(A) Beweise

18. Sei $lin(bpo)$ die Menge aller Linearisierungen einer BPO $bpo = (V, \prec, l)$. Dann gilt
$$\bigcap_{(V,<,l)\in lin(bpo)} < = \prec .$$

19. Sei σ eine aktivierte Schrittfolge eines markierten S/T-Netzes (N, m_0). Dann ist jede Linearisierung von σ eine aktivierte Schaltfolge.

20. Sei $K = (O, \rho)$, $O = (B, E, G)$, ein Prozess eines markierten S/T-Netzes (N, m_0), $N = (S, T, F, W)$, und C ein Schnitt aus Bedingungen von K. Dann ist die durch $m_C(s) = |\{b \in C \mid \rho(b) = s\}|$ für $s \in S$ gegebene Markierung erreichbar.

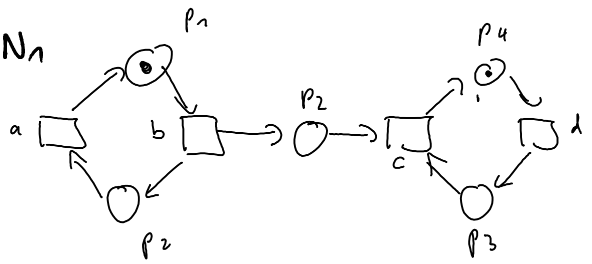
21. Sei m eine erreichbare Markierung eines markierten S/T-Netzes (N, m_0). Dann gibt es einen Prozess $K = (O, \rho)$, $O = (B, E, G)$, und einen Schnitt C aus Bedingungen von K, s.d. $m_C = m$ (*Hinweis*: Konstruiere K aus einer Schaltfolge, die zu m führt).

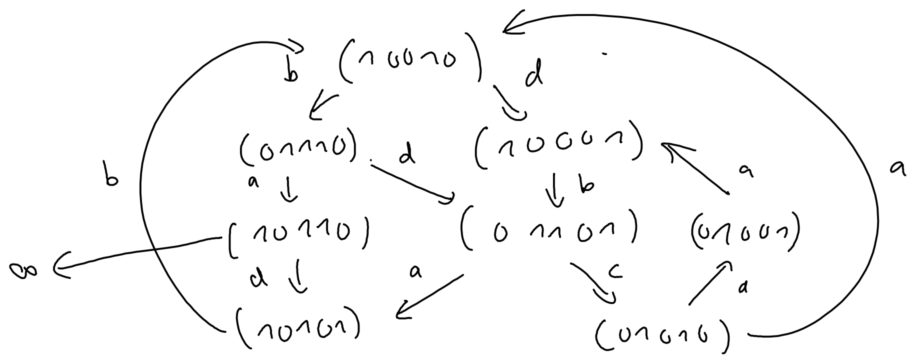
22. Prozessabläufe sind aktivierte BPOs (*Hinweis*: Zeigen Sie, dass jede Schritt-Linearisierung eines Prozessablaufs eine aktivierte Schrittfolge ist).

(B) Beispiele Es seien die markierten S/T-Netze N_1, N_2 und N_3 der nächsten Seite gegeben

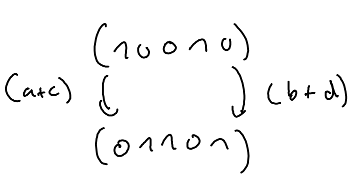
23. Bestimmen Sie den Markierungsgraphen aller drei Netze.
24. Bestimmen Sie den Schritt-Markierungsgraphen der drei Netze.
25. Geben Sie für jedes Netz zwei verschiedene Prozesse mit jeweils mindestens 6 Ereignissen zusammen mit ihren Prozessabläufen an.
26. Diskutieren Sie das Verhalten der drei Netze – was können Sie aus den Verhaltensbeschreibungen ablesen?

23) N_1



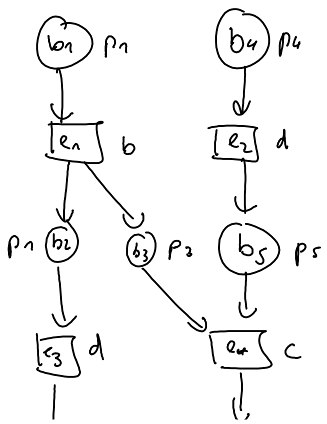


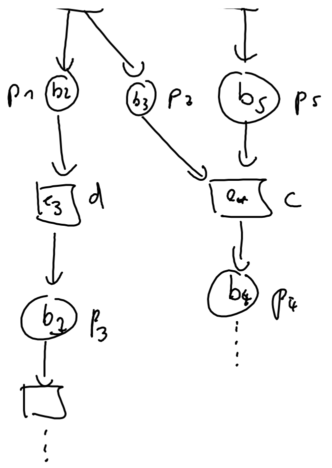
Schrittmarkierungsgraph (?):



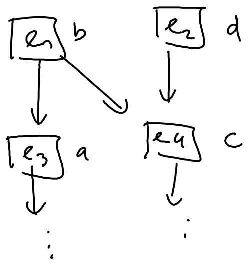
Was auch immer das ist?

Prozessgraph:





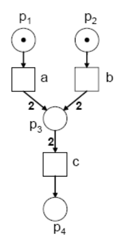
Ereignisgraph (?)



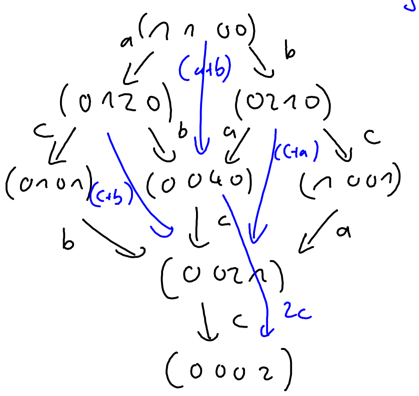
$\langle p_1, p_2, p_3, p_4 \rangle$ (?)

Schrittmarkengraph

N3

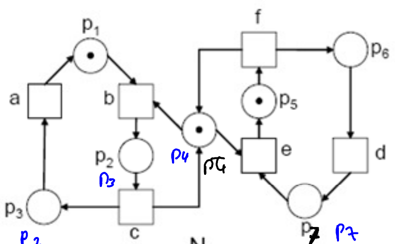


N_3

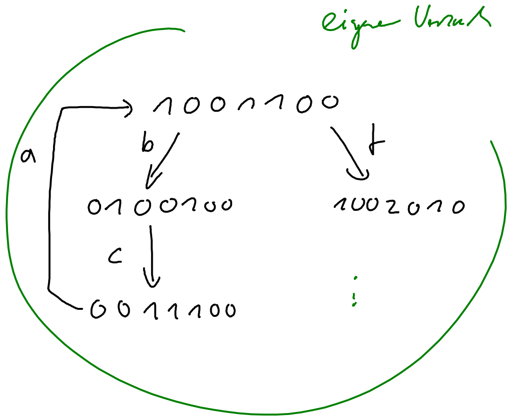


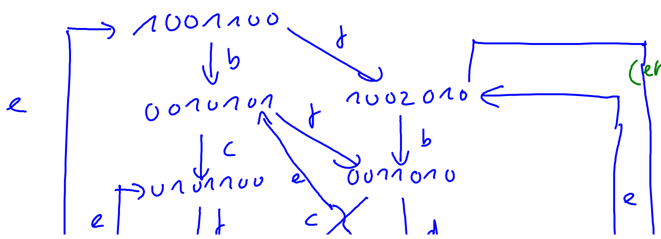
N2

eigene Versuch

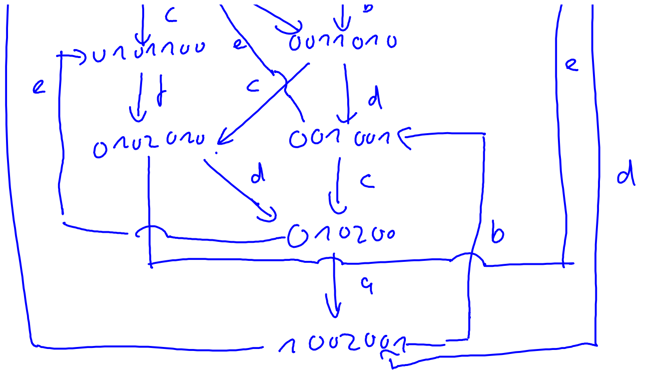


N_2





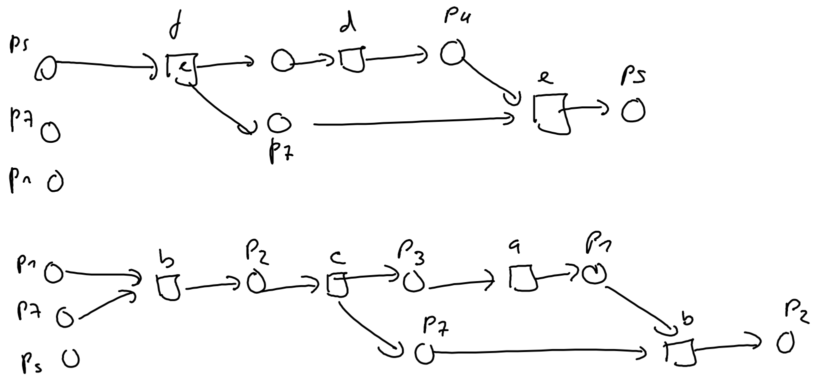
(keinen vergleichs Vorgang Netz (Prof))



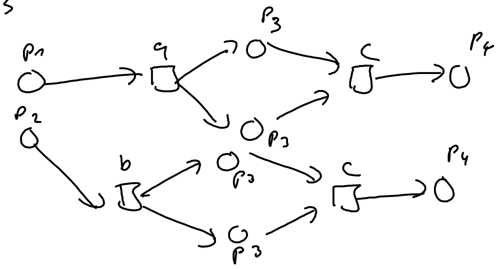
(?)

(original Beschriftung)

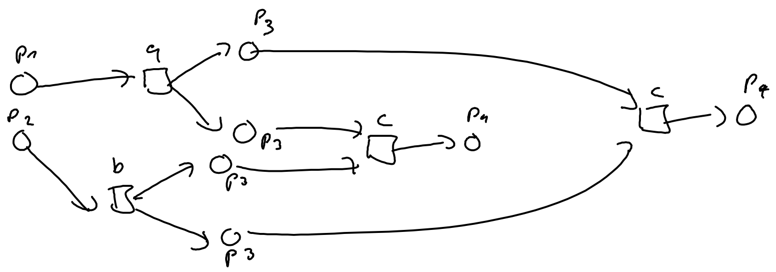
N2



N3



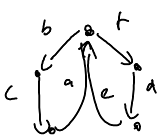
alternative Prozess



26) Markierungsgraph: Zyklen → unendlich
keine Zyklen → endlich

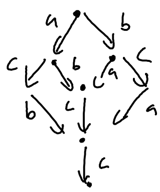
N_1 nicht beschränkt (?)

N_3



(so hätte es sein sollen)
„Art geteilte Ressource"
⇒ hier gibt es keine
Weise Kapazität

N_2



20)

$K = (O, \rho)$, $O = (B, E, G)$ Prozess von (N, m_0)

$N = (S, T, F, W)$

C Bedingungsschnitt von K — maximale Menge von Bedingungen im Prozess

Definiere m_C — siehe Bedeutung

20. Sei $K = (O, \rho)$, $O = (B, E, G)$, ein Prozess eines markierten S/T-Netzes (N, m_0), $N = (S, T, F, W)$, und C ein Schnitt aus Bedingungen von K. Dann ist die durch $m_C(s) = |\{b \in C \mid \rho(b) = s\}|$ für $s \in S$ gegebene Markierung erreichbar.

$m_0 \xrightarrow{\sigma} m_C$ für Schaltsequenz σ — dramatische Stille

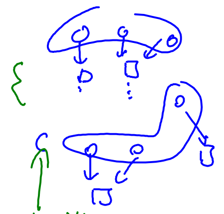
Beweis: Sei $E' = \{e \in E \mid \exists b \in C : (e, b) \in G^+\}$

alle e_i in Sequenz bringen

{ Sei $E' = \{e_1, \dots, e_n\}$, s.d. (so, dass) $(e_i, e_j) \in G^+ \Rightarrow i < j$

Setze $\sigma = \ell(e_1) \dots \ell(e_n)$

$m_0 \xrightarrow{\sigma} m_C$ zeige wir induktiv nach n



n wenn alle Ereignisse vor c Schnitt enthält C erreicht

Schnitt

$\underline{n = 0}$: (Es gibt keine Ereignisse vor $c \Rightarrow c$ ist Anfangsschnitt)

das bedeutet $E' = \emptyset$

$\Rightarrow C = \mathrm{Min}(O)$ ✓

$\underline{n \to n+1}$:

Sei $\sigma = \ell(e_1), \dots \ell(e_n) \ell(e_{n+1})$

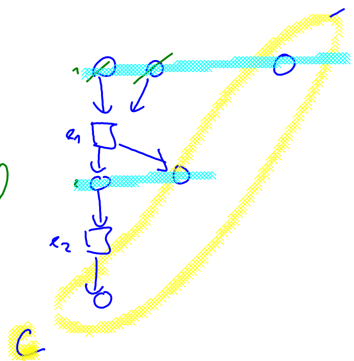
Sei $\sigma' = \ell(e_1), \dots \ell(e_n)$ (hier Teil der Induktionsvoraussetzung an)

Sei $C' = \left[\mathrm{Min}(O) \cup \left(\bigcup_{i=1}^{n} e_i^\bullet\right)\right] \setminus \left(\bigcup_{i=1}^{n} {}^\bullet e_i\right)$

$\Rightarrow E(C') = \{e_1, \dots, e_n\}$

Induktions-Voraussetzung — IV

$\overset{IV}{\Rightarrow} m_0 \xrightarrow{\sigma'} m_{C'}$ (Induktionsvoraussetzung)



Noch zu zeigen $m_{C'} \xrightarrow{\ell(e_{n+1})} m_C$

(Idee zur Konstruktion von Prozessen: e hat für jedes s genau $w(s, \rho(e))$ Vorbedingungen)

Wir haben: Für jede Stelle s hat e_{n+1} genau $W(s, \rho(e_{n+1}))$ Vorbedingungen (Def. von Prozessen 2.3.2 (..)(b))

$\Rightarrow \forall_s : m_{C'}(s) \geq W(s, \rho(e_{n+1})) \Rightarrow$ Beh ✓