System: Hardware- und Software-Systeme

System-Validierung:

- traditionell: Testen und Simulation

am Anfang bei einfachen Fehlern sehr wirksam

---> aber immer weniger effektiv, wenn komplexe und verborgene Fehler auftreten,
---> insbesondere bei bei Systemen mit Asynchronität, Parallelität, Nebenläufigkeit,
---> Fehler oft von speziellen Zeitparametern/Nachrichtenlaufzeiten abhängig

- alternativ: formale Verifikation

---> umfassende Prüfung
---> alle Zweige des Verhaltens werden geprüft

wichtige Vorgehensweisen:

---> Hoare-Systeme (Zusicherungen, Invarianten)

---> temporale Logik

----> Analyse des Zustandsraumes (model checking)

Zustandsraum-Analyse/Model Checking

Vorteil:

---> ohne besondere Kenntnisse anwendbar
---> bei nicht korrektem System:
Generierung von Abläufen die zu den Fehlern führen

Nachteil:

---> Größe des Zustandsraumes

Abhilfe:

---> symbolisches Model Checking
---> Faltung
---> Symmetrien
---> ....

Literatur:

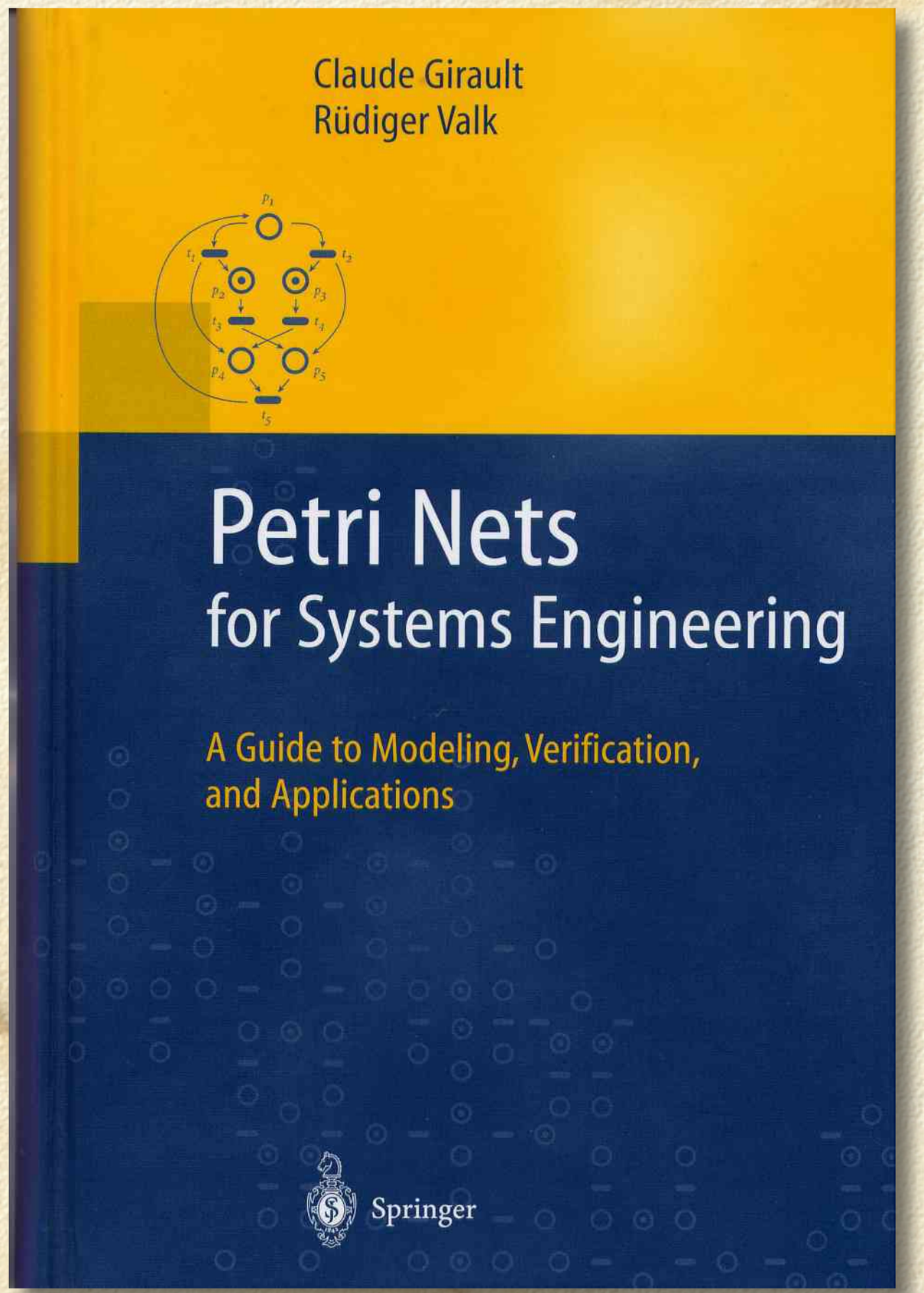
E.M. Clarke et al.: *Model Checking*, The MIT Press, Cambridge, 1999, [CGP99]

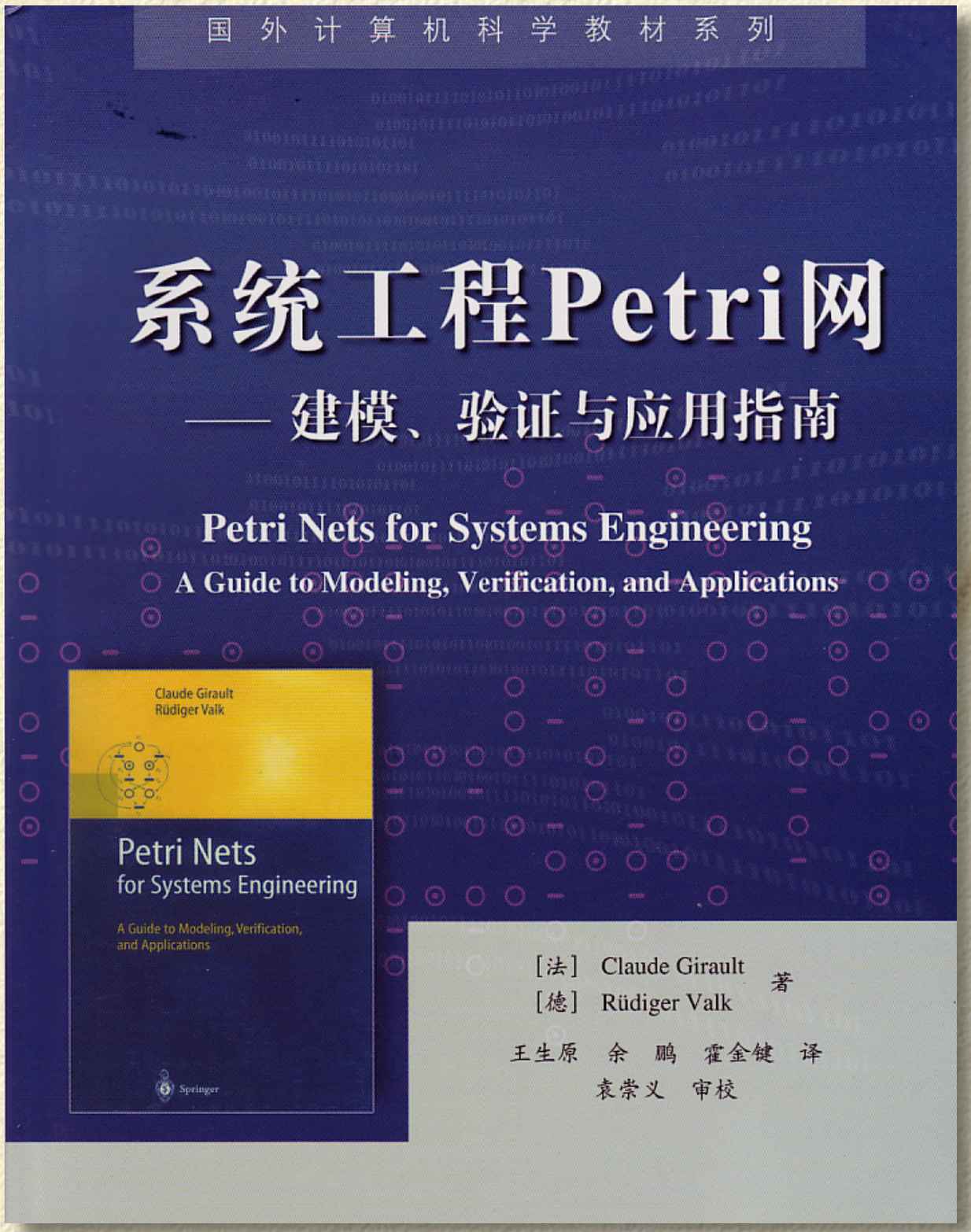
B. Bèrard et al.: *Systems and Software Verification: Model-Checking Techniques and Tools*, Springer, Berlin, 1999, [BBF99]

C. Girault, R. Valk: *Petri Nets for Systems Engineering*, Part III: Verification, Springer, Berlin, 2003, [GV03]

M. Huth, M. Ryan: *Logic in Computer Science*, Cambridge Univ. Press, 2004, [HR04]

C. Baier, J.-P. Katoen: *Principles of Model Checking*, MIT Press, 2008, [BK08]


Claude Girault
Rüdiger Valk
Petri Nets
for Systems Engineering
A Guide to Modeling, Verification,
and Applications
Springer

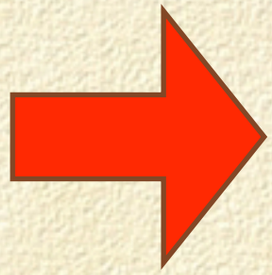

国外计算机科学教材系列
系统工程Petri网
——建模、验证与应用指南
Petri Nets for Systems Engineering
A Guide to Modeling, Verification, and Applications
Claude Girault
Rüdiger Valk
Petri Nets
for Systems Engineering
A Guide to Modeling, Verification,
and Applications
Springer
[法] Claude Girault
[德] Rüdiger Valk
著
王生原 余 鹏 霍金键 译
袁崇义 审校

Verifikation eines Systems

in Kapitel 1:

$$L(TS_{sys}) \subseteq L(TS_{spec})$$
$$L^{\omega}(TS_{sys}) \subseteq L^{\omega}(TS_{spec})$$

System-Verhalten



System-Spezifikation

Verifikation eines Systems

in Kapitel 1:

$$L(TS_{sys}) \subseteq L(TS_{spec})$$
$$L^{\omega}(TS_{sys}) \subseteq L^{\omega}(TS_{spec})$$

System-Verhalten → *System-Spezifikation*

„system enjoys property"

Theorem Proving:

Prozessalgebra

„Systems ***formula*** *implies property* ***formula.***"

$$\phi \Longrightarrow f$$

Model Checking:

„Systems ***semantics*** *is model of property* ***formula.***"

$$M, s \models f$$

Modell

Programm
(auch nebenläufiges)

State Charts

Prozessalgebra-Modell

Petrinetz

....

Verhalten

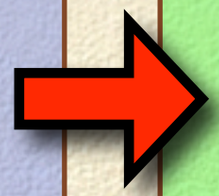
Zustandsgraph

Transitionssystem

Prozessgraph

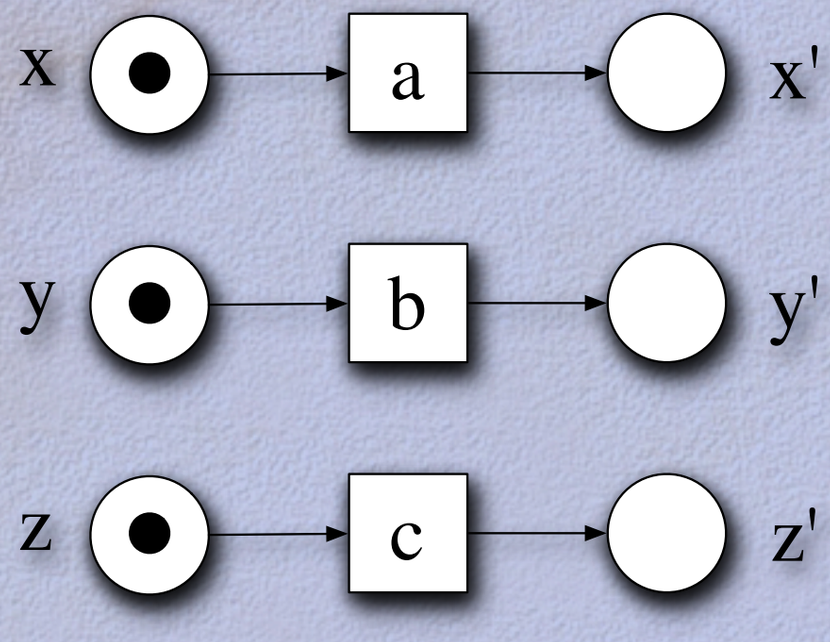
Erreichbarkeitsgraph
Markierungsgraph

....



Zustandsexplosion

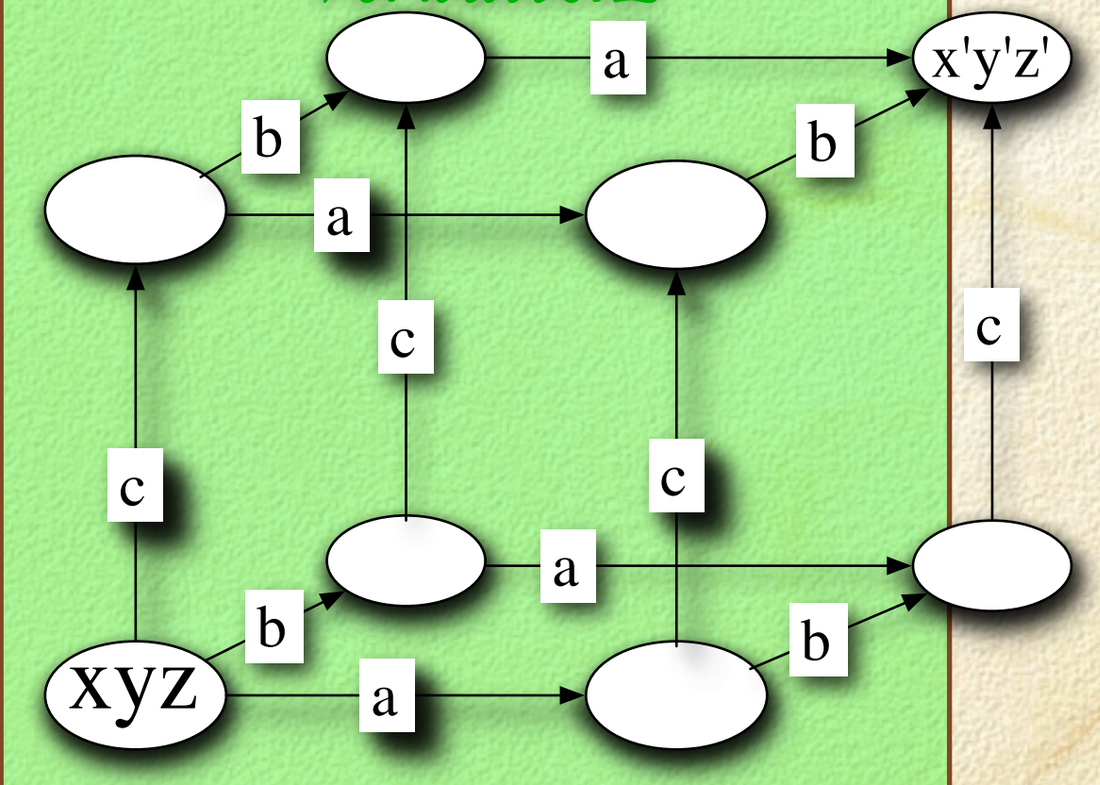
Modell



Petrinetz

....

Verhalten



Erreichbarkeitsgraph
Markierungsgraph

....

Zustandsexplosion

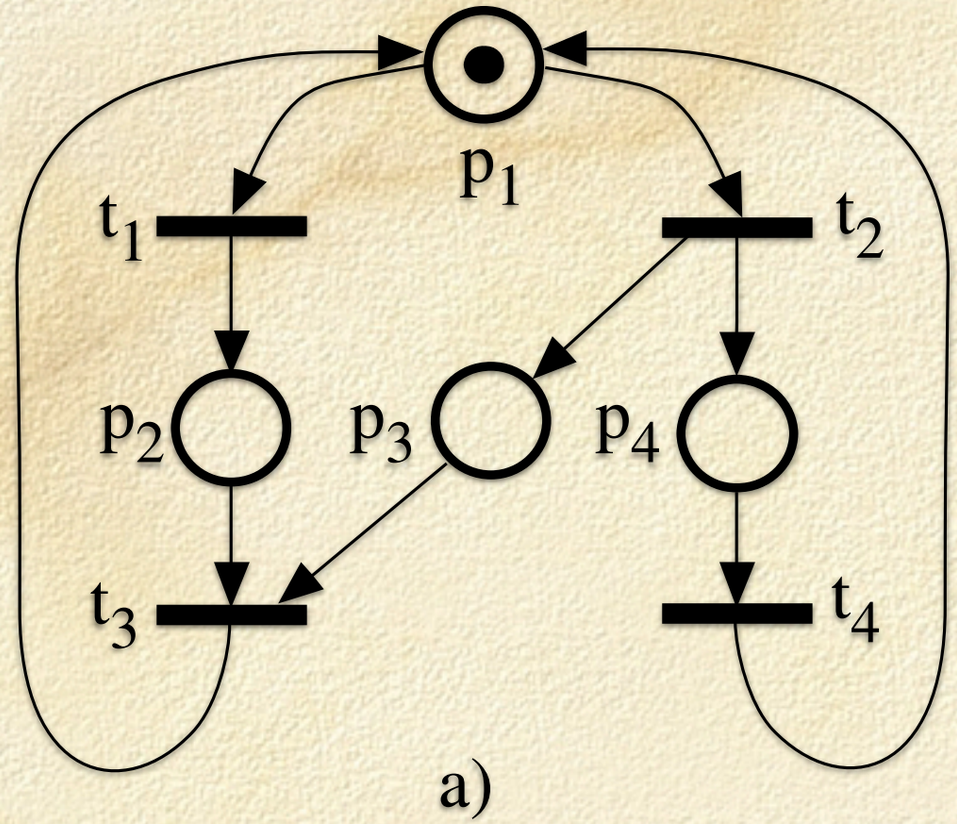
3.2 Elementare Systemeigenschaften

1) *Beschränktheit (boundedness)*, was die Endlichkeit des Zustandraumes bedeutet,

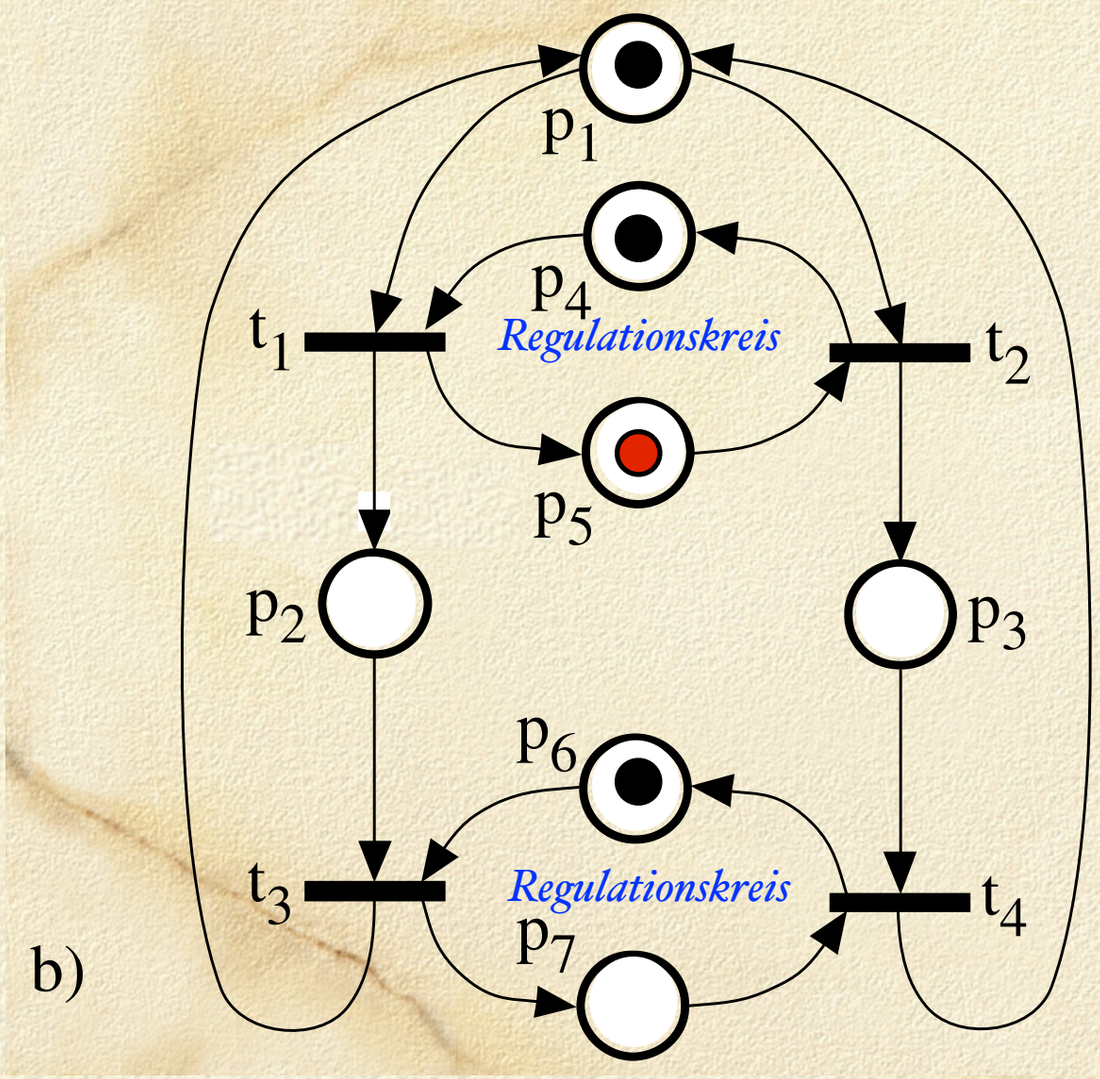
2) *Lebendigkeit (liveness)*, was die potenzielle Ausführbarkeit bedeutet,

3) *Reversibilität (reversibility)*, was diejenigen Systeme charakterisiert, die immer in den Anfangszustand zurückgesetzt werden können,

4) *wechselseitiger Ausschluss (mutual exclusion)*, was die Unmöglichkeit von simultanen Teilmarkierungen (p-mutex) oder Transitionsausführungen (t-mutex) bedeutet.



a)



b)

Abbildung 3.1 : Beispiele: (a) ein unbeschränktes, nicht-lebendiges und nicht-reversibles Netz (b) ein lebendiges Netz, das aber bei vergrößerter Anfangsmarkierung (z.B. $\mathbf{m_0}[p_5] = 1$) nicht lebendig ist.

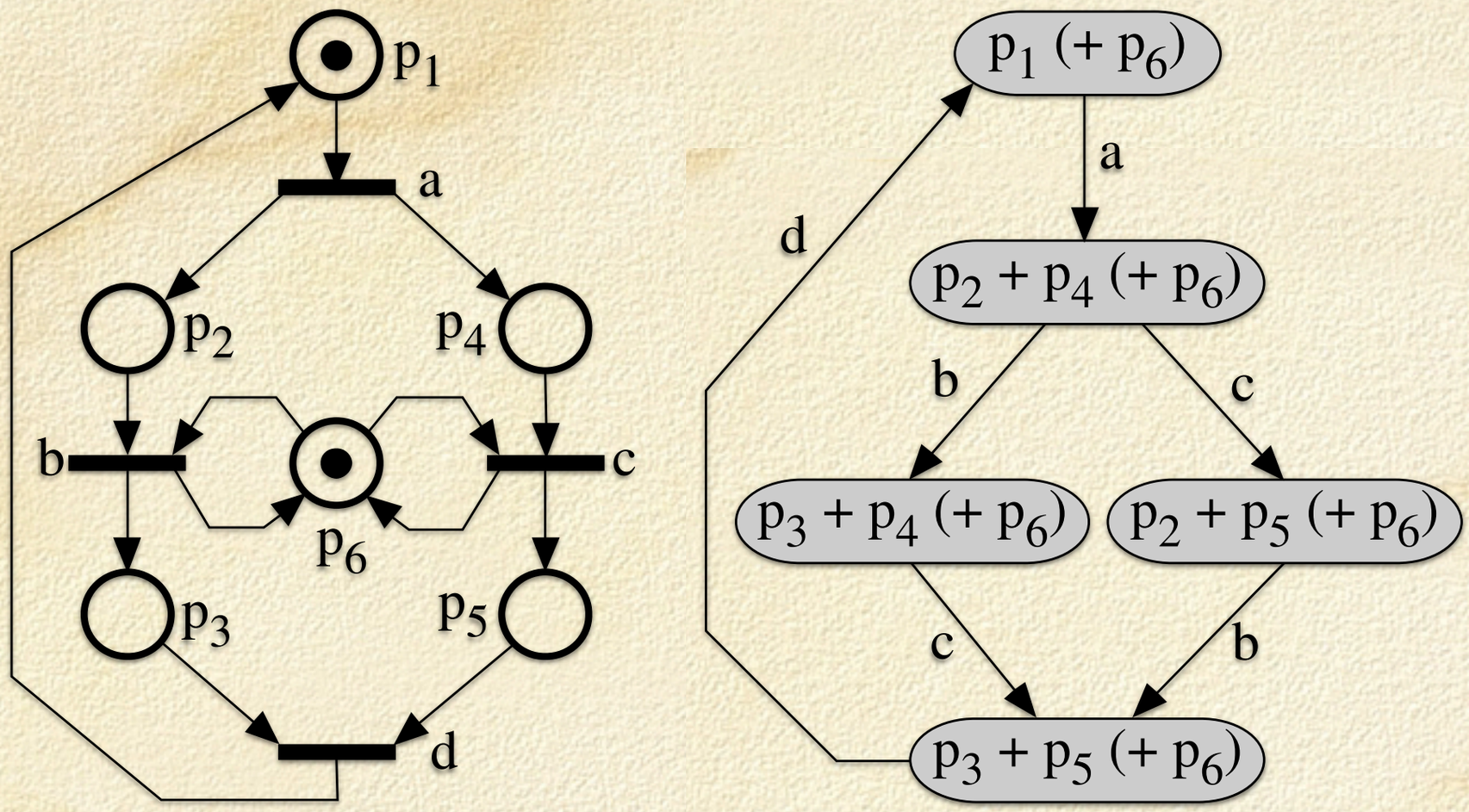
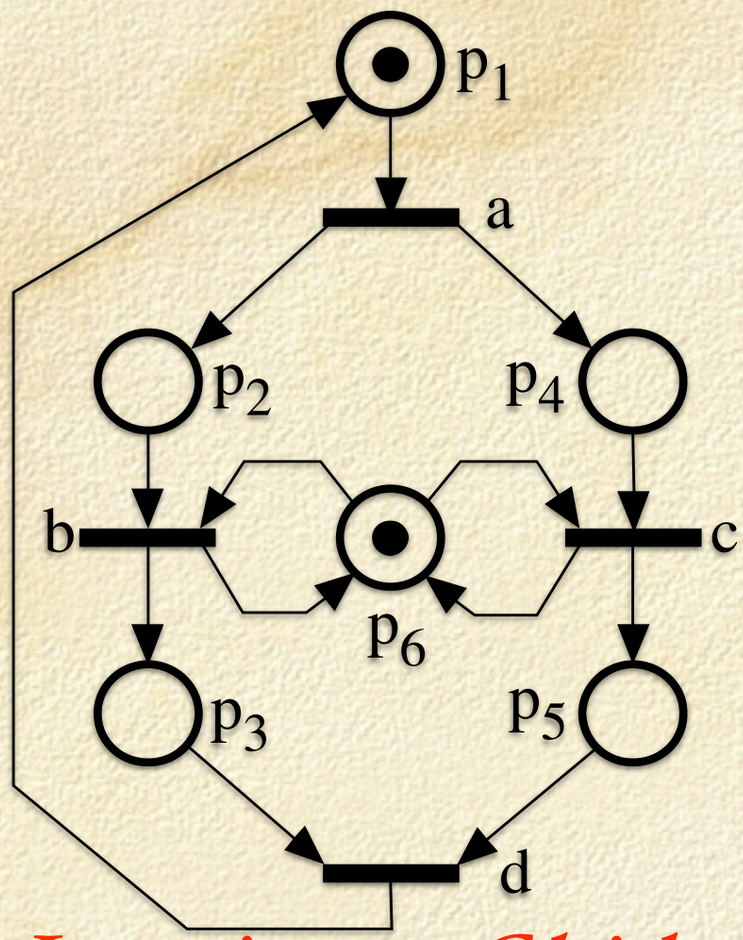
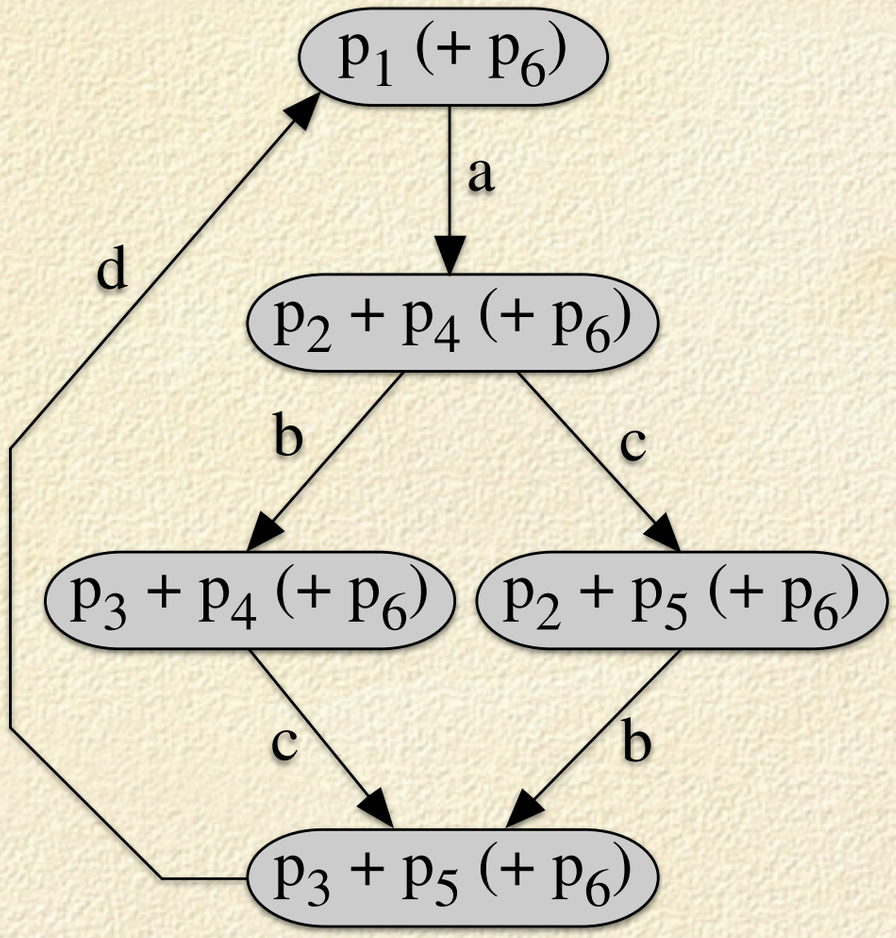


Abbildung 3.2 Beschränktes, lebendiges und reversibles Netz mit Erreichbarkeitsgraph





Platz-Invarianten-Gleichungen

$$\mathbf{m}[p_1] + \mathbf{m}[p_2] + \mathbf{m}[p_3] = \mathbf{m_0}[p_1] + \mathbf{m_0}[p_2] + \mathbf{m_0}[p_3] = k_1(\mathbf{m_0})$$

$$\mathbf{m}[p_1] + \mathbf{m}[p_4] + \mathbf{m}[p_5] = \mathbf{m_0}[p_1] + \mathbf{m_0}[p_4] + \mathbf{m_0}[p_5] = k_2(\mathbf{m_0})$$

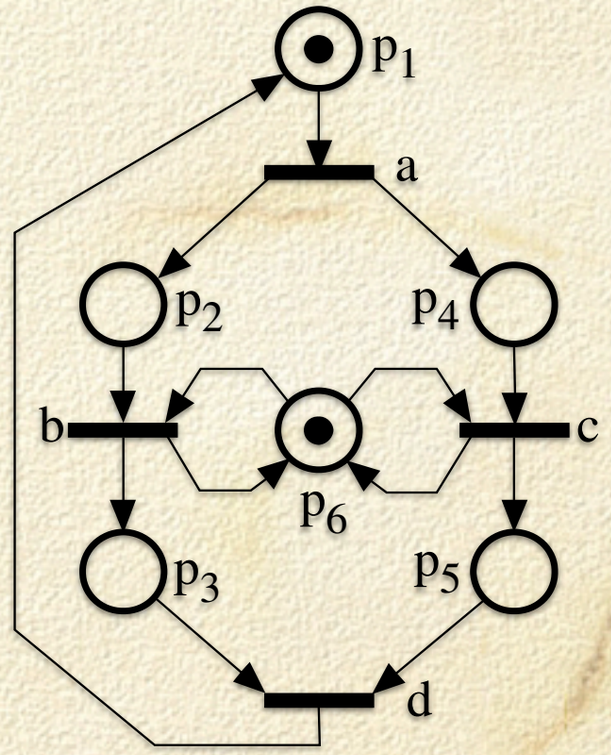
$$\mathbf{m}[p_6] = \mathbf{m_0}[p_6] = k_3(\mathbf{m_0})$$

$$\mathbf{m}[p_1] \leq \min(k_1(\mathbf{m_0}), k_2(\mathbf{m_0}))$$

$$\mathbf{m}[p_i] \leq k_1(\mathbf{m_0}); i = 2, 3$$

$$\mathbf{m}[p_j] \leq k_2(\mathbf{m_0}); j = 4, 5$$

$$\mathbf{m}[p_6] = k_3(\mathbf{m_0})$$



Platz-Invarianten-Gleichungen

$$\mathbf{m}[p_1] + \mathbf{m}[p_2] + \mathbf{m}[p_3] = \mathbf{m_0}[p_1] + \mathbf{m_0}[p_2] + \mathbf{m_0}[p_3] = k_1(\mathbf{m_0})$$

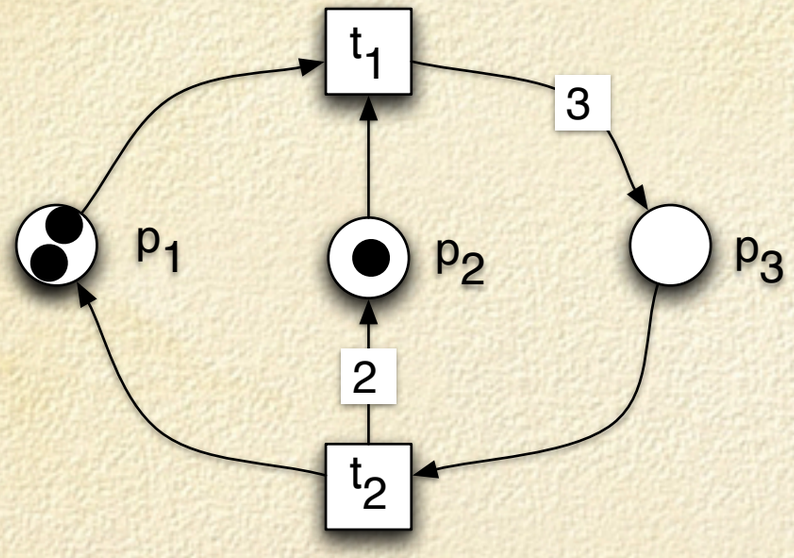
$$\mathbf{m}[p_1] + \mathbf{m}[p_4] + \mathbf{m}[p_5] = \mathbf{m_0}[p_1] + \mathbf{m_0}[p_4] + \mathbf{m_0}[p_5] = k_2(\mathbf{m_0})$$

$$\mathbf{m}[p_6] = \mathbf{m_0}[p_6] = k_3(\mathbf{m_0})$$

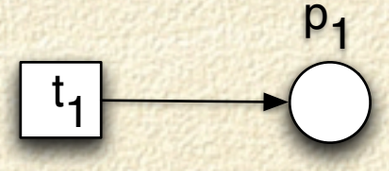
(1) Sei $k \in \mathbb{N}$. Dann heißt ein Platz $p \in P$ *k-beschränkt* (k-bounded) in $\mathcal{N}$, falls $\forall \mathbf{m} \in \mathbf{R}(\mathcal{N}) : \mathbf{m}(p) \leq k$

(2) p heißt *beschränkt* (bounded) in $\mathcal{N}$, falls $\exists k \in \mathbb{N}\, \forall \mathbf{m} \in \mathbf{R}(\mathcal{N}) : \mathbf{m}(p) \leq k$

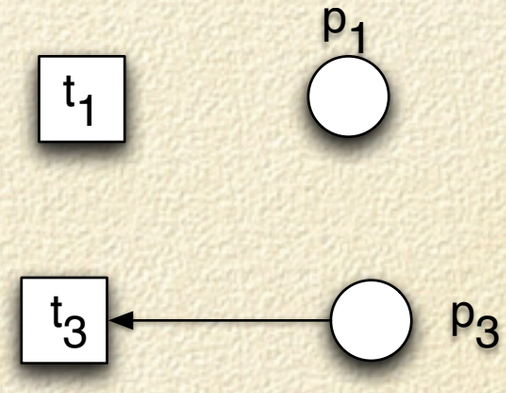
(3) $\mathcal{N}$ heißt k-beschränkt bzw. beschränkt, wenn alle Plätze k-beschränkt bzw. beschränkt sind.



das "kleinste" unbeschränkte Netz?



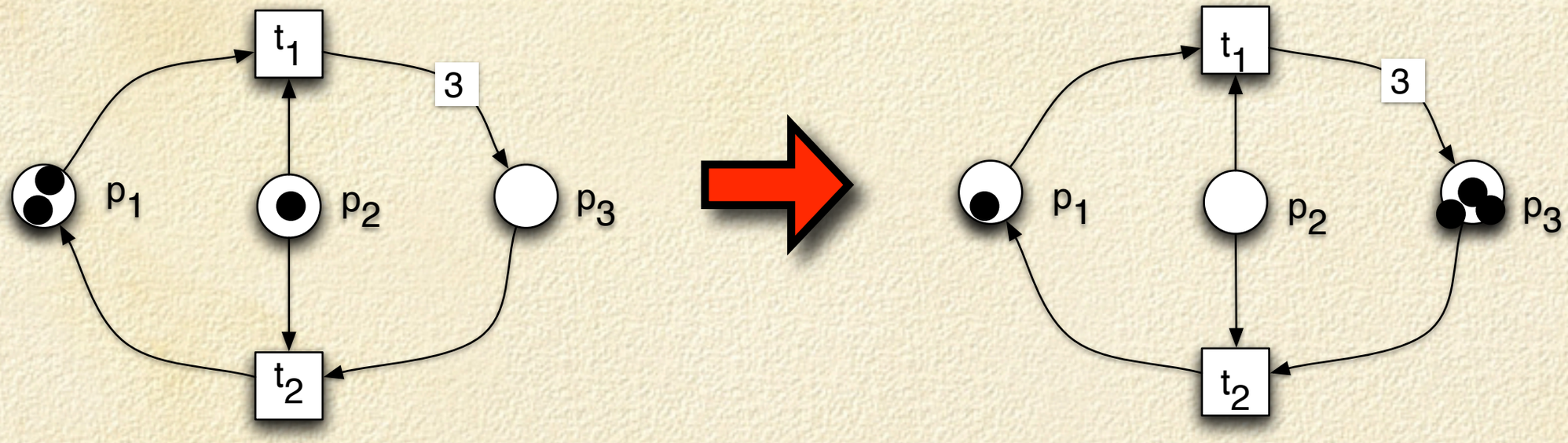
das "kleinste" beschränkte Netz?



$$\forall p \in P \;\; \exists k \in \mathbb{N} \;\; \forall \mathbf{m} \in \mathbf{R}(\mathcal{N}) \;\; : \mathbf{m}(p) \leq k$$

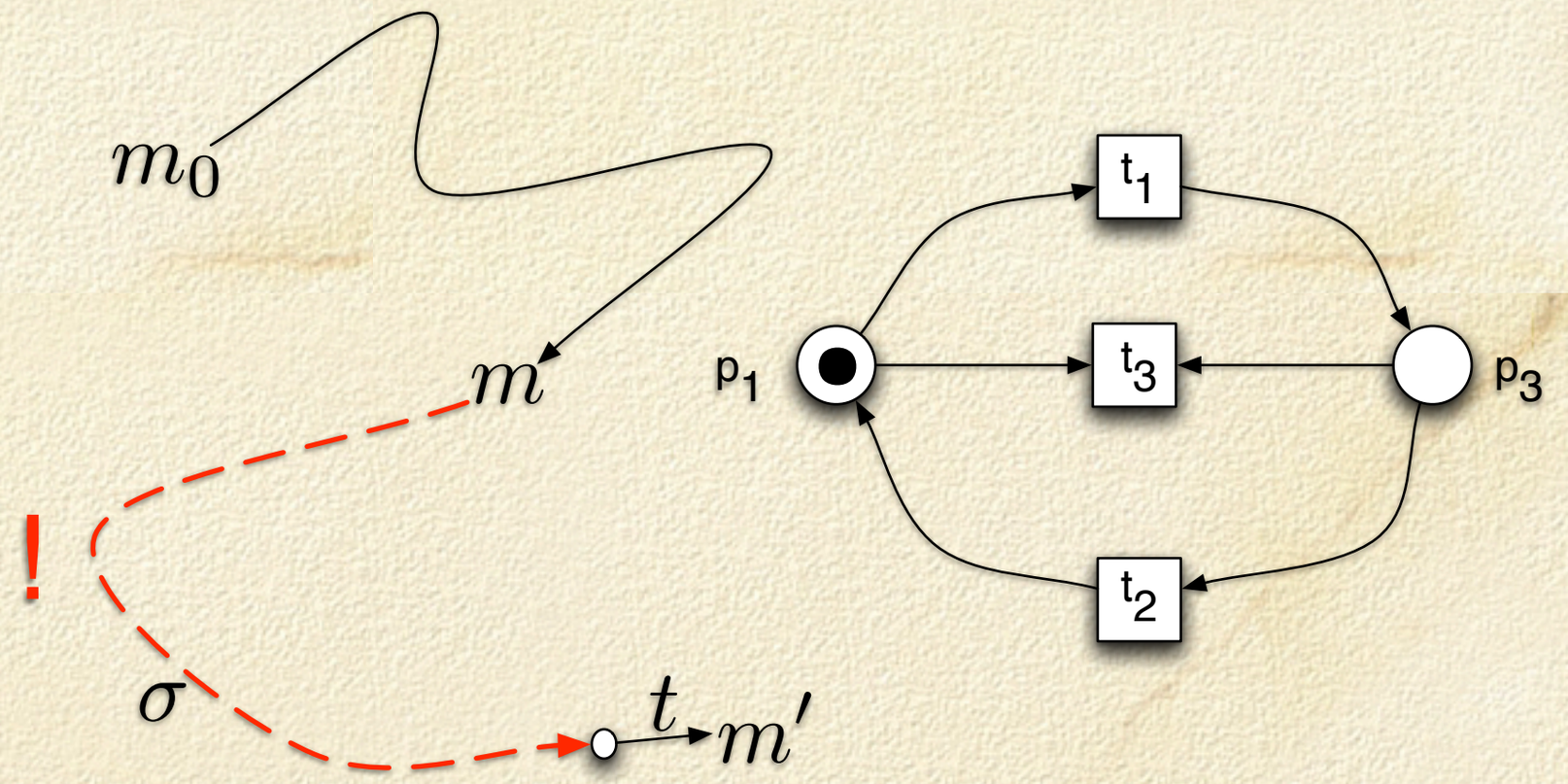
(4) $\langle \mathcal{N}, \mathbf{m_0} \rangle$ heißt *verklemmungsfrei* (deadlock-free) falls

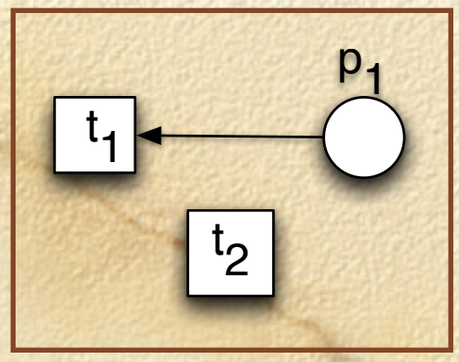
$\forall \mathbf{m} \in \mathbf{R}(\mathcal{N}, \mathbf{m_0}).\ \exists t \in T : \mathbf{m} \xrightarrow{t}$



(4) $\langle \mathcal{N}, \mathbf{m_0} \rangle$ heißt *verklemmungsfrei* (deadlock-free) falls
$\forall \mathbf{m} \in \mathbf{R}(\mathcal{N}, \mathbf{m_0}).\ \exists t \in T : \mathbf{m} \xrightarrow{t}$

(5) t heißt *lebendig* (live) in $\langle \mathcal{N}, \mathbf{m_0} \rangle$ falls
$\forall \mathbf{m} \in \mathbf{R}(\mathcal{N}, \mathbf{m_0}).\ \exists \sigma \in T^* : \mathbf{m} \xrightarrow{\sigma t} \mathbf{m'}$

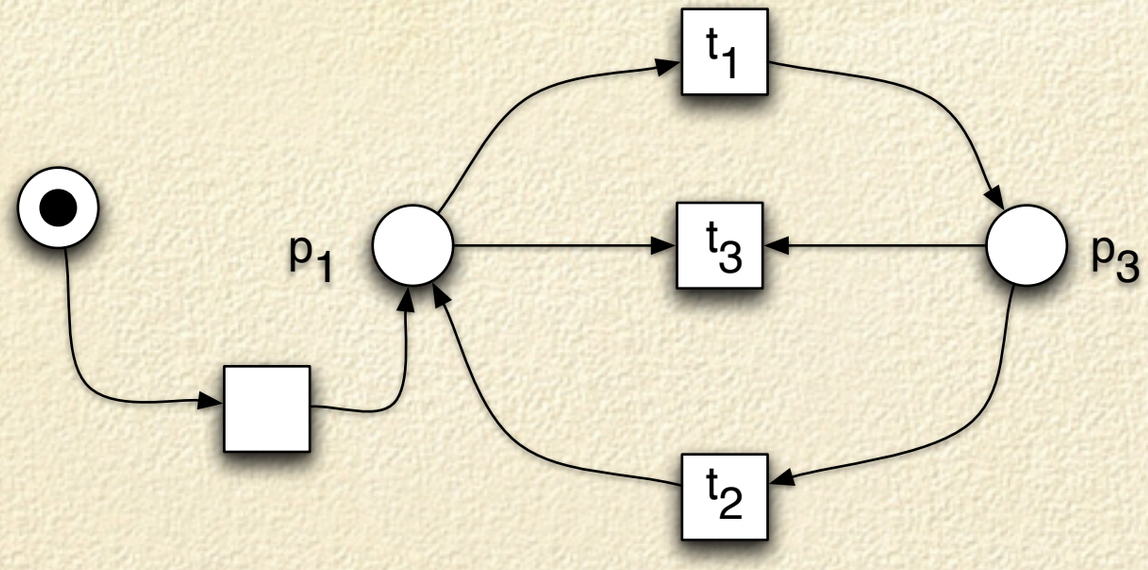




verklemmungsfrei aber nicht lebendig

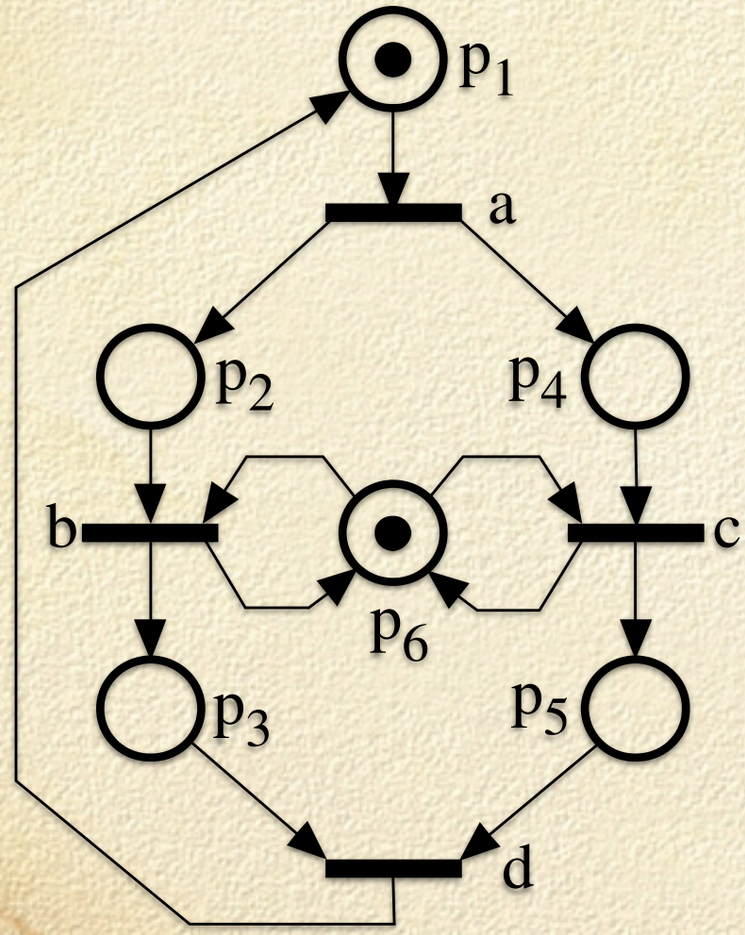
(7) $\mathbf{m} \in \mathbf{R}(\mathcal{N}, \mathbf{m_0})$ heißt *Rücksetzzustand* (home state) falls
$\forall \mathbf{m'} \in \mathbf{R}(\mathcal{N}, \mathbf{m_0}).\ \exists \sigma \in T^*\colon \mathbf{m'} \xrightarrow{\sigma} \mathbf{m}$

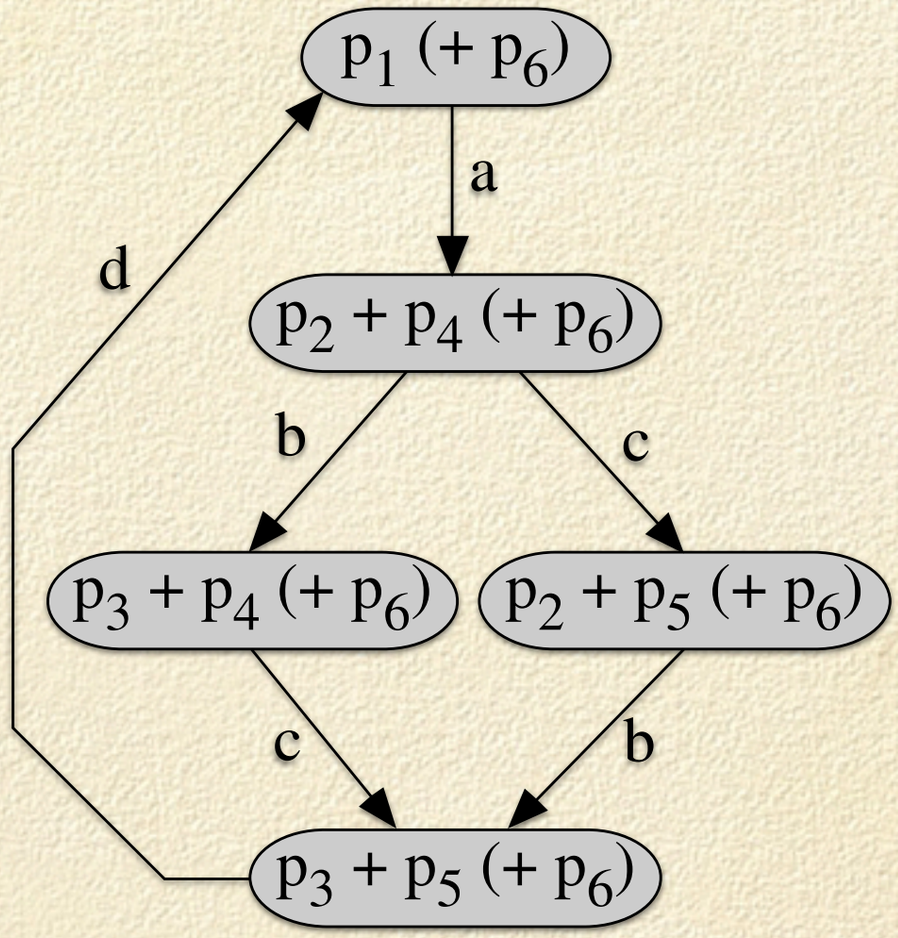
(8) $\langle \mathcal{N}, \mathbf{m_0} \rangle$ heißt *reversibel* (reversible) falls
$\forall \mathbf{m} \in \mathbf{R}(\mathcal{N}, \mathbf{m_0}).\ \exists \sigma \in T^*\ \colon \mathbf{m} \xrightarrow{\sigma} \mathbf{m_0}$



Ein Netz ist reversibel, falls $\mathbf{m}_0$ *ein Rücksetzzustand ist.*

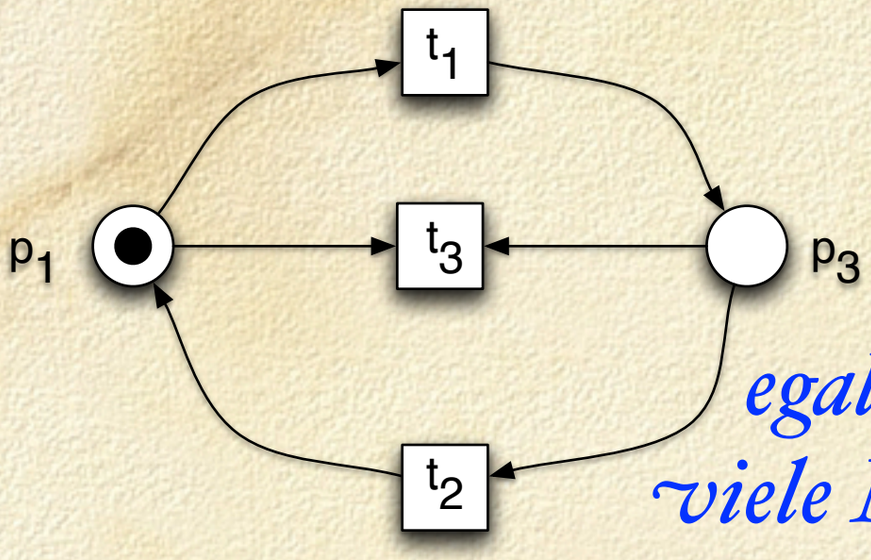
(9) *wechselseitiger Ausschluss* (mutual exclusion) in $\langle \mathcal{N}, \mathbf{m_0} \rangle$:
p_i und p_j sind in *Markierungs-Ausschluss* (marking mutual exclusion) falls
$\nexists \mathbf{m} \in \mathbf{R}(\mathcal{N}, \mathbf{m_0})$: $(\mathbf{m}[p_i] > 0) \wedge (\mathbf{m}[p_j] > 0)$
t_i und t_j sind in *Schalt-Ausschluss* (firing mutual exclusion) falls
$\nexists \mathbf{m} \in \mathbf{R}(\mathcal{N}, \mathbf{m_0}) : \mathbf{m} \geq W(\bullet, t_i) + W(\bullet, t_j)]$

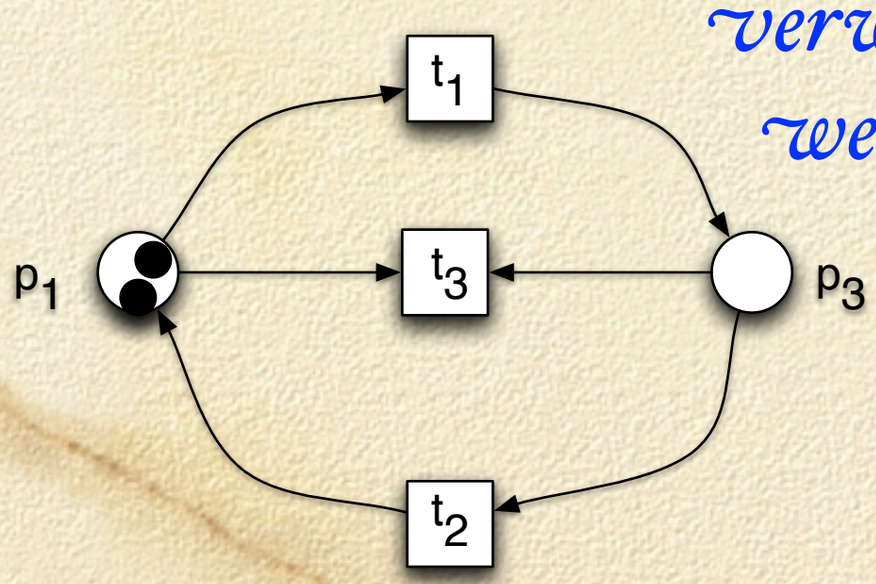




(10) **Strukturelle Eigenschaften :**

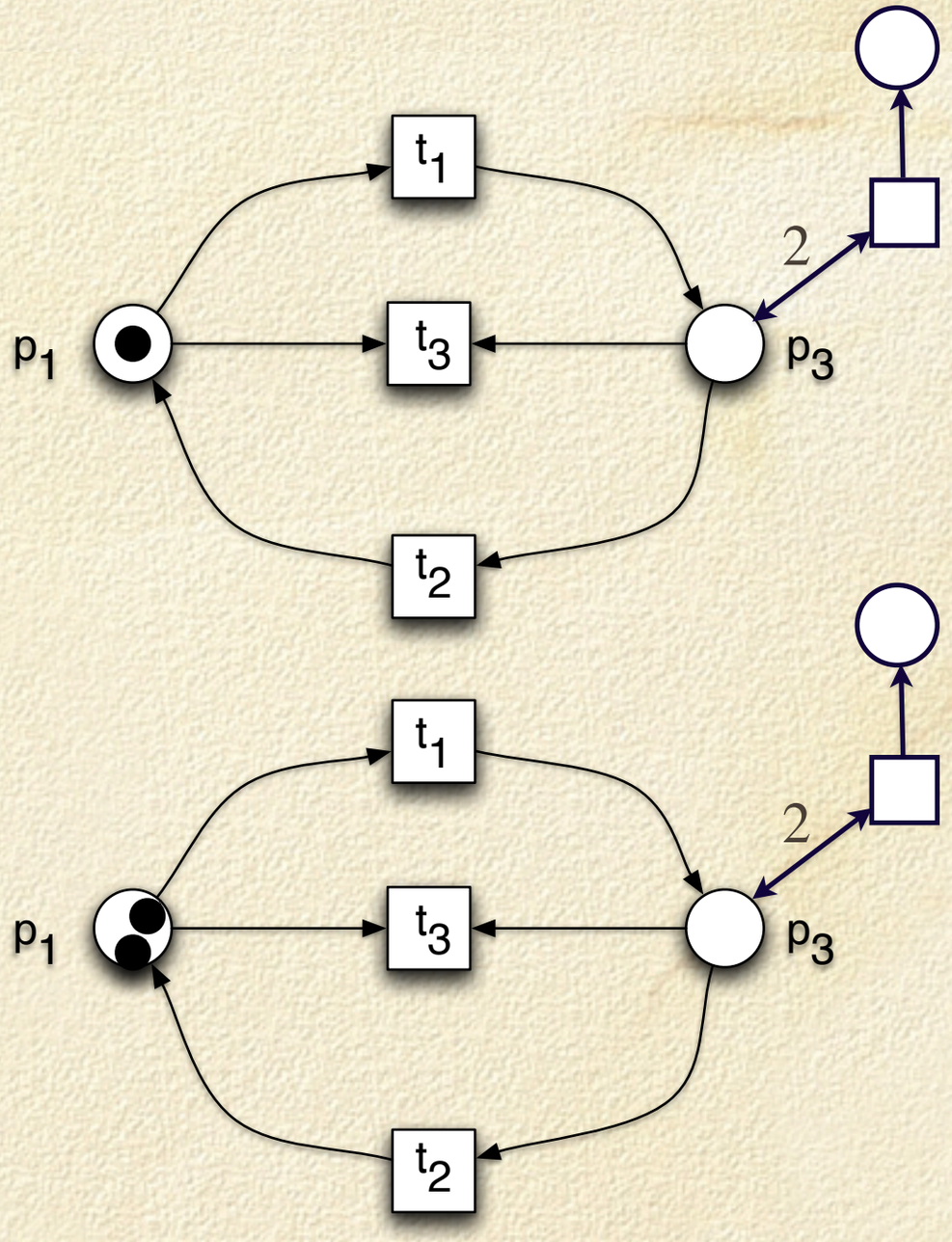
$\mathcal{N}$ heißt *strukturell beschränkt* (structurally bounded) falls $\forall \mathbf{m_0}$: $\langle \mathcal{N}, \mathbf{m_0} \rangle$ ist beschränkt





strukturell beschränkt

egal, wie viele Marken verwendet werden



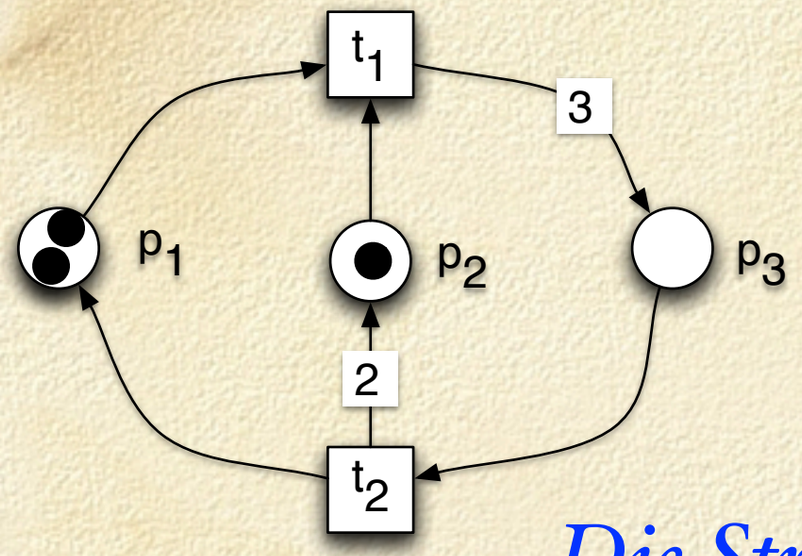
strukturell **nicht** beschränkt

Die Struktur erzwingt Beschränktheit.

(10) Strukturelle Eigenschaften :

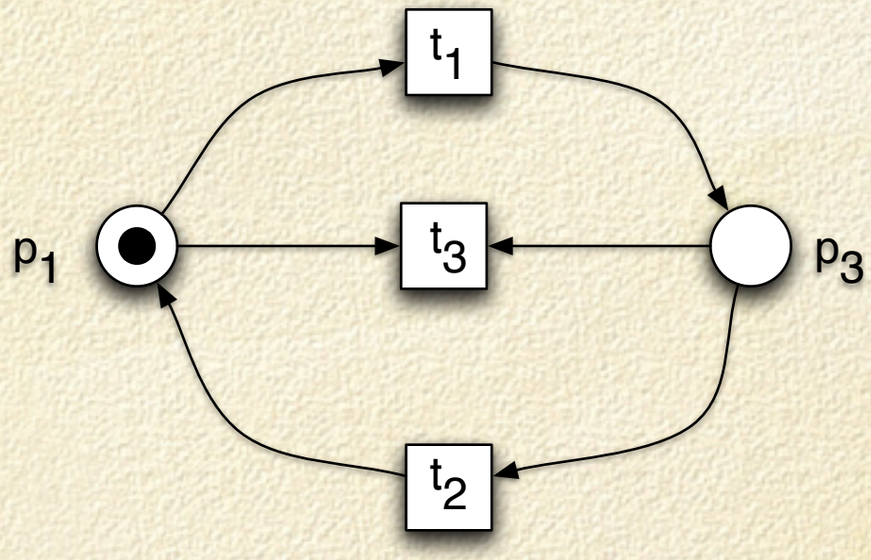
$\mathcal{N}$ heißt *strukturell beschränkt* (structurally bounded) falls $\forall \mathbf{m_0}$: $\langle \mathcal{N}, \mathbf{m_0} \rangle$ ist beschränkt

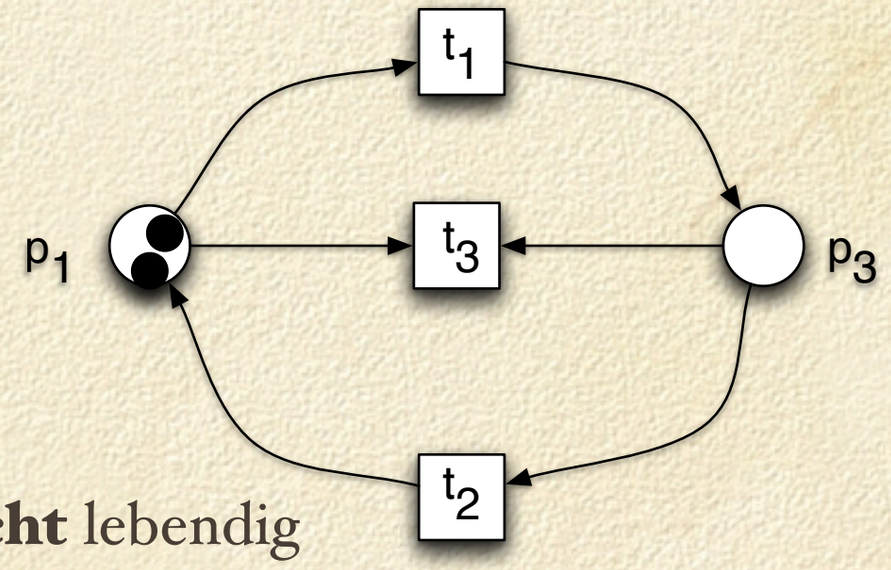
$\mathcal{N}$ heißt *strukturell lebendig* (structurally live) falls $\exists \mathbf{m_0}$: $\langle \mathcal{N}, \mathbf{m_0} \rangle$ ist lebendig



strukturell lebendig

Die Struktur erlaubt eine lebendige Anfangsmarkierung.





strukturell **nicht** lebendig

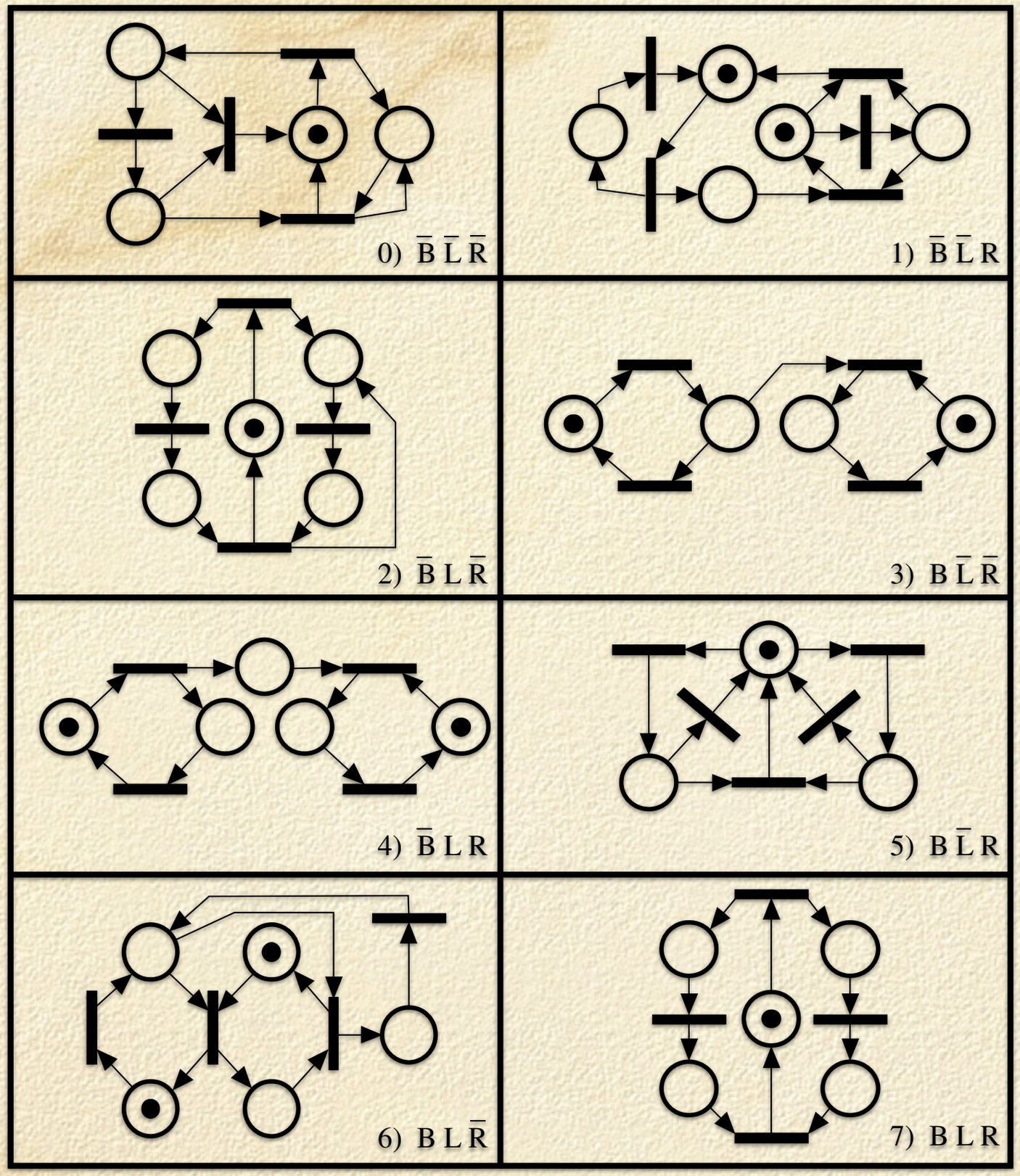


Abbildung 3.4 Beschränktheit (B), Lebendigkeit (L) und Reversibilität (R) sind unabhängige Eigenschaften

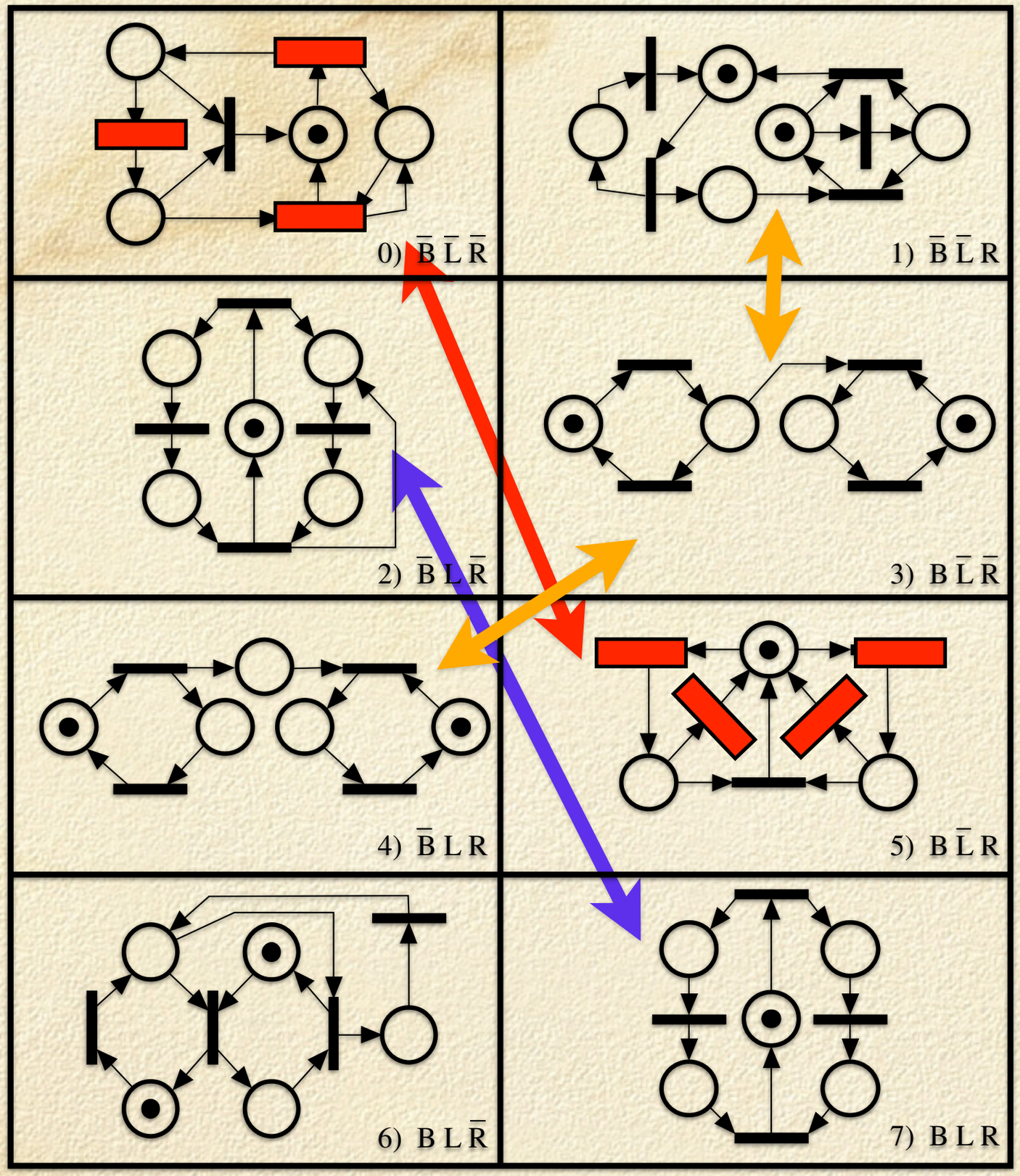


Abbildung 3.4 Beschränktheit (B), Lebendigkeit (L) und Reversibilität (R) sind unabhängige Eigenschaften

3.3 Verifikation durch den Erreichbarkeitsgraphen

Verifikationsmethoden für Petrinetze:

a) enumerative Methoden:

Analyse des Erreichbarkeitsgraphen

b) Transformationen

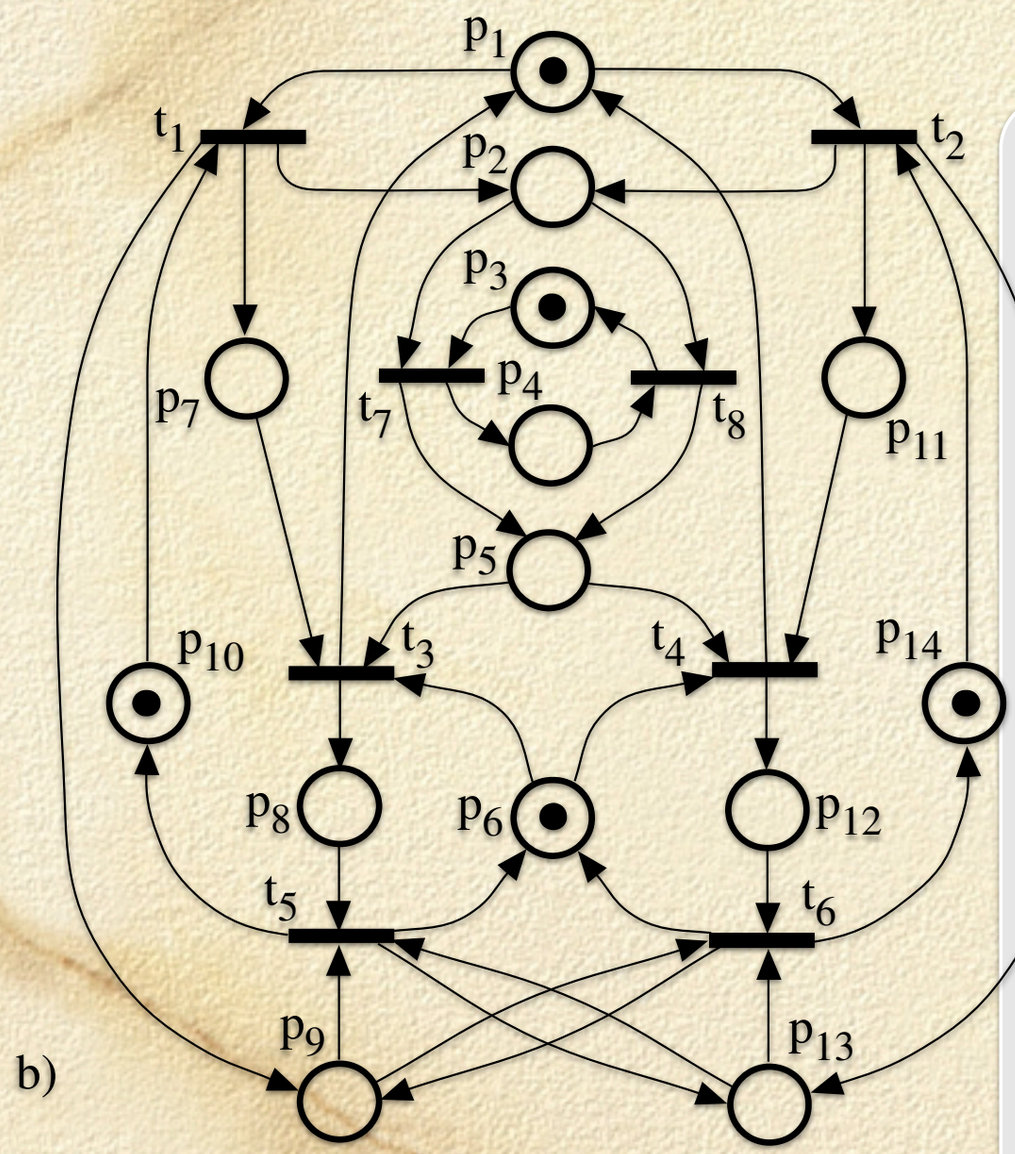
Reduktion

c) strukturelle Analyse

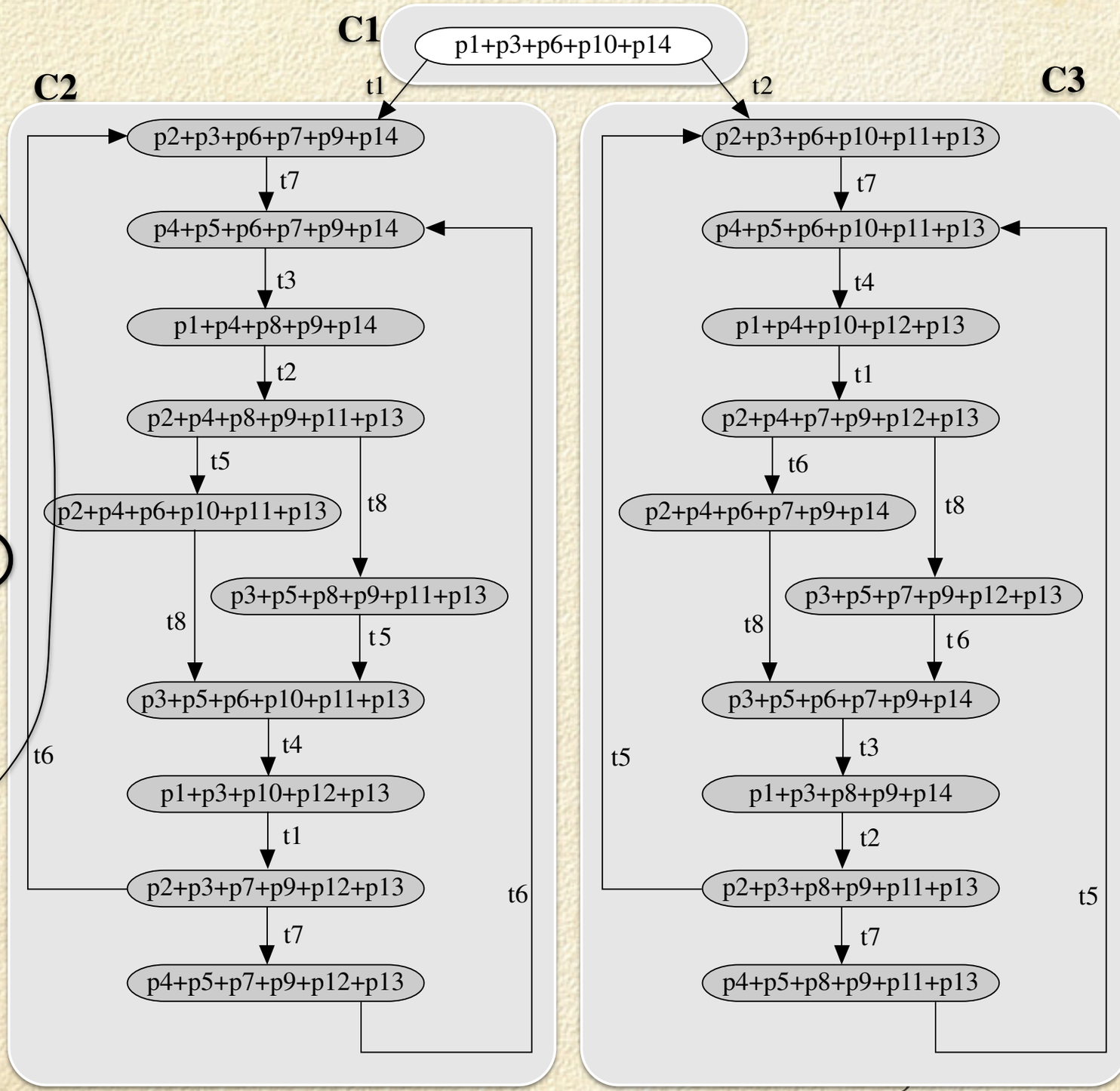
lineare Algebra-Methoden (P-Invarianten, T-Invarianten)

graphenbasierte Methoden

Definition 3.1 *Der* Erreichbarkeitsgraph *eines Netzsystems* $\mathcal{S} = \langle \mathcal{N}, \mathbf{m_0} \rangle$ *ist ein gerichteter Graph* $\mathrm{RG}(\mathcal{S}) = (V, E)$, *mit Knotenmenge* $V = \mathbf{R}(\mathcal{S})$ *und Kantenmenge* $E = \{\langle \mathbf{m}, t, \mathbf{m}' \rangle | \mathbf{m}, \mathbf{m}' \in \mathbf{R}(\mathcal{S}) \text{ und } \mathbf{m} \xrightarrow{t} \mathbf{m}'\}$.



b)



Algorithmus 3.1 (Berechnung des Erreichbarkeitsgraphen)

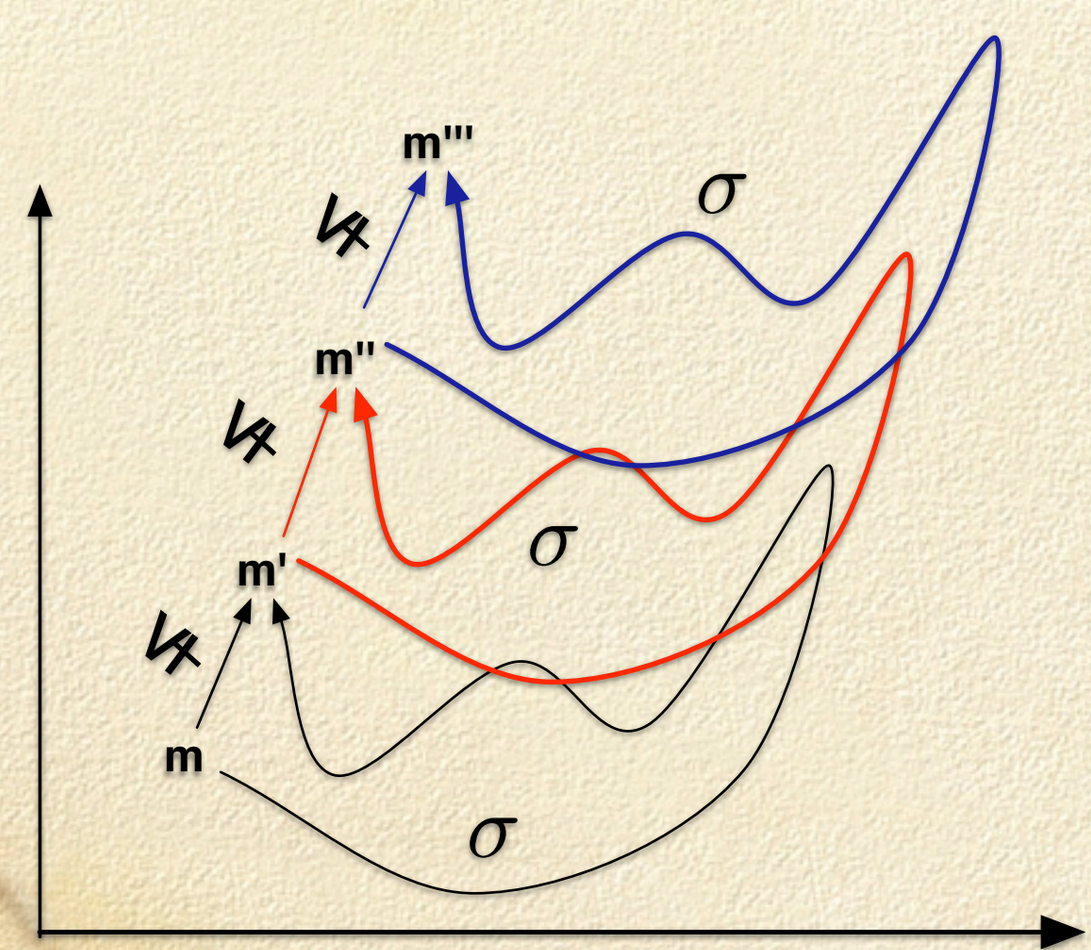
Input - Das Netzsystem $\mathcal{S} = \langle \mathcal{N}, \mathbf{m_0} \rangle$
Output - Der gerichtete Graph $\mathrm{RG}(\mathcal{S}) = (V, E)$, falls das Netzsystem beschränkt ist.

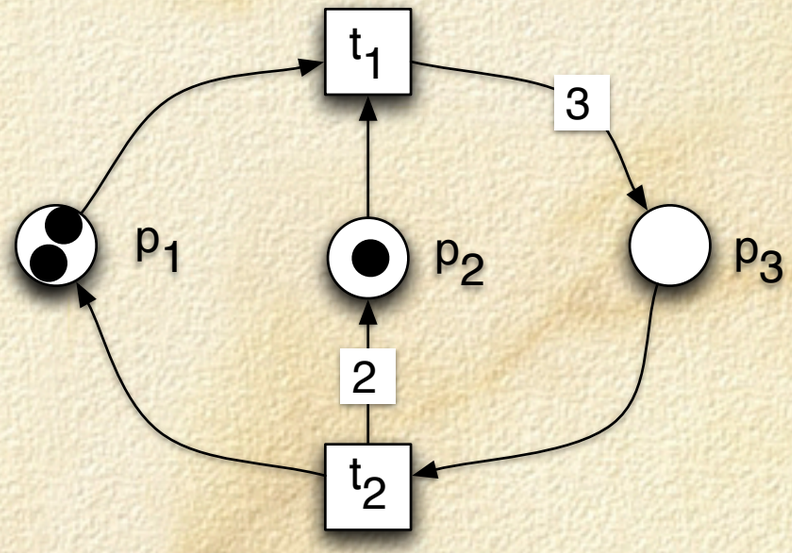
1. Initialisiere $\mathrm{RG}(\mathcal{S}) = (\{\mathbf{m_0}\}, \emptyset)$; $\mathbf{m_0}$ sei ungefärbt;
2. **while** Es gibt ungefärbte Knoten in V. **do**
   - 2.1 Wähle einen ungefärbte Knoten $\mathbf{m} \in V$ und färbe ihn.
   - 2.2 **for** Für jede in $\mathbf{m}$ aktivierte Transition t **do**
     - 2.2.1 Berechne $\mathbf{m}'$ mit $\mathbf{m} \xrightarrow{t} \mathbf{m}'$;
     - 2.2.2 **if** Es gibt einen Knoten $\mathbf{m}'' \in V$ derart, dass $\mathbf{m}'' \xrightarrow{\sigma} \mathbf{m}'$ und $\mathbf{m}'' < \mathbf{m}'$.
       then Der Algorithmus terminiert ohne Ergebnis.;
       (Das Netzsystem ist unbeschränkt.)
     - 2.2.3 **if** Es gibt keinen Knoten $\mathbf{m}'' \in V$ derart, dass $\mathbf{m}'' = \mathbf{m}'$
       then $V := V \cup \{\mathbf{m}'\}$, wobei $\mathbf{m}'$ ein ungefärbter Knoten sei.
     - 2.2.4 $E := E \cup \{\langle \mathbf{m}, t, \mathbf{m}' \rangle\}$
3. Der Algorithmus terminiert mit Ergebnis. ($\mathrm{RG}(\mathcal{S})$ ist der Erreichbarkeitsgraph.)

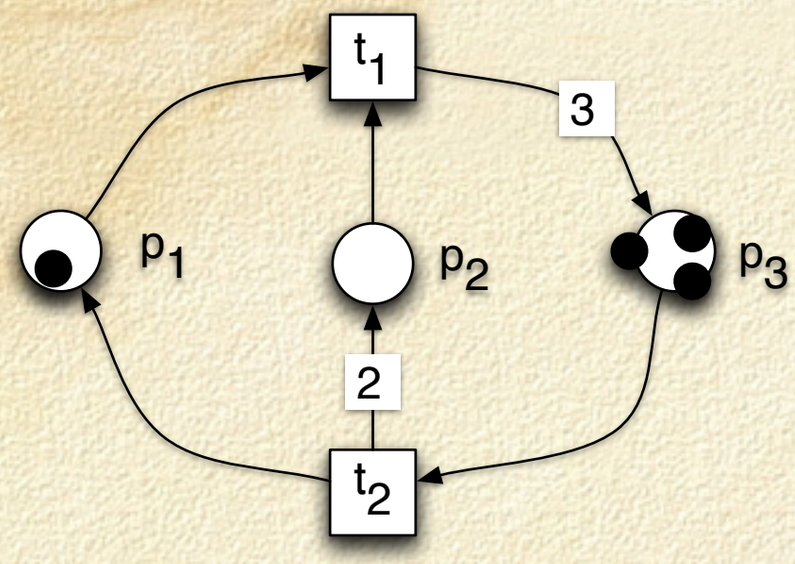
Als Abbruchkriterium dient: das System $\mathcal{S} = \langle \mathcal{N}, \mathbf{m_0} \rangle$ ist genau dann unbeschränkt, wenn es zwei erreichbare Markierungen $\mathbf{m}, \mathbf{m}' \in \mathrm{RS}(\mathcal{S})$ gibt, die folgende Bedingungen erfüllen:

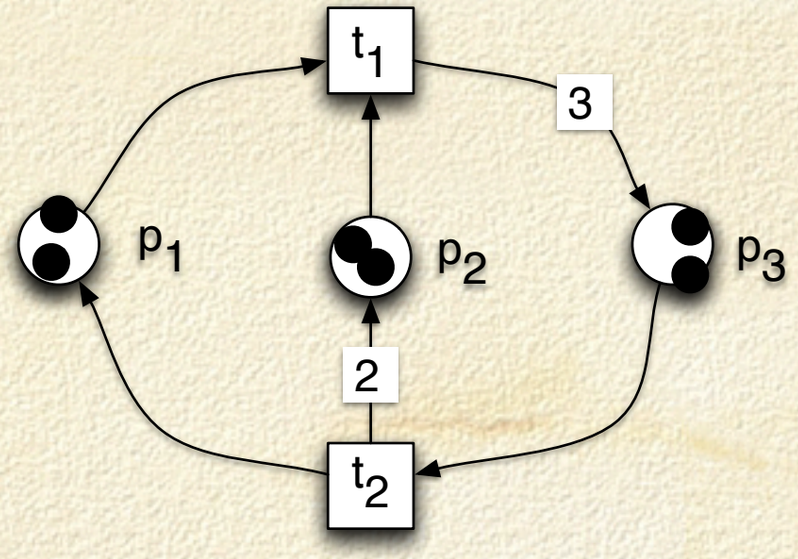
a) $\exists \sigma \in T^* : \mathbf{m} \xrightarrow{\sigma} \mathbf{m}'$

b) $\mathbf{m} \lneq \mathbf{m}'$









$$\begin{pmatrix}2\\1\\0\end{pmatrix} \xrightarrow{t_1} \begin{pmatrix}1\\0\\3\end{pmatrix} \xrightarrow{t_2} \not\exists$$

$$\begin{pmatrix}2\\2\\2\end{pmatrix} \xrightarrow{t_1} \begin{pmatrix}1\\1\\4\end{pmatrix} \xrightarrow{t_2} \not\exists$$

Algorithmus 3.1 (Berechnung des Erreichbarkeitsgraphen)

Input - Das Netzsystem $\mathcal{S} = \langle \mathcal{N}, \mathbf{m_0} \rangle$
Output - Der gerichtete Graph $\mathrm{RG}(\mathcal{S}) = (V, E)$, falls das Netzsystem beschränkt ist.

1. Initialisiere $\mathrm{RG}(\mathcal{S}) = (\{\mathbf{m_0}\}, \emptyset)$; $\mathbf{m_0}$ sei ungefärbt;
2. **while** Es gibt ungefärbte Knoten in V. **do**
   - 2.1 Wähle einen ungefärbte Knoten $\mathbf{m} \in V$ und färbe ihn.
   - 2.2 **for** Für jede in $\mathbf{m}$ aktivierte Transition t **do**
     - 2.2.1 Berechne $\mathbf{m}'$ mit $\mathbf{m} \xrightarrow{t} \mathbf{m}'$;
     - 2.2.2 **if** Es gibt einen Knoten $\mathbf{m}'' \in V$ derart, dass $\mathbf{m}'' \xrightarrow{\sigma} \mathbf{m}'$ und $\mathbf{m}'' < \mathbf{m}'$.
       then Der Algorithmus terminiert ohne Ergebnis.;
       (Das Netzsystem ist unbeschränkt.)
     - 2.2.3 **if** Es gibt keinen Knoten $\mathbf{m}'' \in V$ derart, dass $\mathbf{m}'' = \mathbf{m}'$
       then $V := V \cup \{\mathbf{m}'\}$, wobei $\mathbf{m}'$ ein ungefärbter Knoten sei.
     - 2.2.4 $E := E \cup \{\langle \mathbf{m}, t, \mathbf{m}' \rangle\}$
3. Der Algorithmus terminiert mit Ergebnis. ($\mathrm{RG}(\mathcal{S})$ ist der Erreichbarkeitsgraph.)

Invarianz-Eigenschaften

a) Markierungs-Invarianz

b) Lebendigkeits-Invarianz

Spezifikationssprache durch:

Markierungsprädikate Π

Alle aussagenlogischen Formeln mit Atomen der Form:

$$\sum_{p \in A} k_p \mathbf{m}[p] \leq k$$

wobei k_p und k rationale Konstanten und A eine Teilmenge der Plätze ist.

Definition 3.2 *Eine* Markierungs-Invarianzeigenschaft *(marking invariance property) eines Netzsystems* $\mathcal{S} = \langle \mathcal{N}, \mathbf{m_0} \rangle$ *ist ein Prädikat der Form* $\forall \mathbf{m} \in \mathrm{RS}(\mathcal{S}) \,.\, \Pi(\mathbf{m})$ *oder* $\forall \mathbf{m} \in \mathrm{RS}(\mathcal{S}) \,.\, \exists t \in T \,.\, \Pi(\mathbf{m})$, *wobei* Π *ein Markierungsprädikat ist.*

Beispiele dazu sind:

$\sum_{p \in A} k_p \mathbf{m}[p] \leq k$

1) *k-Beschränktheit* (k-boundedness) eines Platzes p: $\forall \mathbf{m} \in \mathrm{RS}(\mathcal{S}) \,.\, \mathbf{m}[p] \leq k$.

2) *Markierungs-Ausschluss* (marking mutual exclusion) zwischen p und p': $\forall \mathbf{m} \in \mathrm{RS}(\mathcal{S}) \,.\, ((\mathbf{m}[p] = 0) \vee (\mathbf{m}[p'] = 0))$.

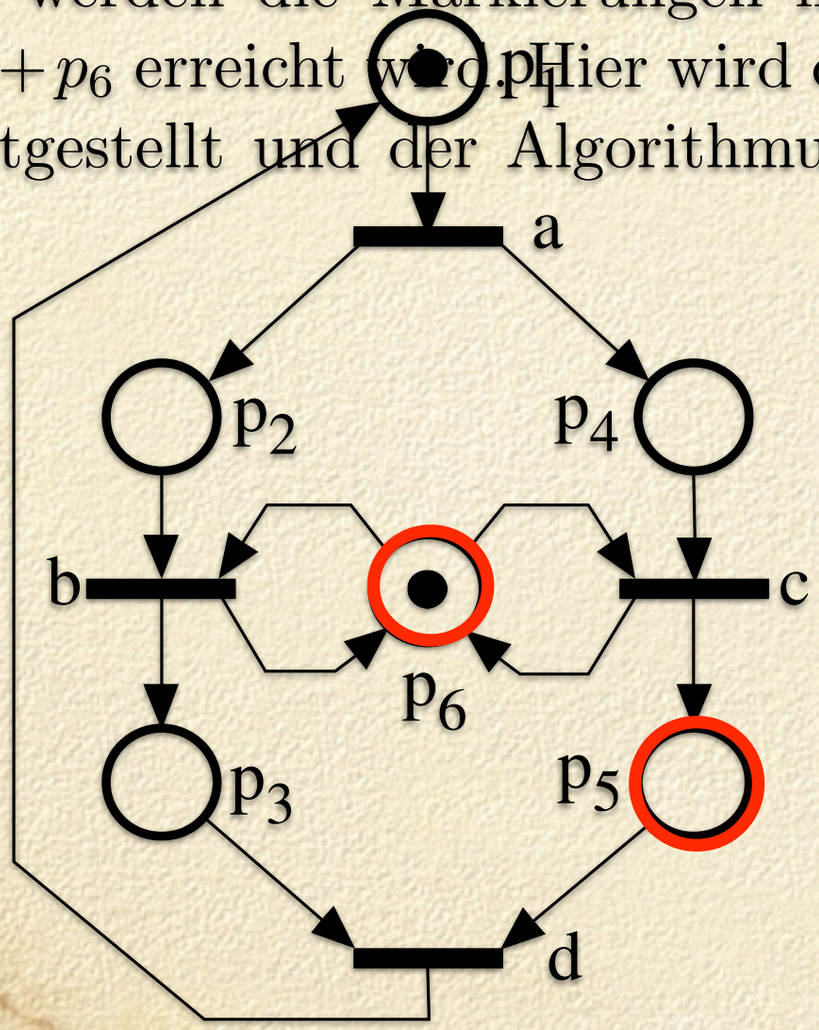
3) *Verklemmungsfreiheit* (deadlock-freeness): $\forall \mathbf{m} \in \mathrm{RS}(\mathcal{S}) \,.\, \exists t \in T \,.\, W(\bullet, t) \leq \mathbf{m}$.

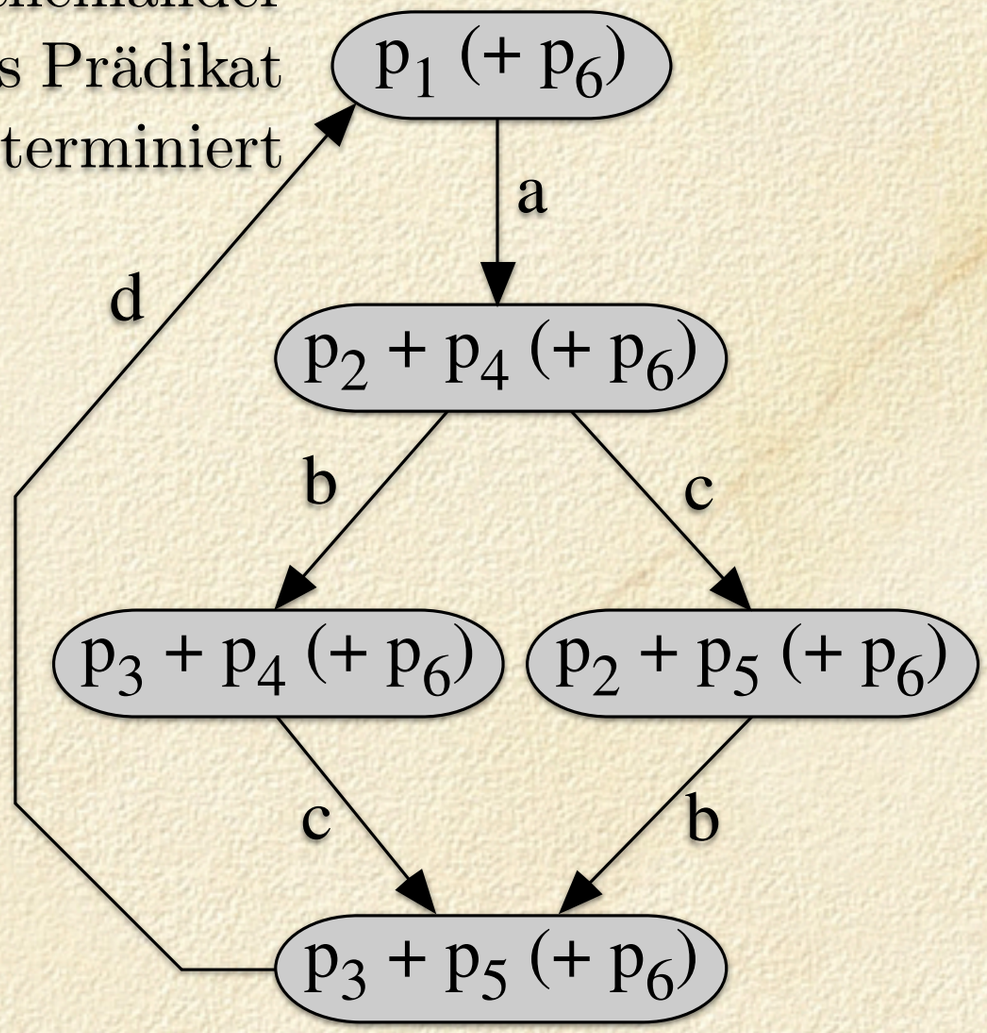
Algorithmus 2.4 (Entscheiden einer Markierungs-Invarianzeigenschaft)

Input - Der Erreichbarkeitsgraph $\mathrm{RG}(\mathcal{N}, \mathbf{m_0})$. Die Markierungs-Invarianzeigenschaft Π.
Output - TRUE falls die Eigenschaft Π erfüllt ist; FALSE falls die Eigenschaft Π nicht erfüllt ist.

1. Initialisiere alle Elemente von $\mathrm{RS}(\mathcal{S})$ als ungefärbt.
2. **while** Es gibt einen ungefärbten Knoten $\mathbf{m} \in \mathrm{RS}(\mathcal{S})$ **do**
   2.1 Wähle einen einen ungefärbten Knoten $\mathbf{m} \in \mathrm{RS}(\mathcal{S})$ und färbe ihn.
   2.2 **if** $\mathbf{m}$ erfüllt nicht Π.
   then return FALSE (Die Eigenschaft Π ist nicht erfüllt.)
3. Return TRUE

Beispiel 2.4 Analyse von Markierungs-Invarianzeigenschaften Betrachte das Netzsystem in Abb. 5.2, für das $\mathrm{RS}(\mathcal{S}) = \{p_1+p_6, p_2+p_4+p_6, p_3+p_4+p_6, p_2+p_5+p_6, p_3+p_5+p_6\}$ gilt. Die Anwendung des Algorithmus 5.2 zur Überprüfung des Markierungsausschlusses der Plätze p_5 und p_6 (d.h. $\Pi(\mathbf{m}) = (\mathbf{m}[p_5] = 0) \vee (\mathbf{m}[p_6] = 0)$) beginnt damit, dass alle Elemente von $\mathrm{RS}(\mathcal{S})$ ungefärbt sind (Schritt 1). Dann werden die Markierungen nacheinander geprüft bis $p_2+p_5+p_6$ erreicht wird. Hier wird das Prädikat Π als ungültig festgestellt und der Algorithmus terminiert mit FALSE.





Definition 3.4 *Eine* Lebendigkeits-Invarianzeigenschaft *(liveness invariance property) eines Netzsystems* $\mathcal{S} = \langle \mathcal{N}, \mathbf{m_0} \rangle$ *ist ein Prädikat der Form* $\forall \mathbf{m} \in \mathrm{RS}(\mathcal{S}) \,.\, \exists \mathbf{m}' \in \mathrm{RS}(\mathcal{N}, \mathbf{m}) \,.\, \Pi(\mathbf{m}')$, *wobei* Π *ein Markierungsprädikat ist.*

Beispiele dazu sind:

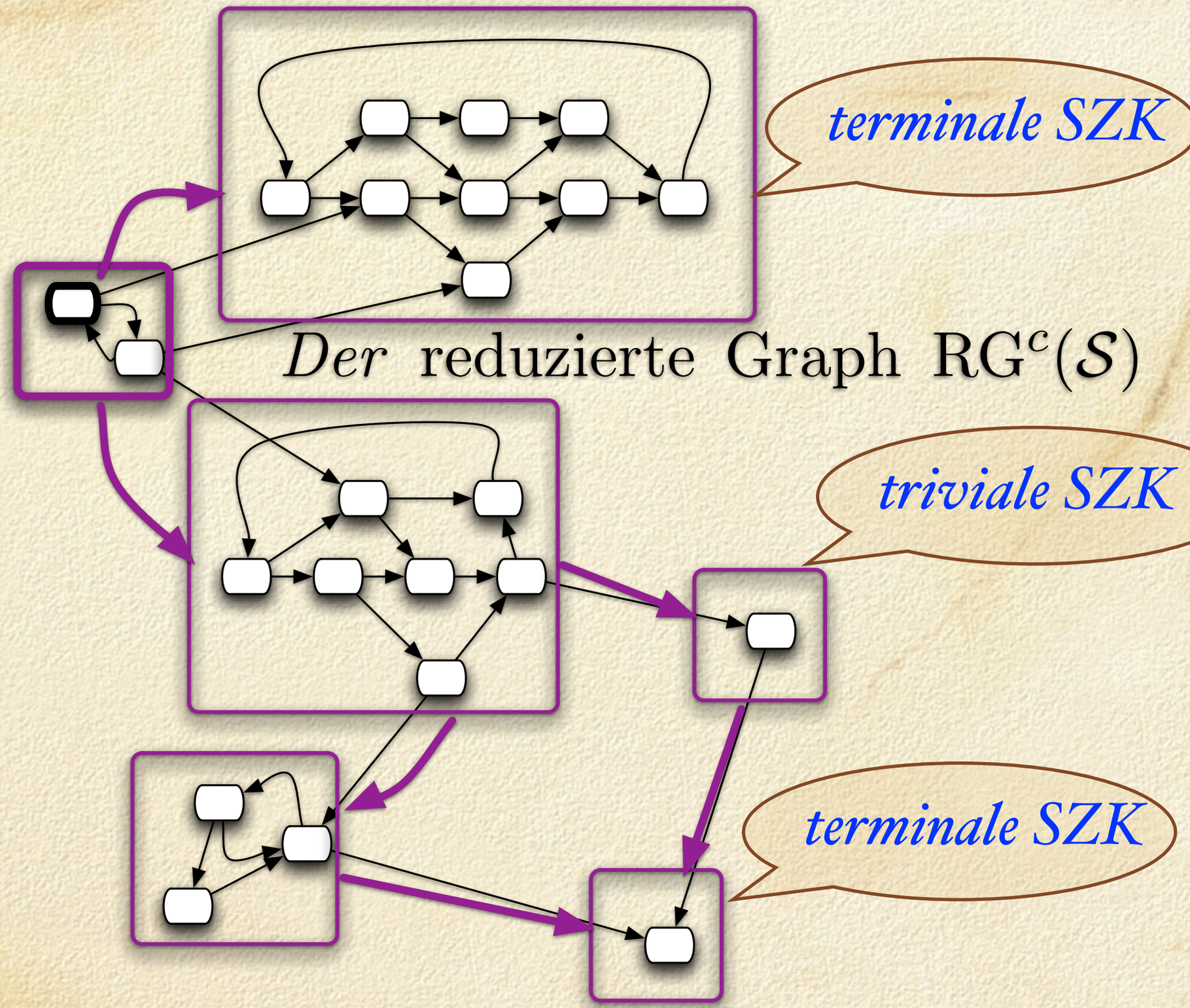
1) *Lebendigkeit von* t (liveness of t): $\forall \mathbf{m} \in \mathrm{RS}(\mathcal{S}) \,.\, \exists \mathbf{m}' \in \mathrm{RS}(\mathcal{N}, \mathbf{m}) \,.\, W(\bullet, t) \leq \mathbf{m}'$.

2) $\mathbf{m}_H$ ist *Rücksetzzustand* (home state): $\forall \mathbf{m} \in \mathrm{RS}(\mathcal{S}) \,.\, \exists \mathbf{m}' \in \mathrm{RS}(\mathcal{N}, \mathbf{m}) \,.\, \mathbf{m}' = \mathbf{m}_H$.

3) *Reversibilität* (reversibility): $\forall \mathbf{m} \in \mathrm{RS}(\mathcal{S}) \,.\, \exists \mathbf{m}' \in \mathrm{RS}(\mathcal{N}, \mathbf{m}) \,.\, \mathbf{m}' = \mathbf{m_0}$.

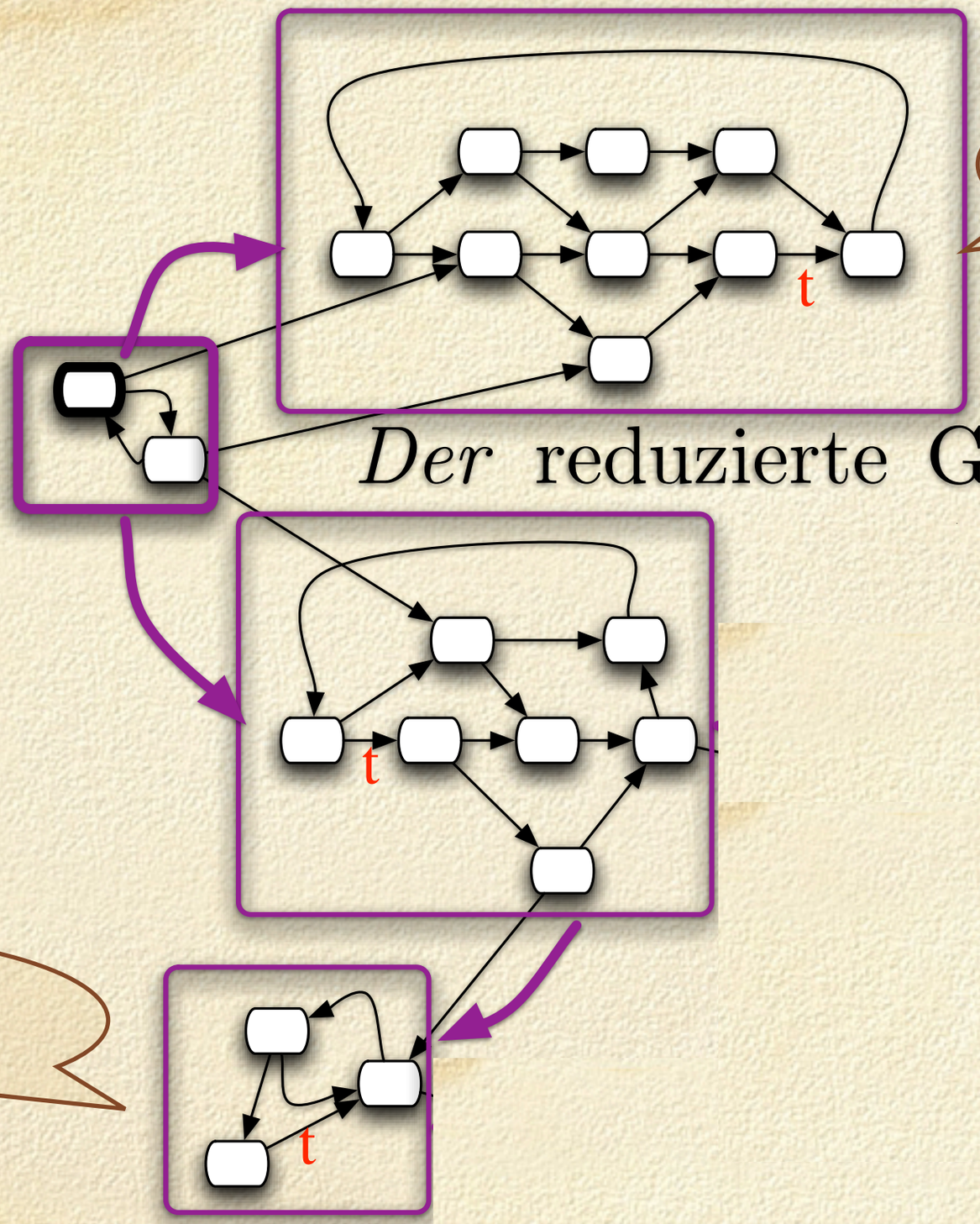
Definition 3.5 *Ein* Pfad *von* $\mathbf{m}_1$ *nach* $\mathbf{m}_k$ *in einem Erreichbarkeitsgraphen* $\mathrm{RG}(\mathcal{S}) = (V, E)$ *ist eine Folge* $\mathbf{m}_1 \ldots \mathbf{m}_i \mathbf{m}_{i+1} \ldots \mathbf{m}_k$ $(k \geq 2)$ *seiner Knoten mit* $\langle \mathbf{m}_i, t_i, \mathbf{m}_{i+1} \rangle \in E$ *für alle* $i \in \{1, \ldots, k-1\}$ *und jeweils ein* $t_i \in T$. *Ein Teilgraph von* $\mathrm{RG}(\mathcal{S})$ *heißt* streng zusammenhängend *(strongly connected) falls er entweder nur aus einem Knoten besteht oder zwischen je zwei verschiedenen seiner Knoten ein Pfad existiert. Seine Knotenmenge heißt* Zusammenhangskomponente(ZK). *Eine maximale ZK heißt* strenge Zusammenhangskomponente (SZK) *(strongly connected component) von* $\mathrm{RG}(\mathcal{S})$. *Sie heißt* triviale Zusammenhangskomponente *falls sie nur aus einem Knoten ohne Schleife besteht und* terminale strenge Zusammenhangskomponente, *falls von keinem ihrer Knoten eine Kante zu einem Knoten* $\mathbf{m}$ *ausgeht, der nicht in ihr liegt.*

strenge Zusammenhangskomponente (SZK)



Der reduzierte Graph $\mathrm{RG}^c(\mathcal{S})$

strenge Zusammenhangskomponente (SZK)



Der reduzierte Graph $\mathrm{RG}^c(\mathcal{S})$

Ist t *lebendig?*

terminale SZK

Definition 3.6 *Der* reduzierte Graph $\mathrm{RG}^c(\mathcal{S}) = (V_c, E_c)$ *eines Erreichbarkeitsgraphen* $\mathrm{RG}(\mathcal{S}) = (V, E)$ *hat als Knotenmenge die SZKs von* $\mathrm{RG}(\mathcal{S})$, *d.h.* $V_c := \{C_1, \ldots, C_r\}$.

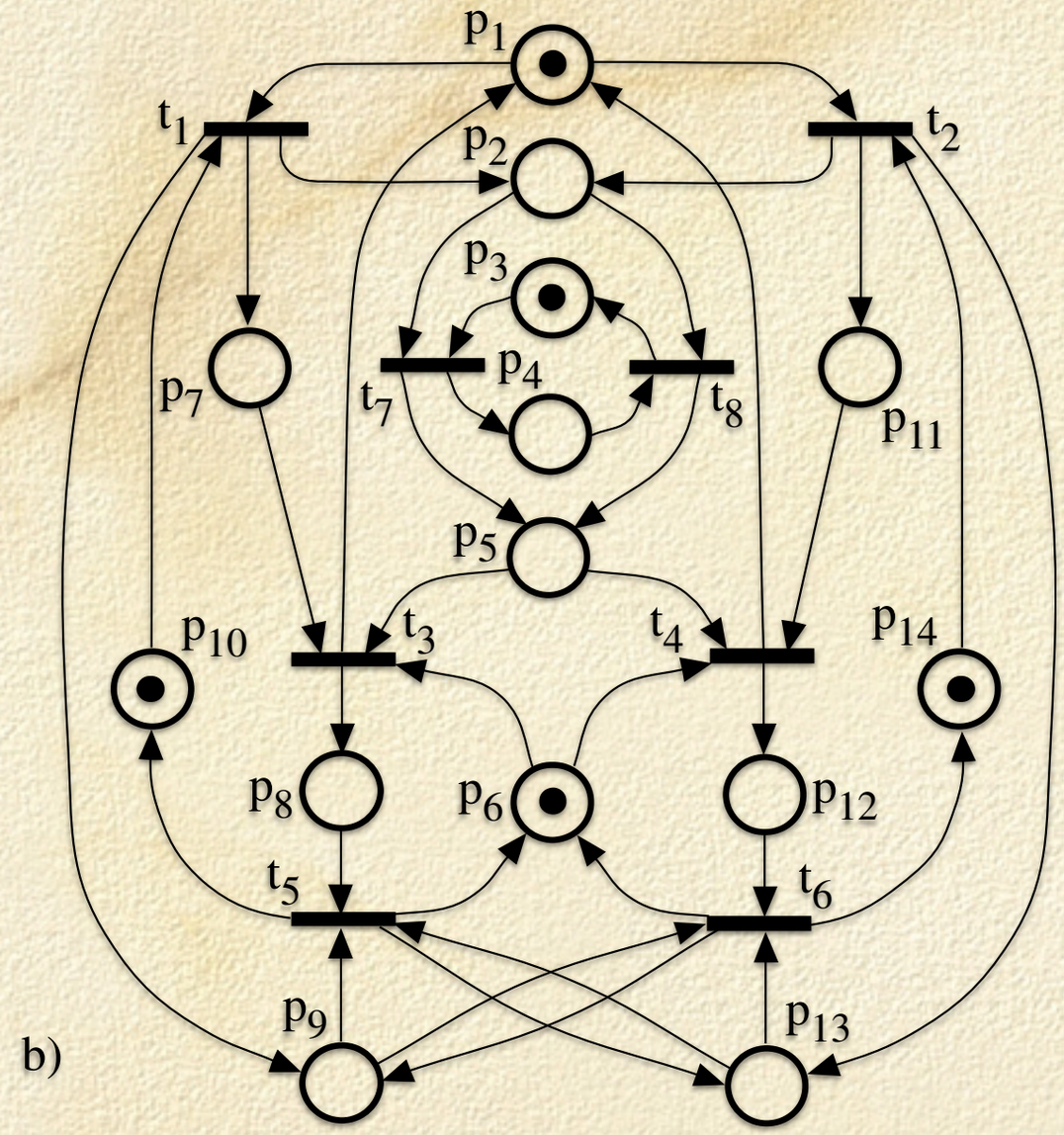
Die Kantenmenge ist $E_c := \{\langle C_i, t, C_j\rangle |$ *Es gibt eine Kante* $\langle \mathbf{m}_i, t, \mathbf{m}'_j\rangle \in E$ *derart, dass* $\mathbf{m}$ *bzw.* $\mathbf{m}'$ *in SZKs* C_i *bzw.* C_j *liegen und* $C_i \neq C_j$ *gilt.*$\}$

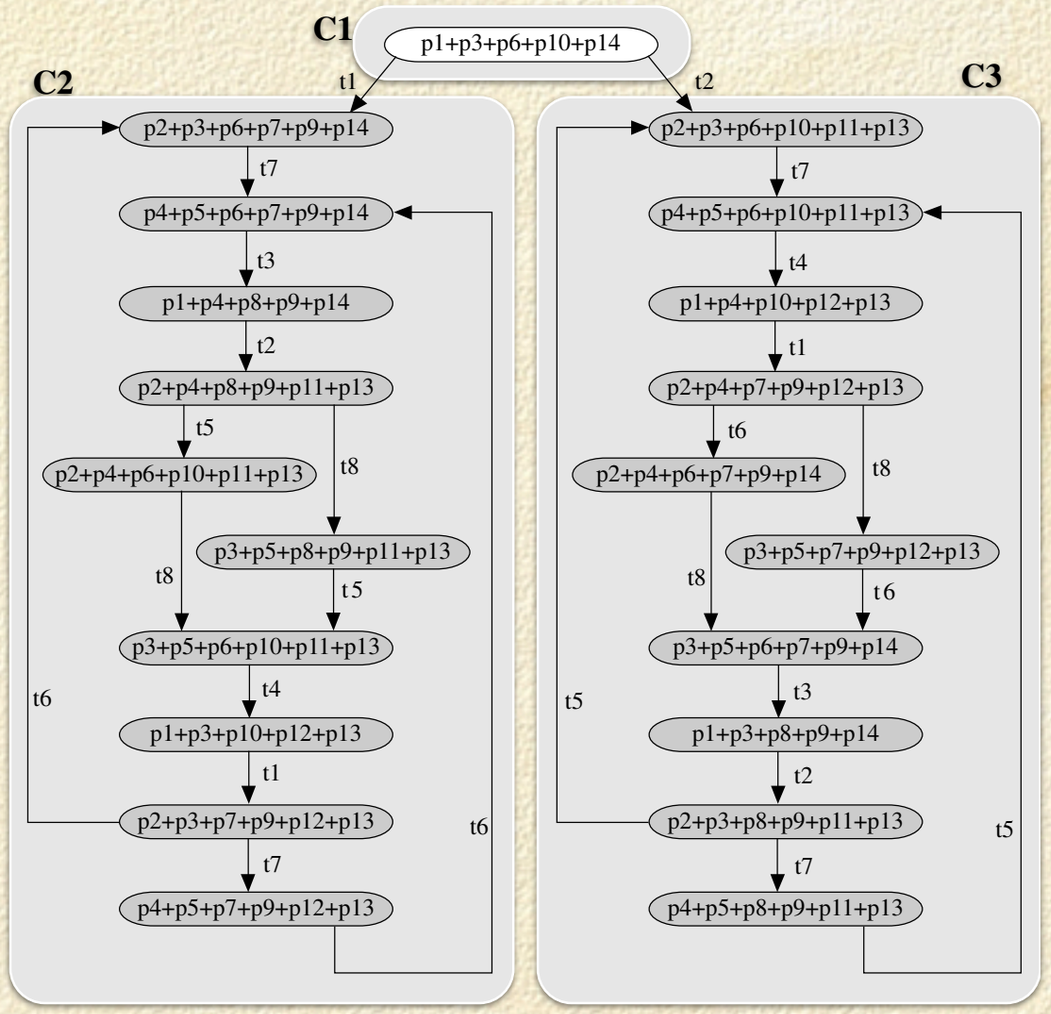
Algorithmus 3.3 (Entscheiden einer Lebendigkeits-Invarianzeigenschaft)

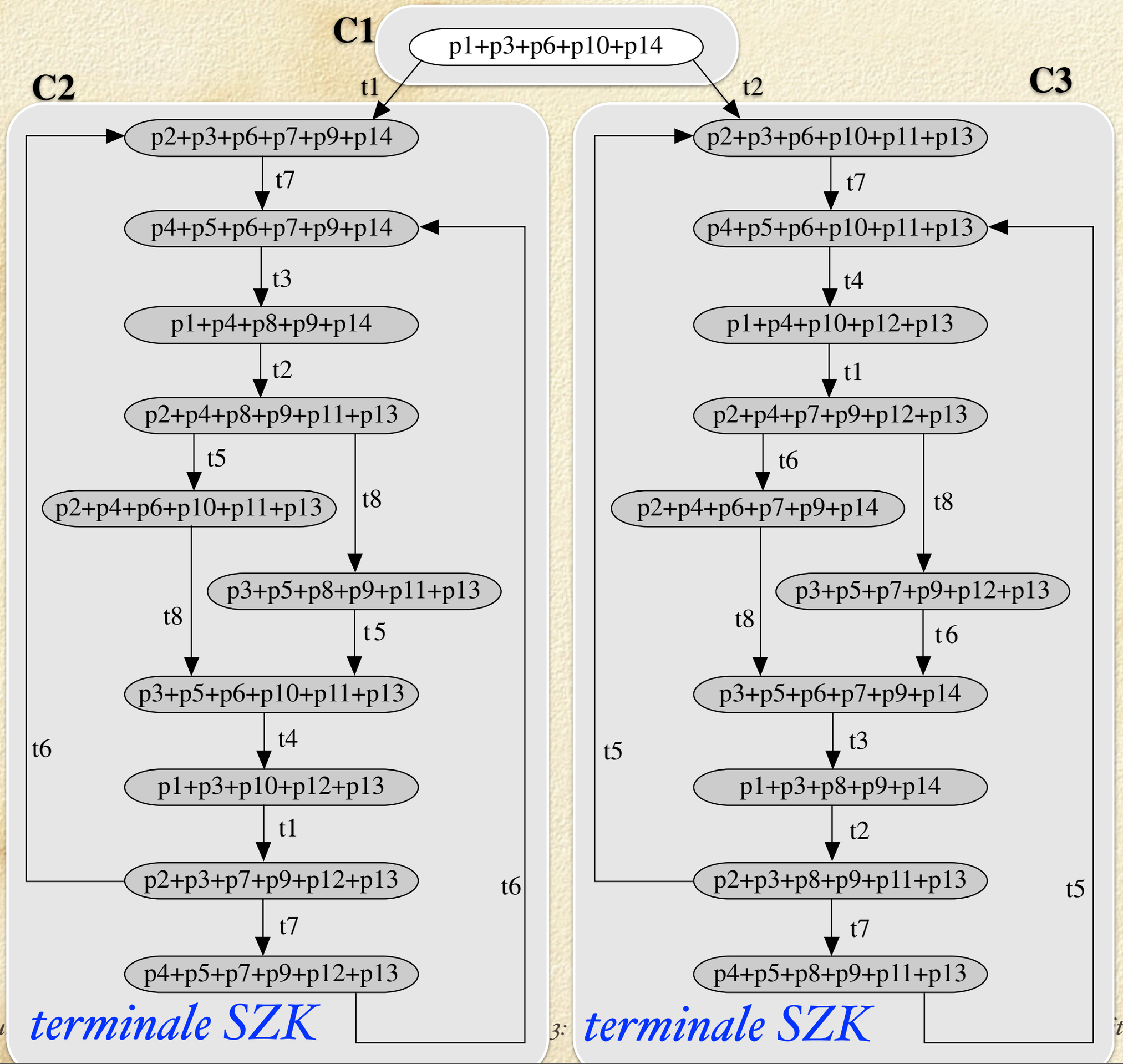
Input - Der Erreichbarkeitsgraph $\mathrm{RG}(\mathcal{N}, \mathbf{m_0})$. Die Lebendigkeits-Invarianzeigenschaft Π.
Output - TRUE falls die Eigenschaft Π erfüllt ist; FALSE falls die Eigenschaft Π nicht erfüllt ist.

1. Berechne die strengen Zusammenhangskomponenten (SZKs) $C_1, \ldots, C_r$ von $\mathrm{RG}(\mathcal{N}, \mathbf{m_0})$.
2. Berechne den Graphen $\mathrm{RG}^c(\mathcal{S}) = (V_c, E_c)$ durch Wandeln der SZKs $C_1, \ldots, C_r$ in jeweils einen Kno
   d.h. $V_c = \{C_1, \ldots, C_r\}$. $\langle C_i, t, C_j \rangle \in E_c$ genau dann, wenn eine Kante $\langle \mathbf{m}, t, \mathbf{m}' \rangle \in E$
   derart existiert, dass $\mathbf{m}$ in der SZK C_i und $\mathbf{m}'$ in der SZK C_j liegt und $i \neq j$ gilt.
3. Berechne die Menge F der terminalen SZKs von $\mathrm{RG}^c(\mathcal{S})$.
4. **while** es gibt $C_i \in F$ **do**
   4.1 **if** C_i enthält keine Markierung $\mathbf{m}'$, die Π erfüllt.
   then return FALSE
   4.2 Entferne C_i aus F
5. Return TRUE

Beispiel 3.7 Analysis einer Lebendigkeits-Invarianzeigenschaft Wir betrachten das Netzsystem von Abb.5.3 b) mit dem Erreichbarkeitsgraphen von Abb. 5.8. Um den Algorithmus 5.3 auszuführen, müssen zunächst die SZKs berechnet werden. Diese sind schon in Abb. 5.8 durch die Bereiche C_1, C_2 und C_3 eingezeichnet, wobei C_2 und C_3 terminal sind. Wir prüfen die Lebendigkeitseigenschaft $(\forall \mathbf{m} \in \mathrm{RS}(\mathcal{S}) \,.\, \forall t \in T \,.\, [\exists \mathbf{m}^t \in \mathrm{RS}(\langle \mathcal{N}, \mathbf{m} \rangle) \,.\, \mathbf{m}^t \geq W(\bullet, t)])$.


p1
t1
t2
p2
p3
p4
p7
t7
t8
p11
p5
p10
t3
t4
p14
p8
p6
p12
t5
t6
p9
p13
b)


C1
p1+p3+p6+p10+p14
t1
t2
C2
C3
p2+p3+p6+p7+p9+p14
t7
p4+p5+p6+p7+p9+p14
t3
p1+p4+p8+p9+p14
t2
p2+p4+p8+p9+p11+p13
t5
t8
p2+p4+p6+p10+p11+p13
p3+p5+p8+p9+p11+p13
t8
t5
p3+p5+p6+p10+p11+p13
t6
t4
p1+p3+p10+p12+p13
t1
p2+p3+p7+p9+p12+p13
t6
t7
p4+p5+p7+p9+p12+p13
p2+p3+p6+p10+p11+p13
t7
p4+p5+p6+p10+p11+p13
t4
p1+p4+p10+p12+p13
t1
p2+p4+p7+p9+p12+p13
t6
t8
p2+p4+p6+p7+p9+p14
p3+p5+p7+p9+p12+p13
t8
t6
p3+p5+p6+p7+p9+p14
t5
t3
p1+p3+p8+p9+p14
t2
p2+p3+p8+p9+p11+p13
t5
t7
p4+p5+p8+p9+p11+p13


C1
p1+p3+p6+p10+p14
t1
t2
C2
C3
p2+p3+p6+p7+p9+p14
t7
p4+p5+p6+p7+p9+p14
t3
p1+p4+p8+p9+p14
t2
p2+p4+p8+p9+p11+p13
t5
t8
p2+p4+p6+p10+p11+p13
p3+p5+p8+p9+p11+p13
t8
t5
p3+p5+p6+p10+p11+p13
t4
t6
p1+p3+p10+p12+p13
t1
p2+p3+p7+p9+p12+p13
t6
t7
p4+p5+p7+p9+p12+p13
terminale SZK
p2+p3+p6+p10+p11+p13
t7
p4+p5+p6+p10+p11+p13
t4
p1+p4+p10+p12+p13
t1
p2+p4+p7+p9+p12+p13
t6
t8
p2+p4+p6+p7+p9+p14
p3+p5+p7+p9+p12+p13
t8
t6
p3+p5+p6+p7+p9+p14
t5
t3
p1+p3+p8+p9+p14
t2
p2+p3+p8+p9+p11+p13
t5
t7
p4+p5+p8+p9+p11+p13
terminale SZK