Geschäftsprozess-Management

Prof. Dr.-Ing. Andreas Ittner

Email: ittner@hs-mittweida.de

WWW: www.andreas-ittner.de

Tel.: +49(0)3727-58-1288

Mob.: +49(0)177-5555-347

Fax.: +49(0)180-35518-22467

Gliederung

HOCHSCHULE MITTWEIDA UNIVERSITY OF APPLIED SCIENCES

- Motivation
- Prozesse und Prozess-Management
  - Geschäftsprozesse, Workflow-Prozesse
  - Prozessdesign, Prozessverbesserungen
- Prozess-Modellierung
  - Zweck, Modellierungselemente und –sprachen
  - Petri-Netze, EPKs, BPMN, …
- Prozess-Analyse
  - Struktur-, Verhaltens-, Erreichbarkeits- und Performance-Analysen
  - Simulation
- Workflow-Management-Systeme
  - Historie, Infrastruktur, Implementierungen, Standards

Prozess-Analyse

HOCHSCHULE MITTWEIDA UNIVERSITY OF APPLIED SCIENCES

1. Fragestellungen und klassische Fehler,
2. Struktur-Elemente und -Eigenschaften (Zusammenhängend, Free Choice, Workflow-Netz, well-handled, -structured, S-Komponenten / s-coverable),
3. Dynamische Elemente und Eigenschaften (Anfangsmarkierung, Beschränktheit, Komplementbildung, tot, lebendig, verklemmungsfrei, Soundness, Invarianten (S-, T-Invarianten)),
4. Erreichbarkeitsanalyse,
5. Linear Algebraische Darstellung,
6. Zusammenhänge bei der Prozess-Analyse.

Prozess-Analyse

HOCHSCHULE MITTWEIDA
UNIVERSITY OF APPLIED SCIENCES

1. Fragestellungen und klassische Fehler,
2. Struktur-Elemente und -Eigenschaften (Zusammenhängend, Free Choice, Workflow-Netz, well-handled, -structured, S-Komponenten / s-coverable),
3. Dynamische Elemente und Eigenschaften (Anfangsmarkierung, Beschränktheit, Komplementbildung, tot, lebendig, verklemmungsfrei, Soundness, Invarianten (S-, T-Invarianten)),
4. Erreichbarkeitsanalyse,
5. Linear Algebraische Darstellung,
6. Zusammenhänge bei der Prozess-Analyse.

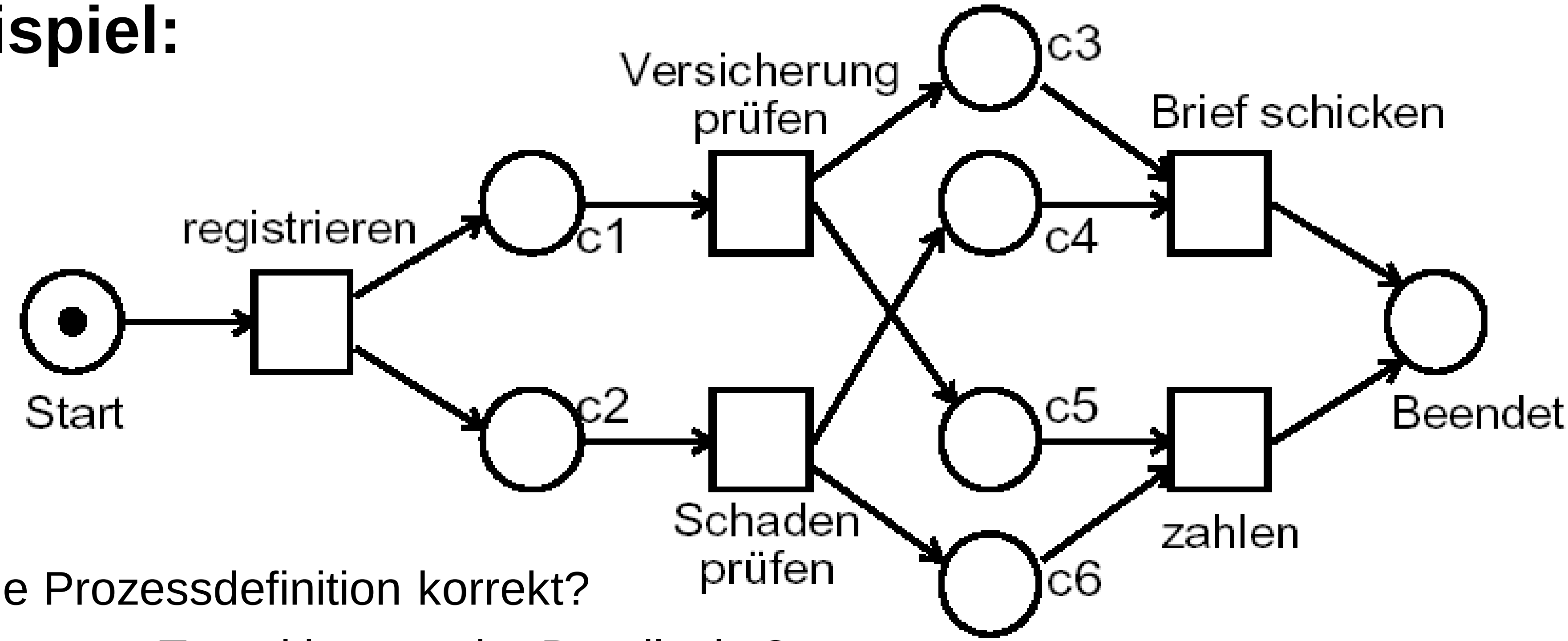
Qualitative Fragestellungen

- Kann es zu Deadlocks (toten Markierungen) kommen?
- Können alle Arten von Cases (Fällen) erfolgreich behandelt werden?
- Werden alle Cases irgendwann beendet?
- Können Tasks (Aufgaben) parallel ausgeführt werden?

Quantitative Fragestellungen

- Wie viele Cases pro Stunde können bearbeitet werden?
- Was ist die durchschnittliche Durchlaufzeit?
- Wie viele Ressourcen werden pro Case benötigt?
- Was ist die durchschnittliche Wartezeit?

Beispiel:



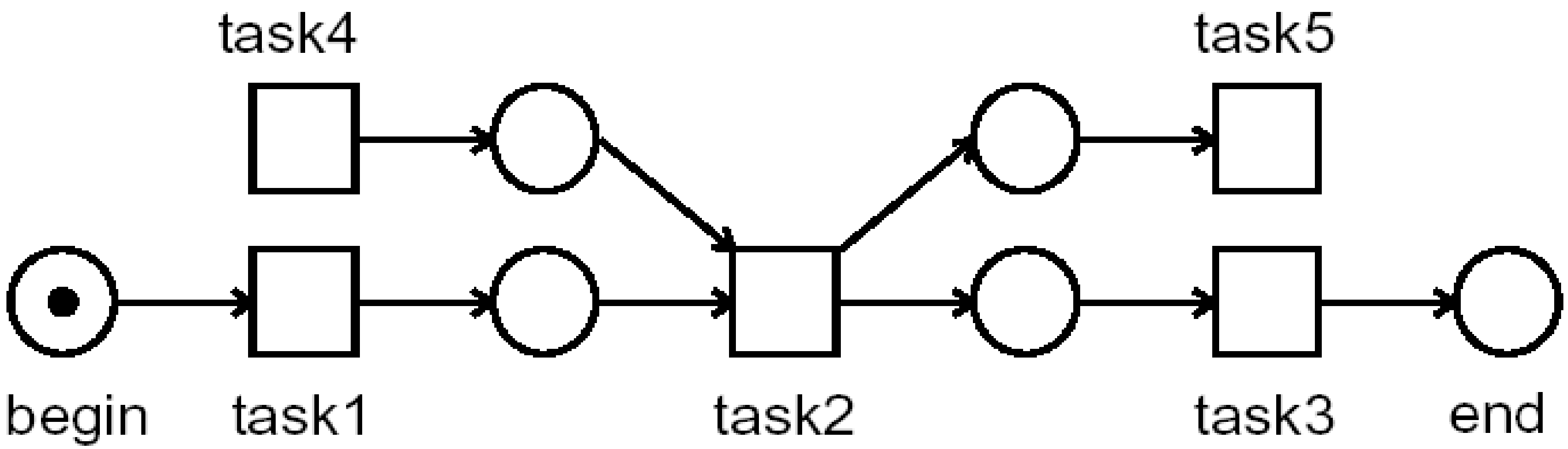
Ist die Prozessdefinition korrekt?

Gibt es tote Transitionen oder Deadlocks?

Ist das Netz k-beschränkt?

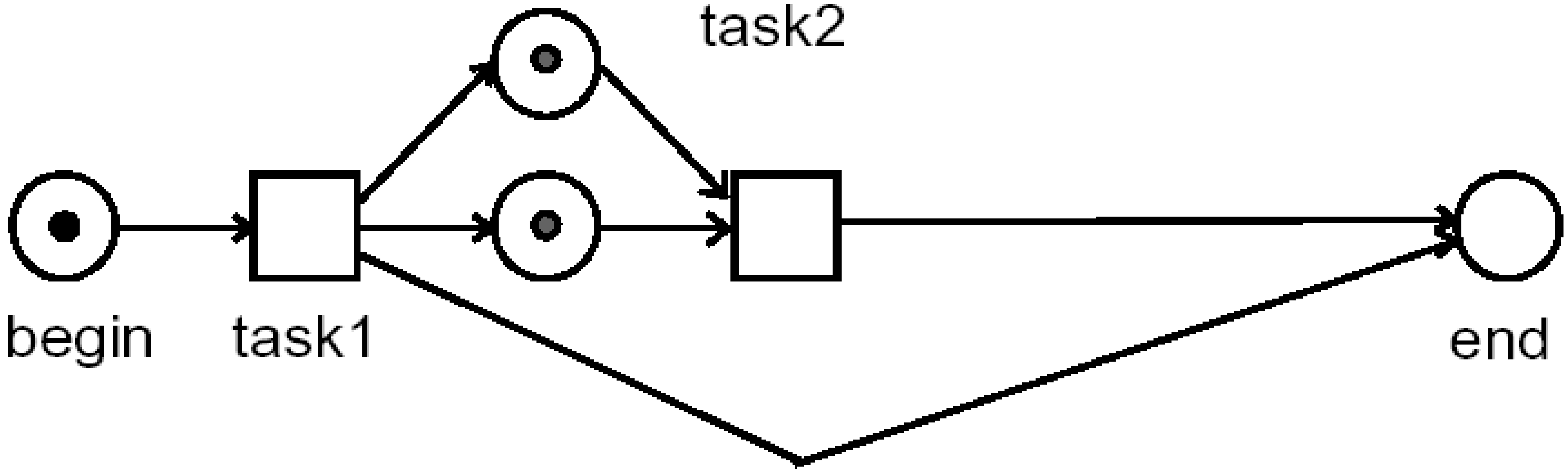
HOCHSCHULE MITTWEIDA UNIVERSITY OF APPLIED SCIENCES

Beispiel: „baumelnde“ Tasks:

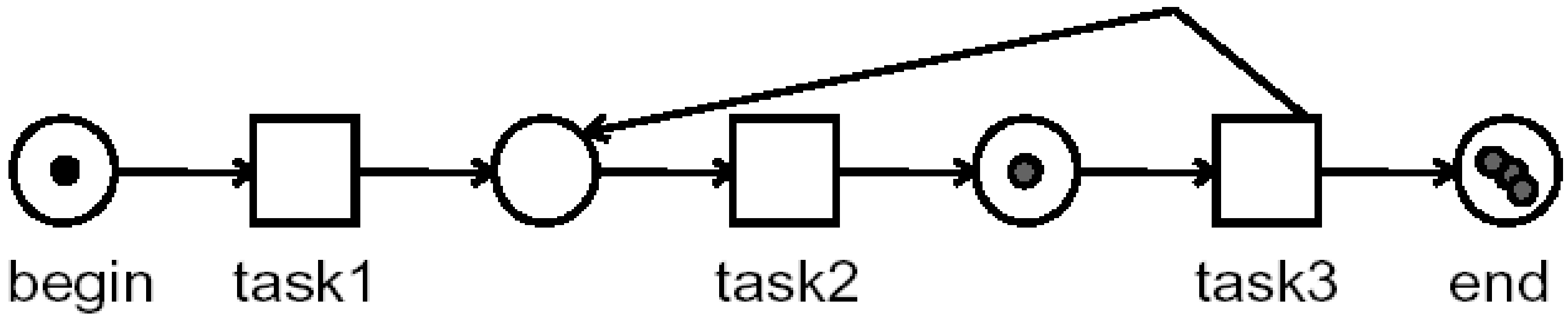


HOCHSCHULE MITTWEIDA UNIVERSITY OF APPLIED SCIENCES

Beispiel: mögliche tote Markierungen (Deadlocks):



Beispiel: Unbeschränkt und unendlich:



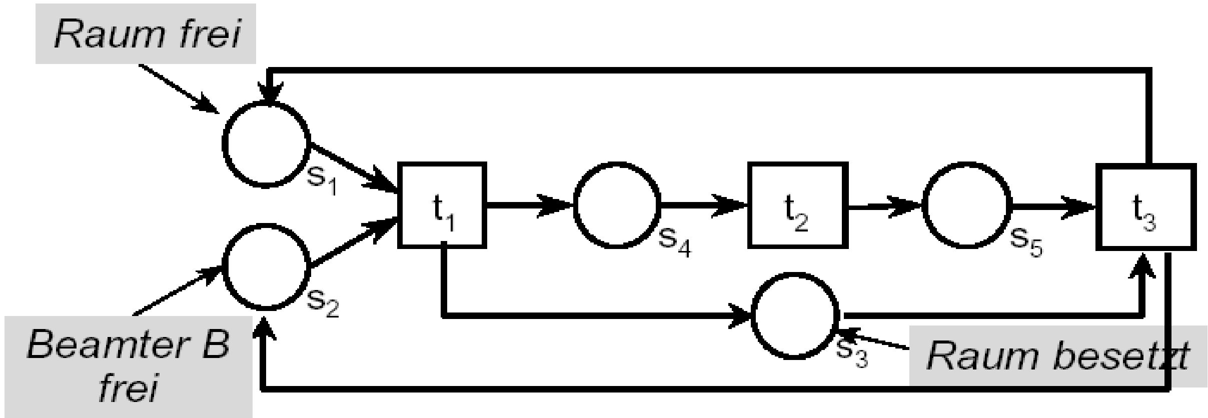
2. Struktur-Elemente und -Eigenschaften

- Eine ***Struktureigenschaft*** eines Petrinetzes hängt nicht von der Markierung des Netzes ab. Es werden Aussagen über das Verhalten aufgrund der Struktur getroffen.
- Stellen und Transitionen (Knoten oder Netzelemente), Kanten wobei **Stellen** passive Komponenten und **Transitionen** aktive Komponenten darstellen.
- Ein Petrinetz beschreibt immer gewisse Struktur-Beziehungen zwischen den Komponenten eines Systems,
  - Vorbereich, Nachbereich,
  - Vorwärtsverzweigung, Rückwärtsverzweigung,
  - Teilnetz, Rand (stellen-, transitionsberandet),
  - Hierarchieerweiterung (in der Entwurfsphase eingesetzt, Darstellung unterschiedlicher Sichtweisen).

HOCHSCHULE MITTWEIDA UNIVERSITY OF APPLIED SCIENCES

Beispiel: Schalterraum einer Behörde

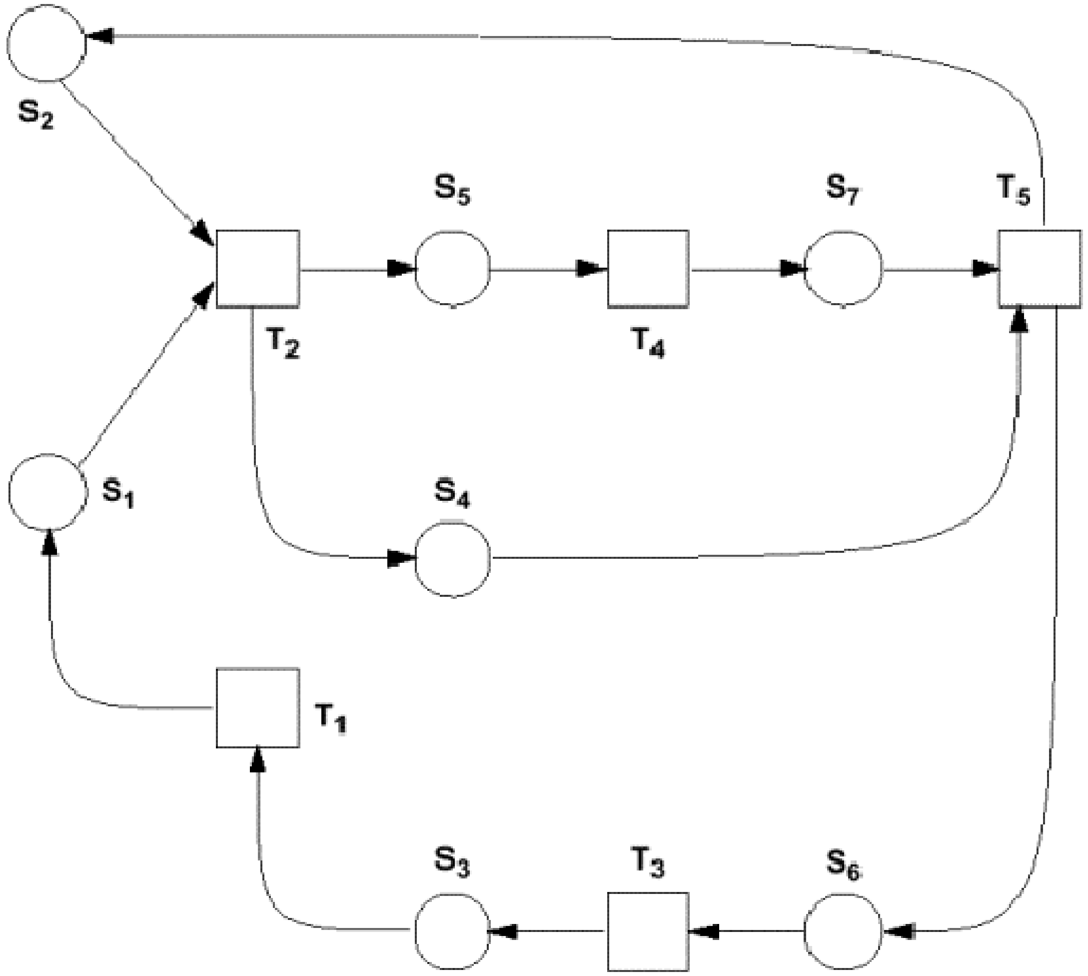
- Ein Schalterraum ist mit dem Beamten B besetzt. Er ist für die Bearbeitung eines Antrages zuständig. Dies erfolgt in Anwesenheit des Antragstellers A.
  Aus Datenschutzgründen darf nur ein Antragsteller im Raum sein.
- mögliche Petrinetz-Modellierung:



- Für eine bestimmte Modellierung kann durch Beschriftung ein Anwendungsbezug hergestellt werden.

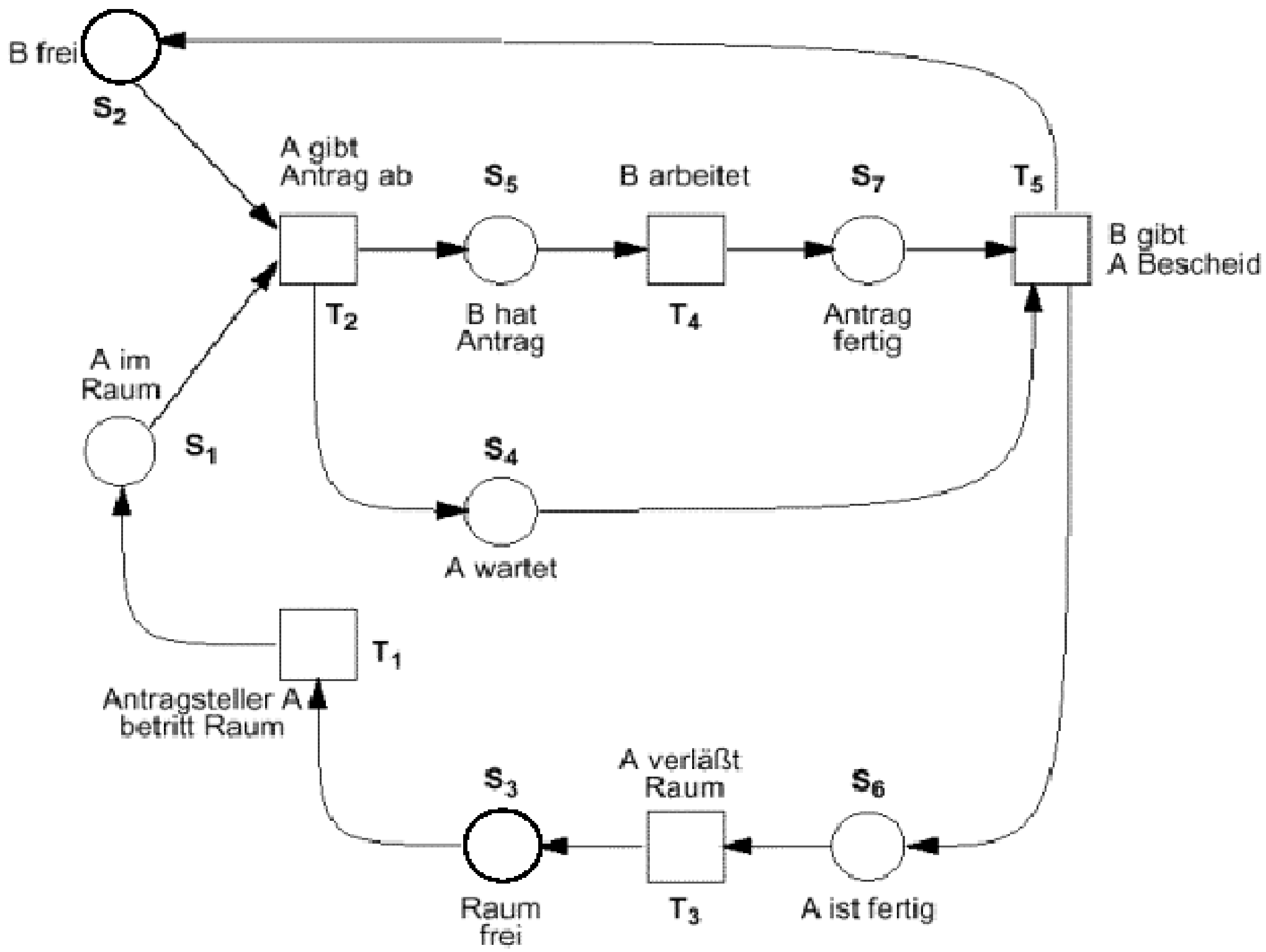
2. Struktur-Elemente und -Eigenschaften

andere mögliche Petrinetz-Modellierung:



2. Struktur-Elemente und -Eigenschaften

Herstellung des Anwendungsbezuges:



2. Struktur-Elemente und -Eigenschaften

Ein Netz $N = (S, T, F)$ ist ***zusammenhängend***,

wenn keine Zerlegung in X_1 und X_2 $(S \cup T = X_1 \cup X_2)$, mit

$X_1, X_2 \neq \emptyset$,

$X_1 \cap X_2 = \emptyset$ mit $F \subseteq (X_1 \times X_1) \cup (X_2 \times X_2)$ existiert.

Es darf also kein Teilnetz existieren, das keine Verbindung zum Rest des Netzes hat.

N heißt ***stark zusammenhängend***,

wenn für je zwei Elemente $x, y \in S \cup T$ mit $x \neq y$

eine Sequenz $(z_1,z_2), (z_2,z_3), \ldots, (z_{n-1},z_n) \in F$ existiert $(n \geq 2)$,

so dass $x = z_1$ und $y = z_n$.

Von jedem Knoten besteht also eine gerichtete Verbindung zu jedem beliebigen Knoten.

2. Struktur-Elemente und -Eigenschaften

HOCHSCHULE MITTWEIDA UNIVERSITY OF APPLIED SCIENCES

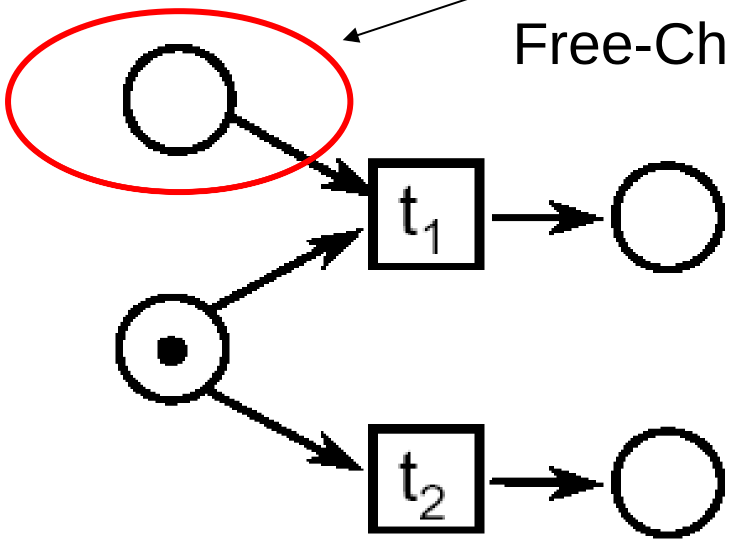
- Ein ***Free-Choice-Netz*** ist ein Netz bei dem
  1. die Transitionen einer vorwärts verzweigten Stelle nicht rückwärts verzweigt sind, bzw.
  2. die Stellen einer rückwärts verzweigten Transition nicht vorwärts verzweigt sind.
- bei einem Konflikt kann also zwischen den Transitionen frei und unabhängig von anderen Stellen ausgewählt werden.
- bei einer Synchronisation werden die Marken in einer Transition wieder zusammengeführt unabhängig von anderen Transitionen.
- Motiv:
  - strukturierte Modellierung,
  - bessere Analysierbarkeit,
  - Modellierung von Geschäftsprozessen (GP) möglich.

2. Struktur-Elemente und -Eigenschaften

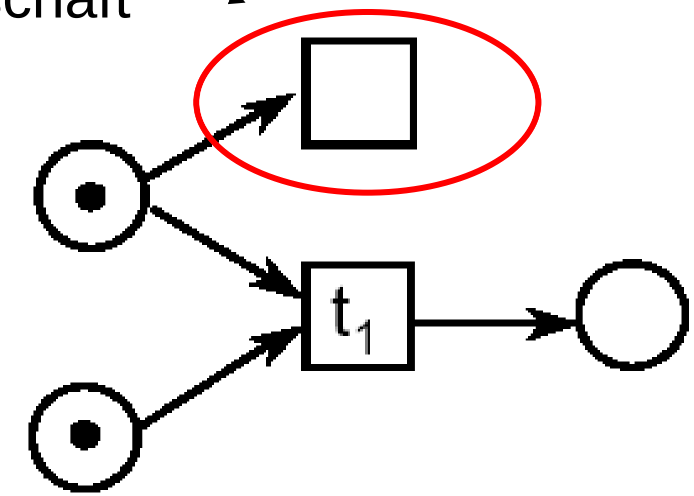
- **Konflikt:**
  - Zwei Transitionen benötigen die gleiche Marke(n)

- **Synchronisation:**
  - Transition kann erst schalten, wann alle Eingangsstellen markiert sind.

Verletzung der
Free-Choice-Eigenschaft



Die Transitionen einer vorwärts verzweigten Stelle dürfen nicht rückwärts verzweigt sein!



Die Stellen einer rückwärts verzweigten Transition dürfen nicht vorwärts verzweigt sein!

2. Struktur-Elemente und -Eigenschaften

- Für ein ***Free-Choice-Netz*** gilt:

  - für Transitionen:

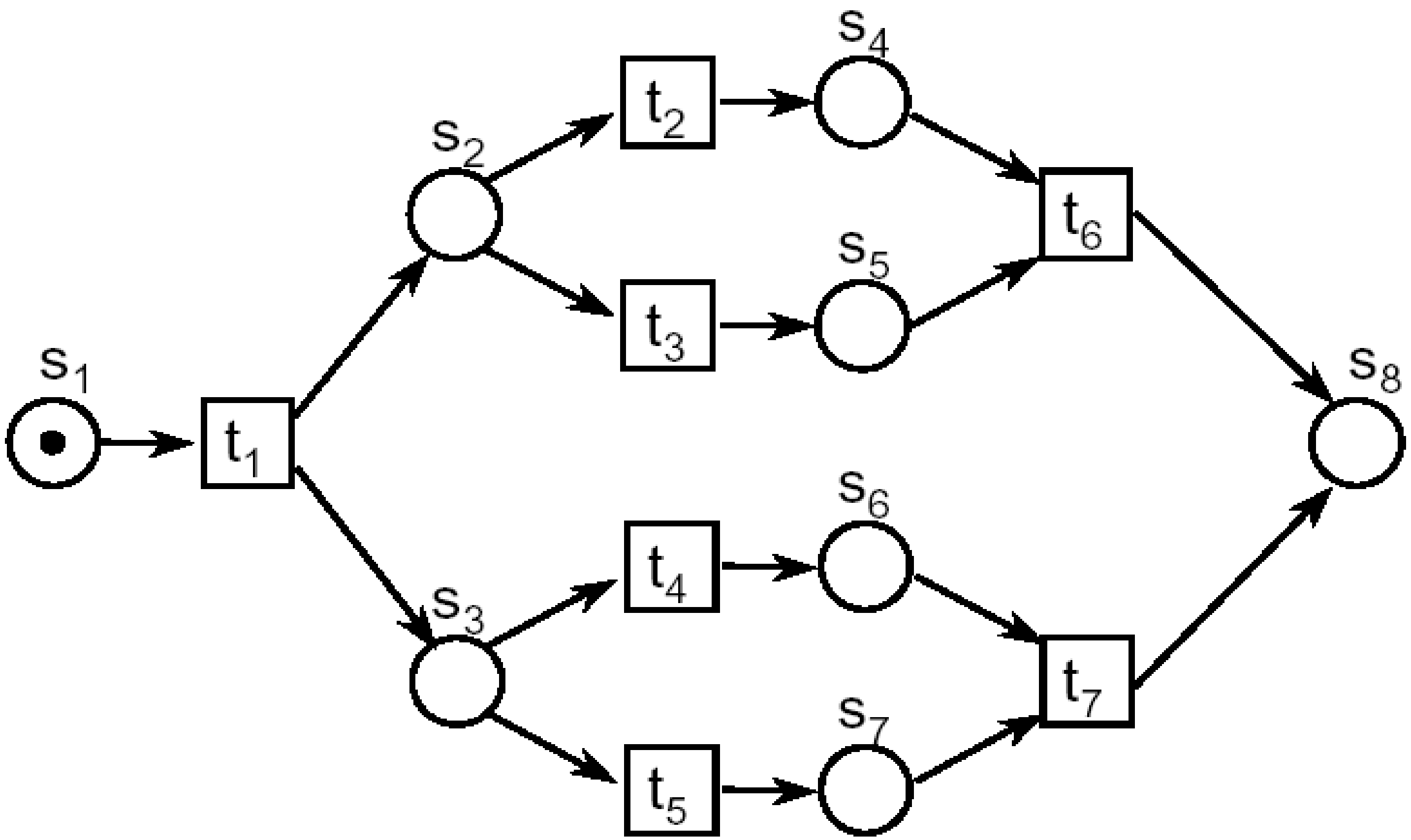
$$\forall t_1, t_2 \in T: \quad \bullet t_1 \cap \bullet t_2 \neq \varnothing \quad \Rightarrow \quad \bullet t_1 = \bullet t_2 = \{ s \}$$

  - für Stellen:

$$\forall s_1, s_2 \in S: \quad s_1\bullet \cap s_2\bullet \neq \varnothing \quad \Rightarrow \quad s_1\bullet = s_2\bullet = \{ t \}$$

HOCHSCHULE MITTWEIDA
UNIVERSITY OF APPLIED SCIENCES

Beispiel: ein Free-Choice-Netz



2. Struktur-Elemente und -Eigenschaften

HOCHSCHULE MITTWEIDA UNIVERSITY OF APPLIED SCIENCES

- Welche Struktureigenschaften hat ein korrektes **Workflow**-Prozess-Modell?
  - definierter Anfang und definiertes Ende („davor‘ und ‚danach‘ ist irrelevant),
  - keine Aufgaben, die nie ausgeführt werden,
  - keine Aufgaben, die nicht zum Ende führen,
- Schlussfolgerung für das Petrinetz:
  - eine Stelle, die die Erzeugung des Falles darstellt (Start),
  - diese Stelle hat keinen Input,
  - eine Stelle, die die Beendigung des Falles darstellt (Ende),
  - diese Stelle hat keinen Output,
  - für jede Transition gilt:
    - es gibt einen Pfad von der Start-Stelle und
    - es gibt einen Pfad zur Ende-Stelle.

→ ein Petri-Netz, das diese Eigenschaften erfüllt, bezeichnen wir als **Workflow-Netz** (WF-Netz).

2. Struktur-Elemente und -Eigenschaften

HOCHSCHULE MITTWEIDA UNIVERSITY OF APPLIED SCIENCES

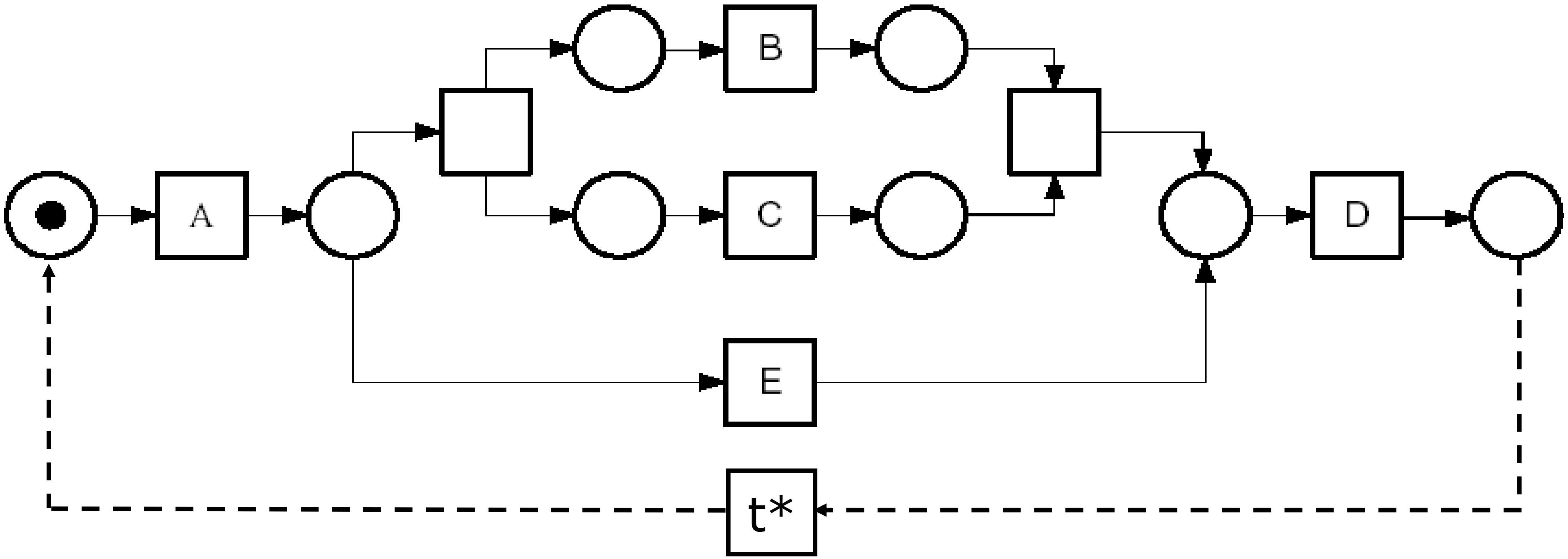
Ein ***Workflow-Netz*** ist ein Petrinetz $N = (S, T, F)$, für welches gilt:

- N hat zwei Stellen i und o mit den folgenden Eigenschaften:
  - Stelle i ist eine Quelle, d.h. $\bullet i = \emptyset$,
  - Stelle o ist eine Senke, d.h. $o \bullet = \emptyset$.
- Wenn zu N eine Transition t* hinzugefügt wird, die o und i verbindet mit $\bullet t^* = \{o\}$, $t^*\bullet = \{ i \}$, so ist das resultierende Netz stark zusammenhängend.
- Folgerung: i und o sind eindeutig.
- Ein WF-Netz ist das Petrinetz-Modell eines Workflow-Prozesses.

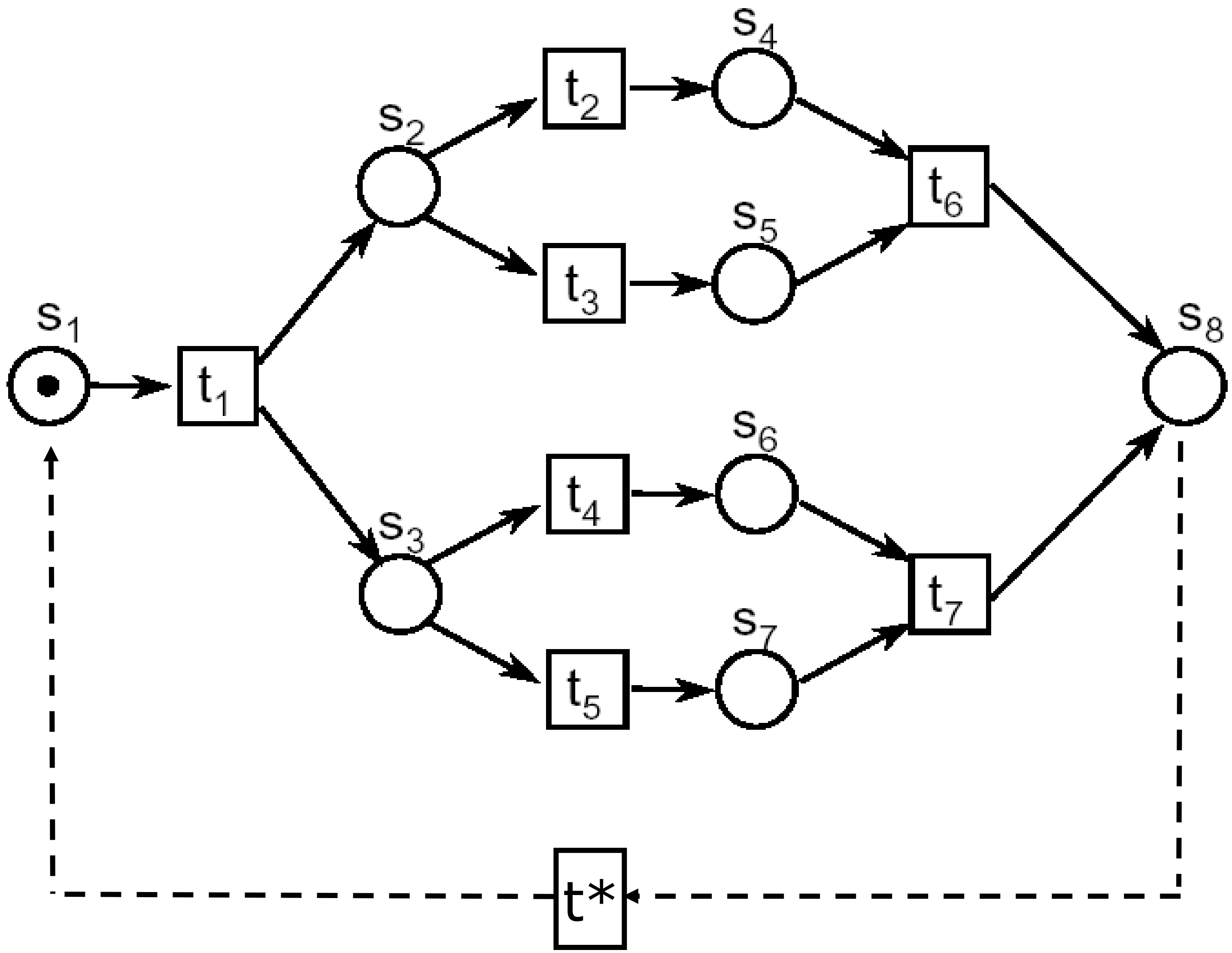
Bemerkung: Die Stellen i und o sind eindeutig und Teil des WF-Netzes, während Transition t* nicht zum WF-Netz gehört, sondern nur zur Prüfung dient, ob das Netz stark zusammenhängend ist.

2. Struktur-Elemente und -Eigenschaften

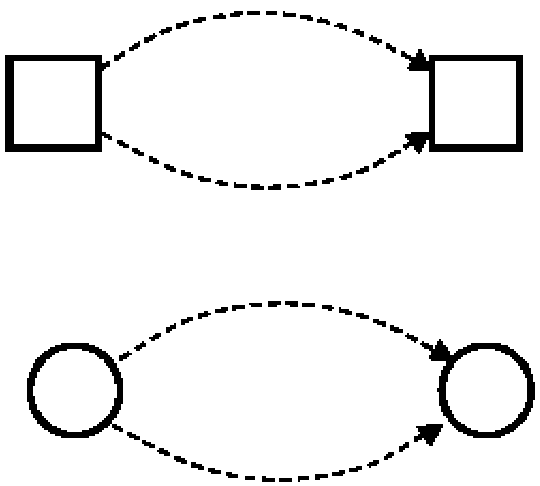
Beispiel: Workflow-Netz (nicht stark zusammenhängend, erst durch Einfügung einer Transition t* (als Prüfung) wird es stark zusammenhängend)



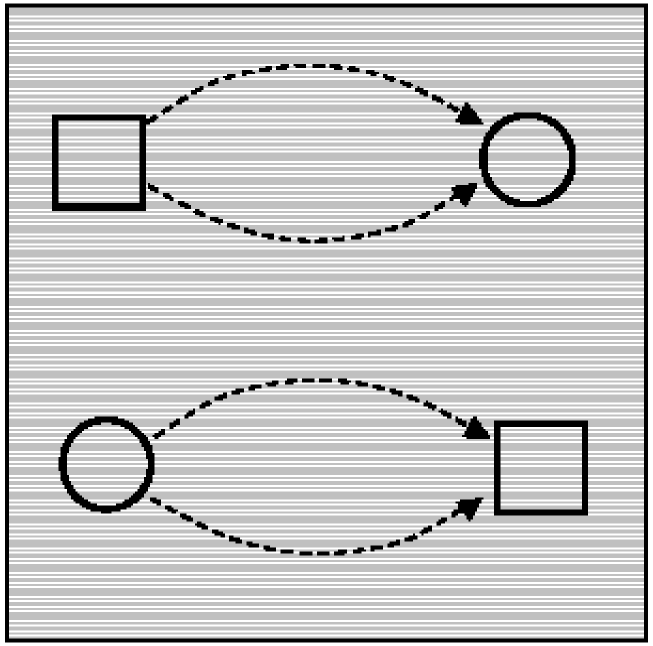
Frage: ist dieses Free-Choice-Netz auch ein WF-Netz?



Routing-Kombinationen:



„gute“ Routingkonstrukte



„schlechte“ Routingkonstrukte

AND-splits mit AND-joins und OR-splits mit OR-joins vereinigen

HOCHSCHULE MITTWEIDA UNIVERSITY OF APPLIED SCIENCES

Well-handled, well-structured

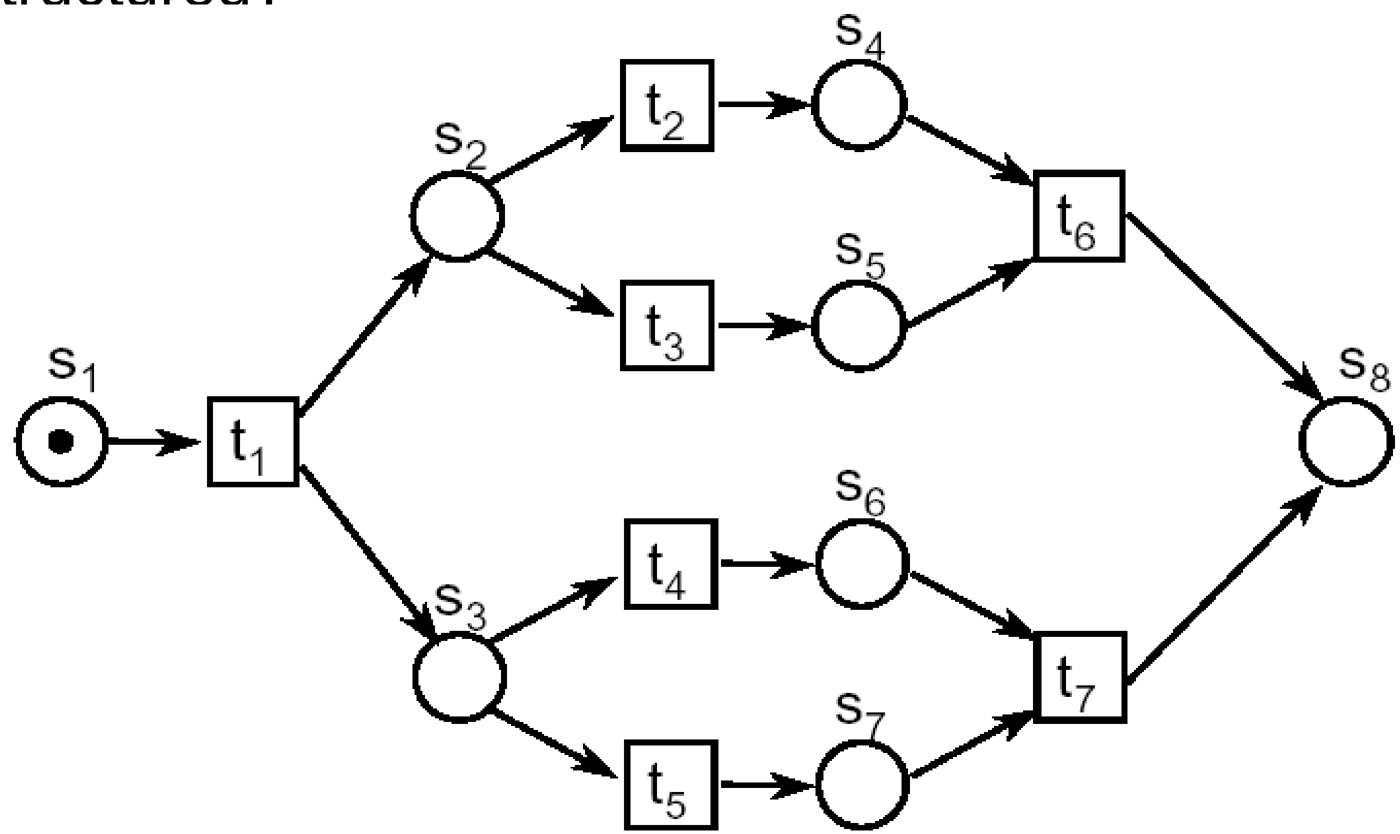
Ein WF-Netz N = (S, T, F) ist genau dann ***well-handled***,

wenn für je zwei Knoten x, y
(x, y: $x \in T$ und $y \in S$ oder $x \in S$ und $y \in T$)

keine zwei Pfade von x nach y existieren, die
- aus mehr als 2 Elementen bestehen UND
- nur x, y gemeinsam haben.

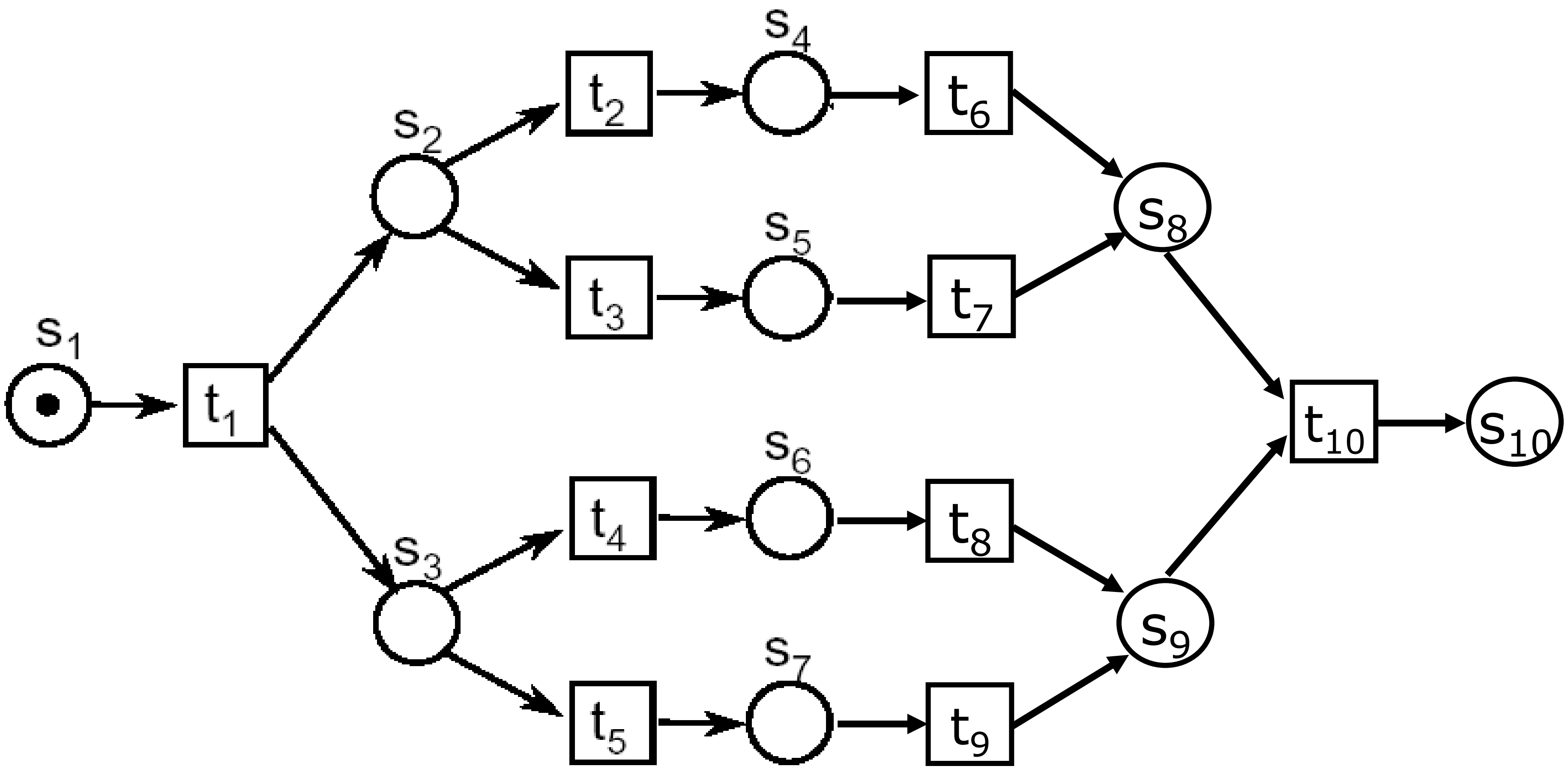
Ein WF-Netz ist ***well-structured,***
wenn das um t^* (mit $\bullet t^* = \{o\}$, $t^*\bullet = \{ i \}$)
erweiterte Netz well-handled ist.

Frage: ist dieses Free-Choice-Netz auch well-handled, vielleicht sogar well-structured?



→ Nein, wegen (t_1,s_8), (s_2,t_6) und (s_3,t_7)!

Beispiel: für ein well-handled und well-structured Netz

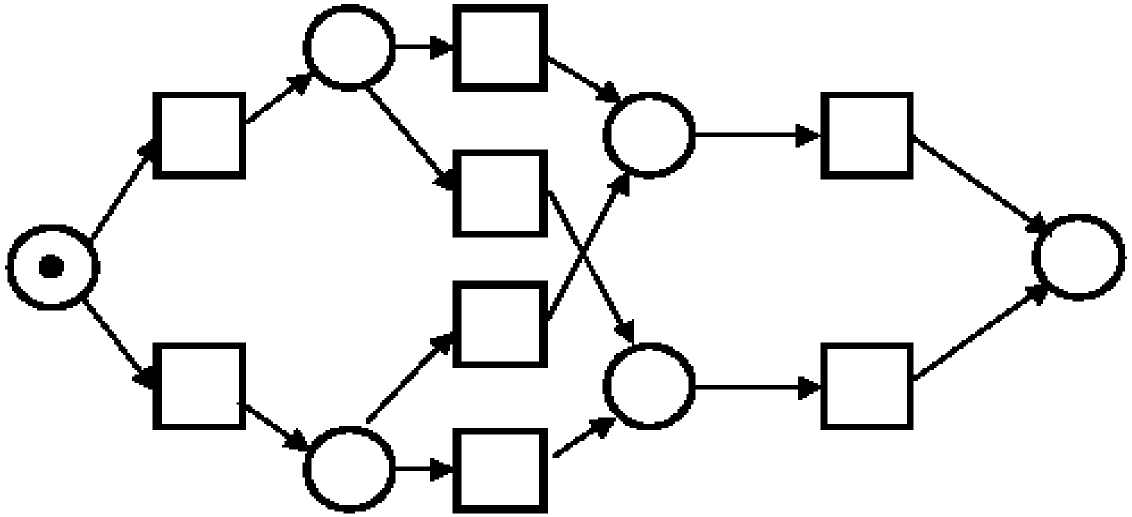


2. Struktur-Elemente und -Eigenschaften

Ein WF-Netz N = (S, T, F) ist genau dann ein ***Zustandsautomat***, wenn gilt:

$$\forall\, t \in T: \qquad |\bullet t| = |t \bullet| = 1,$$

d.h. jede Transition in $t \in T$ hat genau eine Eingangs- und genau eine Ausgangskante.



Motiv: - immer nur genau eine Marke im System, Bearbeitung eines Falles kann also nur sequenziell ablaufen

Bemerkung: Es werden weder Marken erzeugt, noch verschwinden welche! Modellierung paralleler Abläufe nicht möglich!

2. Struktur-Elemente und -Eigenschaften

Ein Teilnetz $N' = (S', T', F')$ eines WF-Netzes $N = (S, T, F)$ mit
$S' \subseteq S$, $T' \subseteq T$, $F' \subseteq F$,
ist eine ***S-Komponente*** von N genau dann, wenn:

- N' ist stark zusammenhängend,
- N' ist Zustandsautomat,
- $\forall s \in S': \quad \bullet s \cup s \bullet \subseteq T'$.

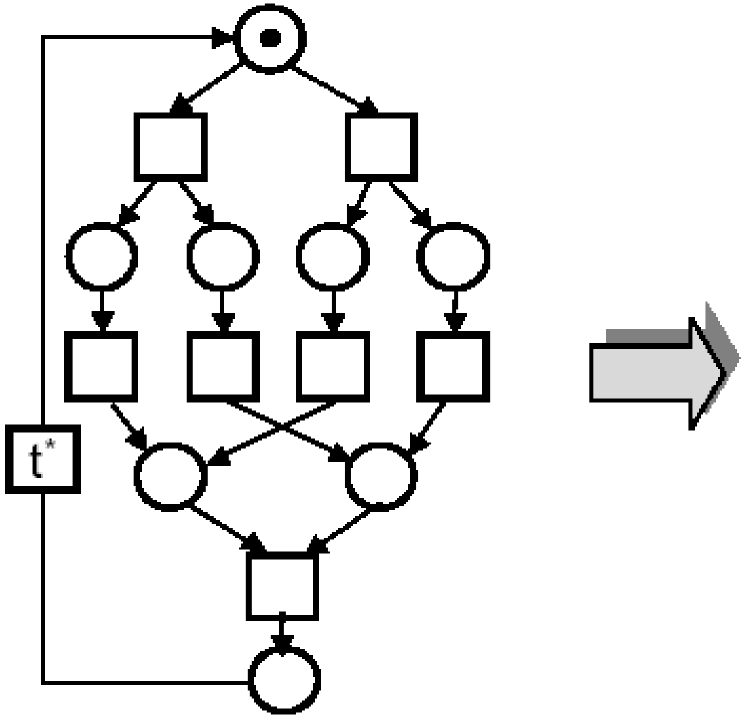
Ein WF-Netz $N = (S, T, F)$ ist ***s-coverable*** genau dann, wenn
$\forall s \in S \quad \exists$ S-Komponente $N' = (S', T', F')$ von N: $s \in S'$.

Für jede Stelle $q \in S'$ und $t \in T$:
$(q, t) \in F \Rightarrow (q, t) \in F'$ und $(t, q) \in F \Rightarrow (t, q) \in F'$.

Besteht ein WF-Netz aus S-Komponenten,
so wird es *s-coverable* genannt.

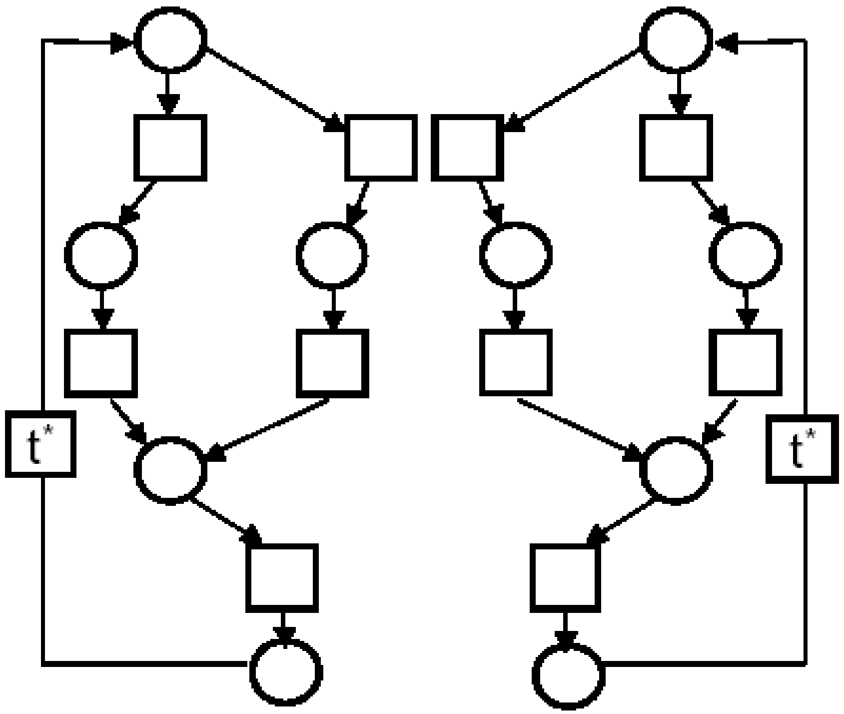
Bemerkung: Für ein s-coverable Petrinetz-Modell existiert eine Dekomposition in S-Komponenten, die Zustandsautomaten sind!

Beispiel: S-Komponenten / S-coverable



lebendiges, beschränktes
free-choice Netz

S-Komponenten:



→ S-coverable

Zusammenfassung:

- free-choice, well-structured, s-coverable sind sehr nützliche Struktureigenschaften zur Prozessanalyse,
- s-coverable sollte jede Workflow-Definition sein;
  diese Eigenschaft ist die Generalisierung der free-choice und well-structured-Eigenschaft,
- ein WF-Netz sollte mindestens entweder free-choice oder well-structured sein, um es effizient analysieren zu können (sonst oft Fehlerquelle).

HOCHSCHULE MITTWEIDA
UNIVERSITY OF APPLIED SCIENCES

Ziel: Allgemeine Aussagen über das Schaltverhalten des Netzes
→ Basis für formale Analyse dynamischer Systeme im Vorfeld der Implementierung

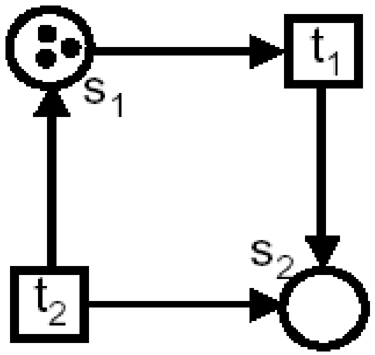
Wiederholung:

- Marken, Markierungen (Darstellung von Abläufen),
- Anfangsmarkierung, Folgemarkierung,
- Erreichbarkeit (erreichbare Markierung, Schaltfolge),
- Konflikt, Synchronisation,
- Trigger,
- unterscheidbare Marken,
- Zeiterweiterung.

3. Dynamische Elemente und Eigenschaften

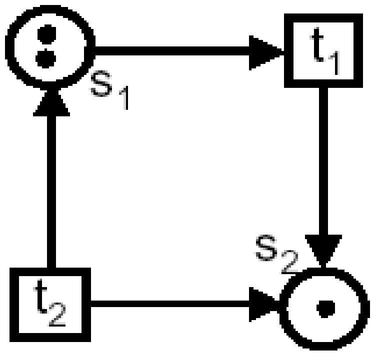
Sei N = (S, T, F) ein Petrinetz und m_0 eine Markierung von N, m_0 wird ***Anfangsmarkierung von N*** genannt.

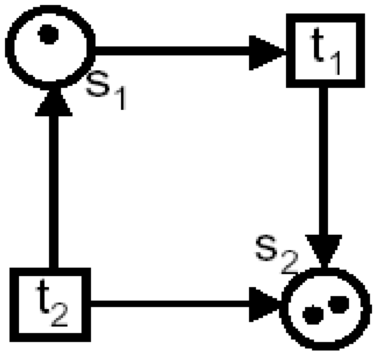
$m_0 = (3, 0)$

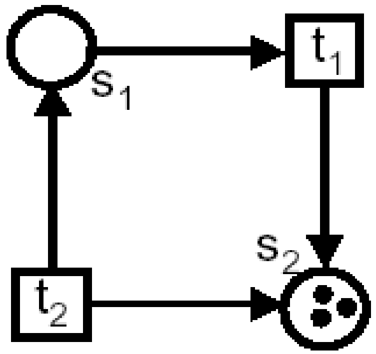


Ablaufzeitpunkte:

τ_1: $m_1 = (2, 1)$ τ_2 : $m_2 = (1, 2)$ τ_3 : $m_3 = (0, 3)$

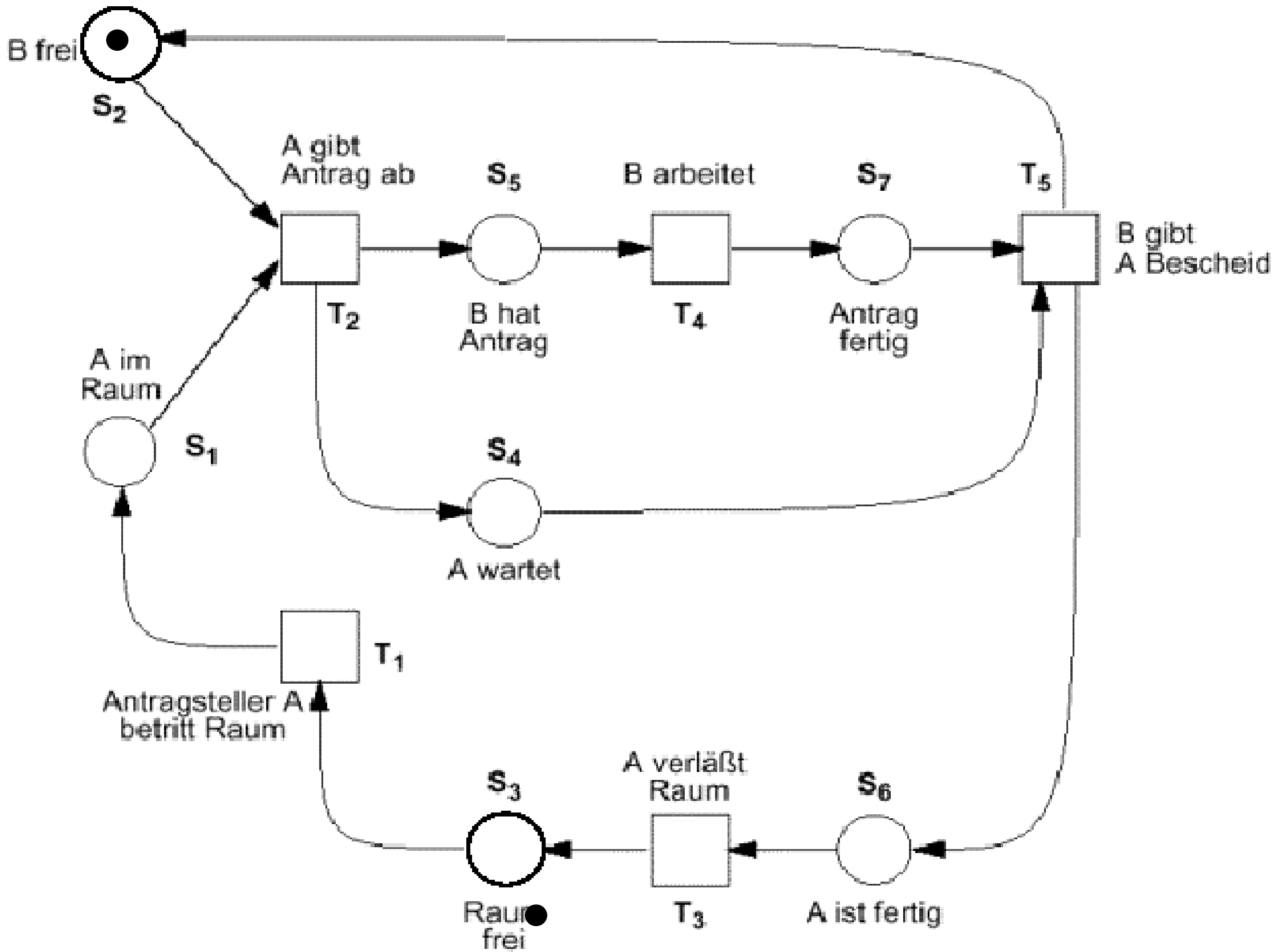






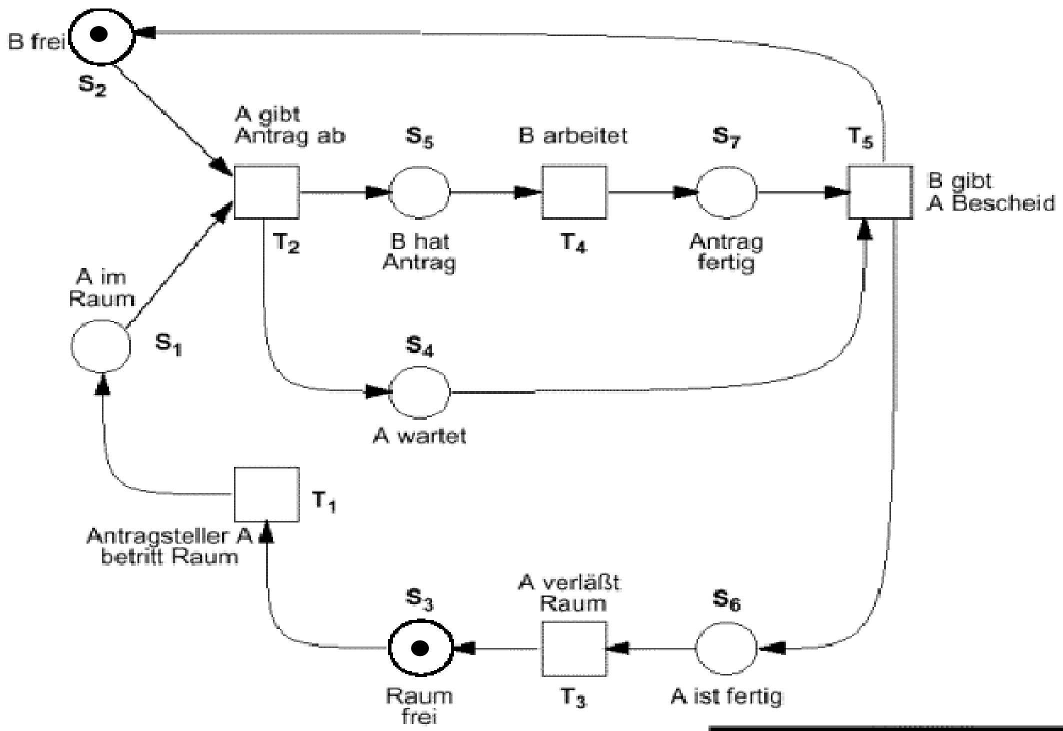
HOCHSCHULE MITTWEIDA UNIVERSITY OF APPLIED SCIENCES

Beispiel: Schalterraum einer Behörde



3. Dynamische Elemente und Eigenschaften

HOCHSCHULE MITTWEIDA
UNIVERSITY OF APPLIED SCIENCES



Zeit-punkt	Markierung der Stellen							Schaltende Transition				
	S_1	S_2	S_3	S_4	S_5	S_6	S_7	T_1	T_2	T_3	T_4	T_5
0		x	x					x				
1	x	x							x			
2				x	x						x	
3				x			x					x
4		x				x				x		
5		x	x									

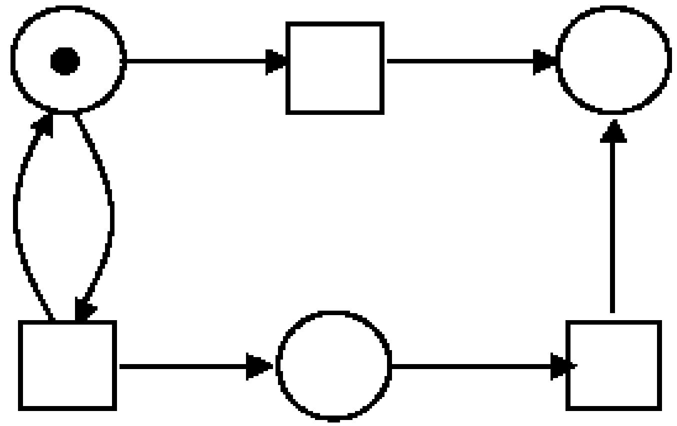
HOCHSCHULE MITTWEIDA
UNIVERSITY OF APPLIED SCIENCES

Beschränktheit

(N,m_0) heißt ***beschränkt***, wenn eine Schranke b existiert,
so dass $m(s) \leq b$ für alle $s \in S$; für alle $m \in [m_0>$.

(N,m_0) heißt ***1-beschränkt*** oder ***sicher***, wenn $b = 1$.

Frage: beschränkt oder unbeschränkt?



→ **unbeschränkt!**

HOCHSCHULE MITTWEIDA
UNIVERSITY OF APPLIED SCIENCES

Komplementbildung

(zur Erzwingung einer maximalen Anzahl von Marken in s)

Sei s eine Stelle;
eine neue Stelle s* mit

$$s^{*}\bullet \quad = \quad \bullet s \setminus s\bullet$$
$$\bullet s^{*} \quad = \quad s\bullet \setminus \bullet s$$
$$m_0(s^{*}) \quad = \quad k(s) - m_0(s),$$

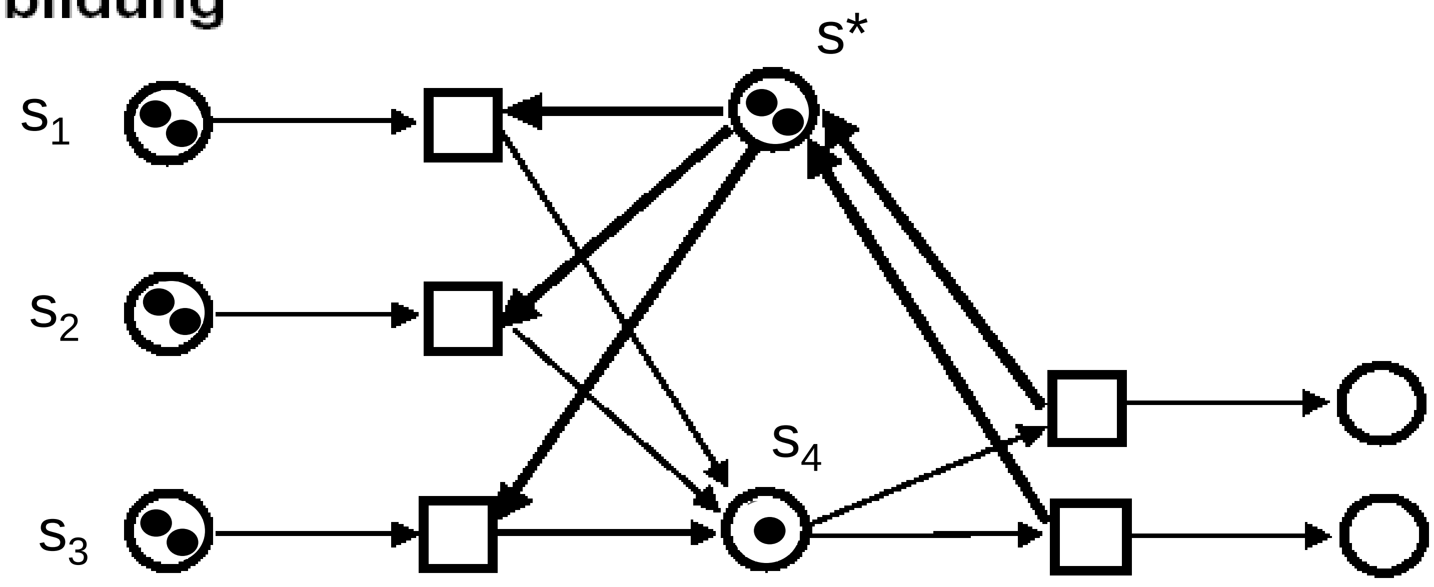
wird als das ***Komplement*** (die Komplementstelle) von s bezeichnet.

k(s) ist die gewünschte Schranke für s

Es gilt: $\forall\ m \in [m_0 > \ :\ m(s) = K(s) - m(s^{*})$

d.h. $m(s) \leq K(s) \quad \forall m \in [m_0 >$

Komplementbildung



Ohne s* können alle in den Stellen s_1, s_2 und s_3 befindlichen Marken in s_4 erscheinen. Um dies zu umgehen, wird Komplement-stelle s* eingeführt, derart dass alle Input-Transitionen von s_4 Output-Transitionen von s* werden und alle Output-Transitionen von s_4 Input-Transitionen von s*.
Um eine Beschränkung mit der Schranke b=3 zu erreichen, muss $m_0(s^*)=b(s_4)-m_0(s_4)$ gelten. Somit muss die Stelle s* Anfangs 2 Marken enthalten.

Tot, Verklemmungsfreiheit:

- Eine Transition t heißt **tot** unter einer Markierung m, wenn kein m´ $\in$ [m> die Transition t aktiviert.
- Eine Markierung heißt **tot**, wenn alle Transitionen tot sind.
- Ein Petrinetz heißt **tot**, wenn seine Anfangsmarkierung m_0 tot ist.
- Eine Markierung heißt **verklemmungsfrei**, wenn keine tote Markierung erreichbar ist.

→ In einem System ist es sehr wichtig tote Markierungen zu erkennen und diese zu eliminieren, um einen einwandfreien Ablauf zu gewährleisten.

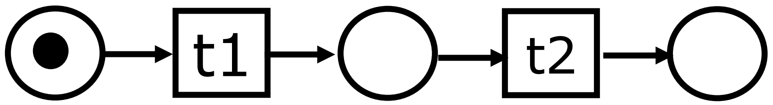
HOCHSCHULE MITTWEIDA UNIVERSITY OF APPLIED SCIENCES

Lebendigkeit:

- Gibt es von jeder erreichbaren Markierung aus einen Weg zu jeder Transition?

- Eine Transition heißt **lebendig** unter einer Markierung m, wenn sie unter keiner Folgemarkierung $m' \in [m>$ tot ist.
- Eine Markierung m heißt **lebendig**, wenn alle Transitionen unter m lebendig sind.
- Ein Petrinetz heißt **lebendig**, wenn seine Anfangsmarkierung m_0 **lebendig** ist.

Tot ≠ nicht lebendig



Die Transitionen t_1 und t_2 sind beide nicht tot unter der Anfangsmarkierung m_0=(1 0 0). Also ist auch m_0 nicht tot.

Allerdings sind beide Transitionen auch nicht lebendig, da sie unter der Folgemarkierung m´=(0 0 1) tot sind. Somit ist auch die Anfangsmarkierung m_0 und damit auch das Petrinetz unter m_0 nicht lebendig!

HOCHSCHULE MITTWEIDA UNIVERSITY OF APPLIED SCIENCES

Deadlock:

- Eine Markierung, nach deren Erreichen es keine Transition gibt, die noch schalten kann, heißt ***Deadlock***.

Livelock:

- Eine Markierung, nach deren Erreichen keine Beendigung und auch kein Deadlock mehr möglich ist, heißt ***Livelock***.

Reversibel:

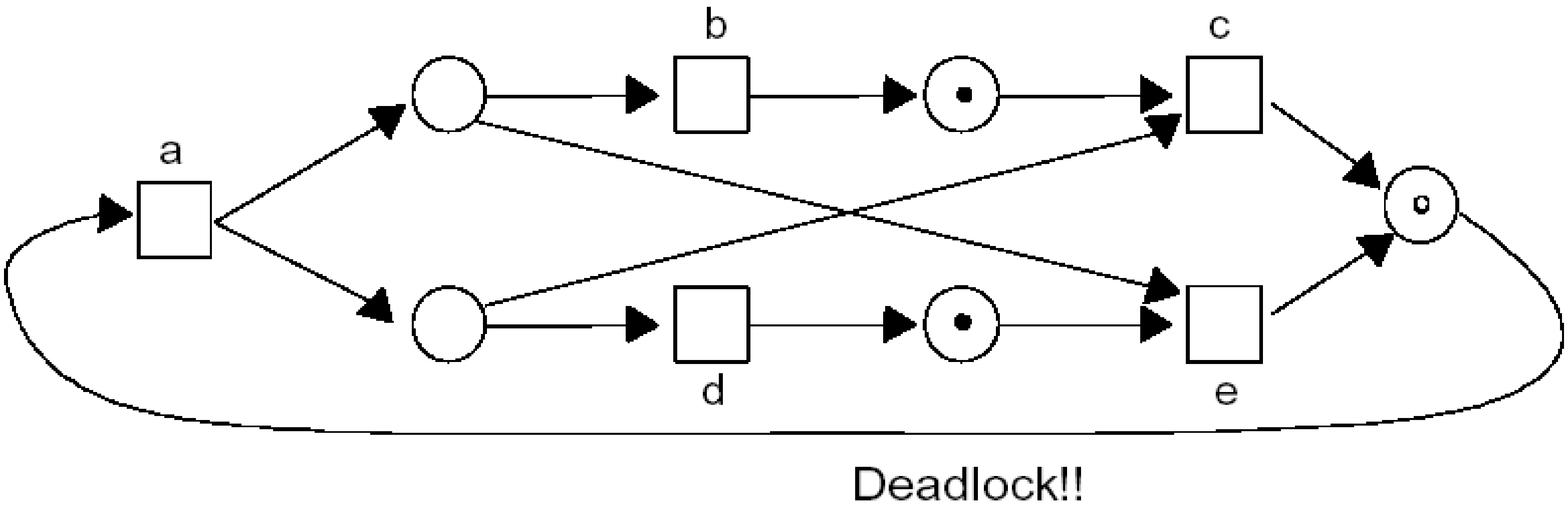
- Ein markiertes Petrinetz mit Anfangsmarkierung m_0 heißt ***reversibel***, wenn m_0 von jeder erreichbaren Markierung aus erreichbar ist.

Terminierung:

- Ein markiertes Petrinetz ***terminiert***, wenn die Menge der Schaltfolgen endlich ist.

HOCHSCHULE MITTWEIDA UNIVERSITY OF APPLIED SCIENCES

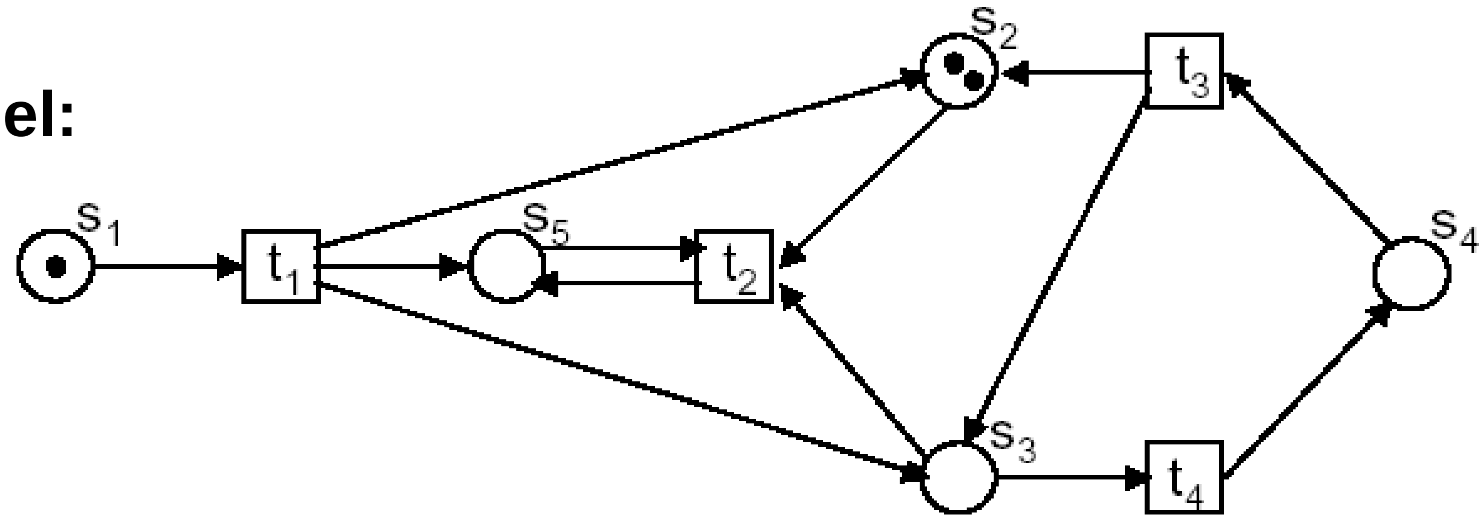
Beispiel:



beliebig lange Schaltfolgen (a b c) möglich, aber nicht verklemmungsfrei (a b d)

HOCHSCHULE MITTWEIDA UNIVERSITY OF APPLIED SCIENCES

Beispiel:



1. Tote Transitionen unter m_0: keine,
2. Deadlock, durch Schalten von t_1 und t_2,
3. Lebendigkeit, nicht gegeben durch die Anwesenheit eines Deadlocks,

Eigenschaften in Sätzen:

- Jedes lebendige markierte Petrinetz mit mindestens einer Transition ist verklemmungsfrei.
- Ein markiertes Petrinetz ist genau dann lebendig, wenn unter keiner erreichbaren Markierung eine tote Transition existiert.
- Ein markiertes Petrinetz ist genau dann beschränkt, wenn die Menge der erreichbaren Markierungen endlich ist.

Entscheidend für einen Workflow-Prozess ist die Eigenschaft **Soundness** (minimale Anforderungen an einen Workflow)

- Beendigungsmöglichkeit:
  Es sollte immer möglich sein einen Fall zu beenden. Dies garantiert, dass keine Deadlocks vorhanden sind.
- Richtigkeit der Beendigung:
  Das Prozessende sollte eindeutig sein, d.h. nach Beendigung eines Falles sollten keine Aufgaben für diesen Fall mehr auszuführen sein.
- Aufgabenerfordernis:
  Jede Aufgabe sollte die Möglichkeit haben ausgeführt zu werden, sollte also für den Prozess erforderlich sein.

Schlussfolgerung für das Petrinetz:

- Für jeden Fall (Case) terminiert das entsprechende WF-Netz irgendwann.
- In diesem Zustand befindet sich genau eine Marke in der Stelle o, alle anderen Stellen sind leer.
  (Bezeichnung: Markierung o=(0 0 … 1); analog: Markierung i für den Anfangszustand)
- Es sollten keine toten Transitionen existieren, d.h. jede Aufgabe (Task) sollte beim Durchlaufen eines geeigneten Pfades im WF-Netz ausführbar sein.

HOCHSCHULE MITTWEIDA
UNIVERSITY OF APPLIED SCIENCES

Soundness (Stabilität, Zuverlässigkeit)
Eine durch ein WF-Netz $N = (P, T, F)$ modellierte Prozessdefinition wird als ***sound*** bezeichnet genau dann, wenn gilt:

- für jede von i aus erreichbare Markierung M existiert eine Schaltfolge von M nach o, d.h.
  $\forall\, M: \quad i \xrightarrow{*} M \quad \Rightarrow \quad M \xrightarrow{*} o$

- die Markierung o ist die einzige von i erreichbare Markierung mit mindestens einer Marke in o, d.h.
  $\forall\, M: \quad (i \xrightarrow{*} M \wedge M \geq o) \Rightarrow \quad M = o$

- es existieren keine toten Transitionen, d.h.
  $\forall\, t \in T \quad \exists\, M, M': \quad i \xrightarrow{*} M \xrightarrow{t} M'$

Soundness:

- Ein WF-Netz ist sound genau dann, wenn es lebendig und beschränkt ist.
- Soundness lässt sich mit Standard-Methoden für Petrinetze nachweisen.
- Für Free-Choice-Netze kann Soundness in polynomieller Zeit gezeigt werden.

Soundness ist eine Minimalanforderung.

Für komplexe WF-Netze ist die Entscheidbarkeit von Soundness ein exponentiell hartes Problem.

Die Definition von Soundness liefert keine Anhaltspunkte für Verbesserungen der Prozessdefinition.

Invarianten:

- Als Invariante eines Systems bezeichnet man eine solche Eigenschaft, die bei der Arbeit des Systems unabhängig vom konkreten Ablauf erhalten bleibt.

Stellen-Invariante (kurz S-Invariante):

- formuliert Bedingung über eine gewisse Konstantheit der Anzahl der Token eines Netzes N.

Transitions-Invariante (kurz T-Invariante):

- ist Folge von Transitionen die von einer Markierung m aus geschaltet werden kann, um wieder die gleiche Markierung m zu erreichen.

- Eine S-Invariante Y weist jeder Stelle eine ***Gewichtung*** zu, derart, dass die ***gewichtete Summe*** der Marken für alle Markierungsübergänge ***konstant*** bleibt.

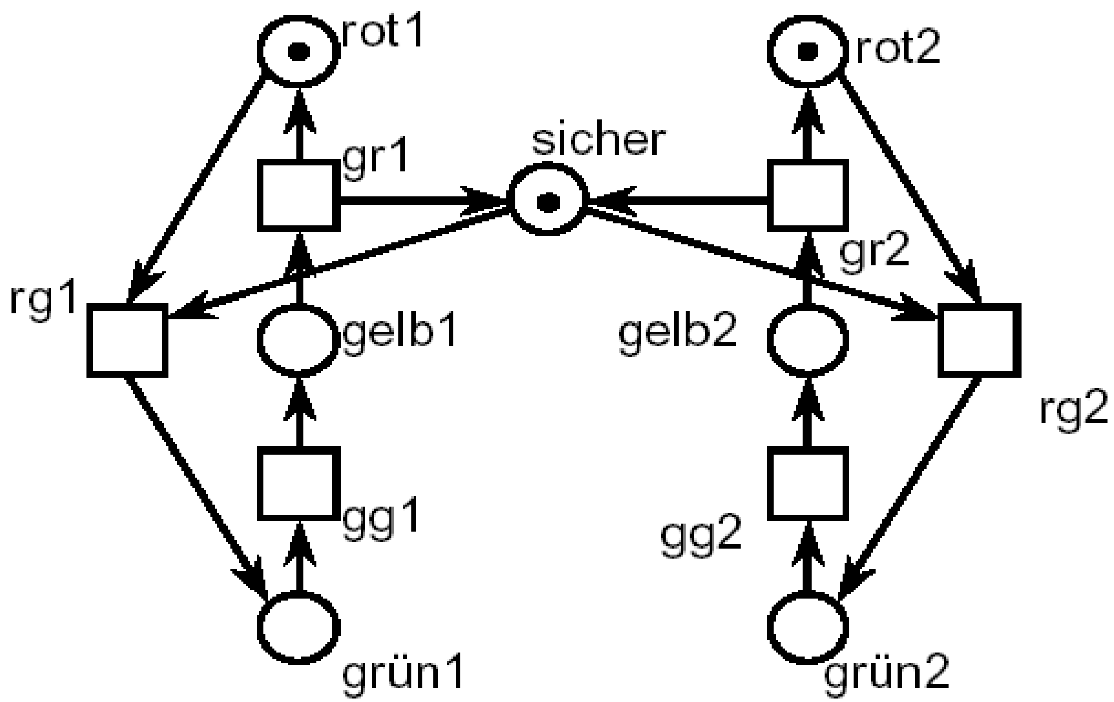
Sei $N=(S,T,F,m_0)$ ein Petrinetz mit der Anfangsmarkierung m_0 und $|s|=n$ ($s \in S$). Dann heißt eine Gewichtung $Y=(y_1(s_1), y_2(s_2), \ldots, y_n(s_n))$ mit den Gewichten y_i der Stellen s_i S-Invariante, wenn:

$$\sum_{i=1}^{n} y_i(m(s_i)) = const., \forall m \in [m_0 >$$

- D.h. bei einer S-Invarianten Y ist die mit Y gewichtete Tokenzahl invariant gegenüber dem Schalten.
- Für jede erreichbare Markierung entspricht die gewichtete Summe der Marken der gewichteten Summe der Marken der Anfangsmarkierung (eine Konstante).

HOCHSCHULE MITTWEIDA UNIVERSITY OF APPLIED SCIENCES

Stellen-Invarianten: Beispiel



$$\begin{aligned} rot1 + gelb1 + grün1 &= 1 \quad (1, 1, 1, 0, 0, 0, 0) \\ rot2 + gelb2 + grün2 &= 1 \quad (0, 0, 0, 1, 1, 1, 0) \\ sicher + grün1 + grün2 + gelb1 + gelb2 &= 1 \quad (0, 1, 1, 0, 1, 1, 1) \\ rot1 + rot2 - sicher &= 1 \quad (1, 0, 0, 1, 0, 0, -1) \end{aligned}$$

HOCHSCHULE MITTWEIDA UNIVERSITY OF APPLIED SCIENCES

Transitions-Invarianten:

- Eine T-Invariante X = (x (t_1),x (t_2),x (t_3), ... ,x (t_n)) gibt an, wie oft welche Transitionen von einer Markierung m aus geschaltet werden müssen, um wieder die gleiche Markierung m zu erreichen.

  d.h.

- Eine T-Invariante ist eine Schaltfolge $\tau = t_1\ t_2 \ldots t_n$ von m nach m´, so dass gilt : m (s_i) = m´(s_i), für alle i.