfreies, aber nicht lebendiges Netz

Aufgabe 3

- b) Argumentieren Sie anhand des Erreichbarkeitsgraphen, welche der (drei) Ihnen bekannten Eigenschaften das dazugehörige Petrinetz besitzt.

Verklemmungsfrei: 

Lebendig: 

Fair: 

- c) Ist das zu einem Erreichbarkeitsgraphen zugehörige Petrinetz eindeutig?

Aufgabe 3

- b) Argumentieren Sie anhand des Erreichbarkeitsgraphen, welche der (drei) Ihnen bekannten Eigenschaften das dazugehörige Petrinetz besitzt.

Verklemmungsfrei: 

Lebendig: 

Fair: 

- c) Ist das zu einem Erreichbarkeitsgraphen zugehörige Petrinetz eindeutig?

Transitionen können existieren, die nie Schaltbereit sind. Das wäre im Erreichbarkeitsgraphen nicht ersichtlich.

Kapazitäten können oft nicht genau abgelesen werden, sondern nur nach unten beschränkt werden.

↳ Also: Nein