Systemmodellierung
Teil 1: Ereignisdiskrete Systeme

Kapitel 1:
Einführung

UNIVERSITÄT
DES
SAARLANDES

AES
Automatisierungs- und Energiesysteme

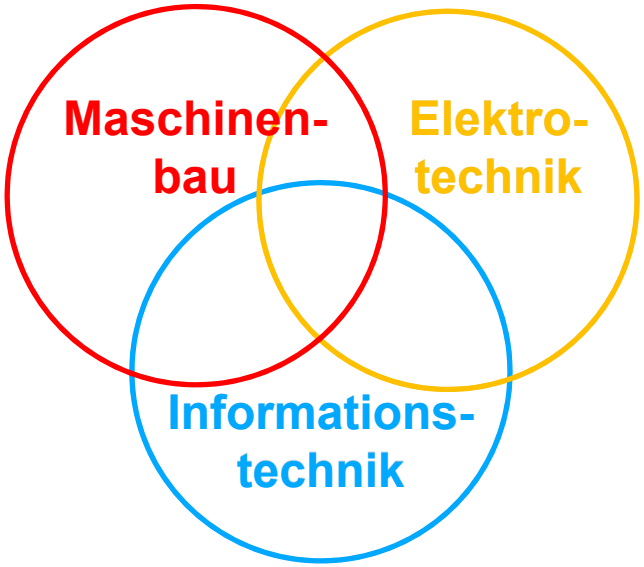
Was ist das wesentliche Neue in SM

- **Bisher: Statik, Dynamik, …**

  Technische Grundlagen wichtiger mechatronischer **Elemente**

- **SM:**

  Übergang von den Einzelelementen zu **Systemen**
  (z.B. Bremssystem, Industrieroboter, Kompressorstation, …)



Warum *System*betrachtung?

Einordnung von SM in die mechatronischen Grunddisziplinen?

Rolle der **Informationstechnik** in der Systemtechnik?

Kapitel 1: Einführung

- Begriffe *System*, *Prozess*, *Modell*
- Allgemeine Prinzipien und Methoden der Modellierung

Kapitel 2: Ereignisdiskrete Systemmodelle

- Ereignisbegriff
- Formale Darstellungen
  - Boole‘sche Algebra
  - Automaten
  - Petrinetze

Kapitel 3: Einführung in hybride Modelle

- Definition und Beispiele
- Darstellung als hybrider Automat

Kapitel 4: Realisierung mit Statecharts

Inhalt von Kapitel 1: Einführung

1. Was ist ein System?
   - Verschiedene technische Systembegriffe
   - Grundkomponenten
   - Schnittstellen

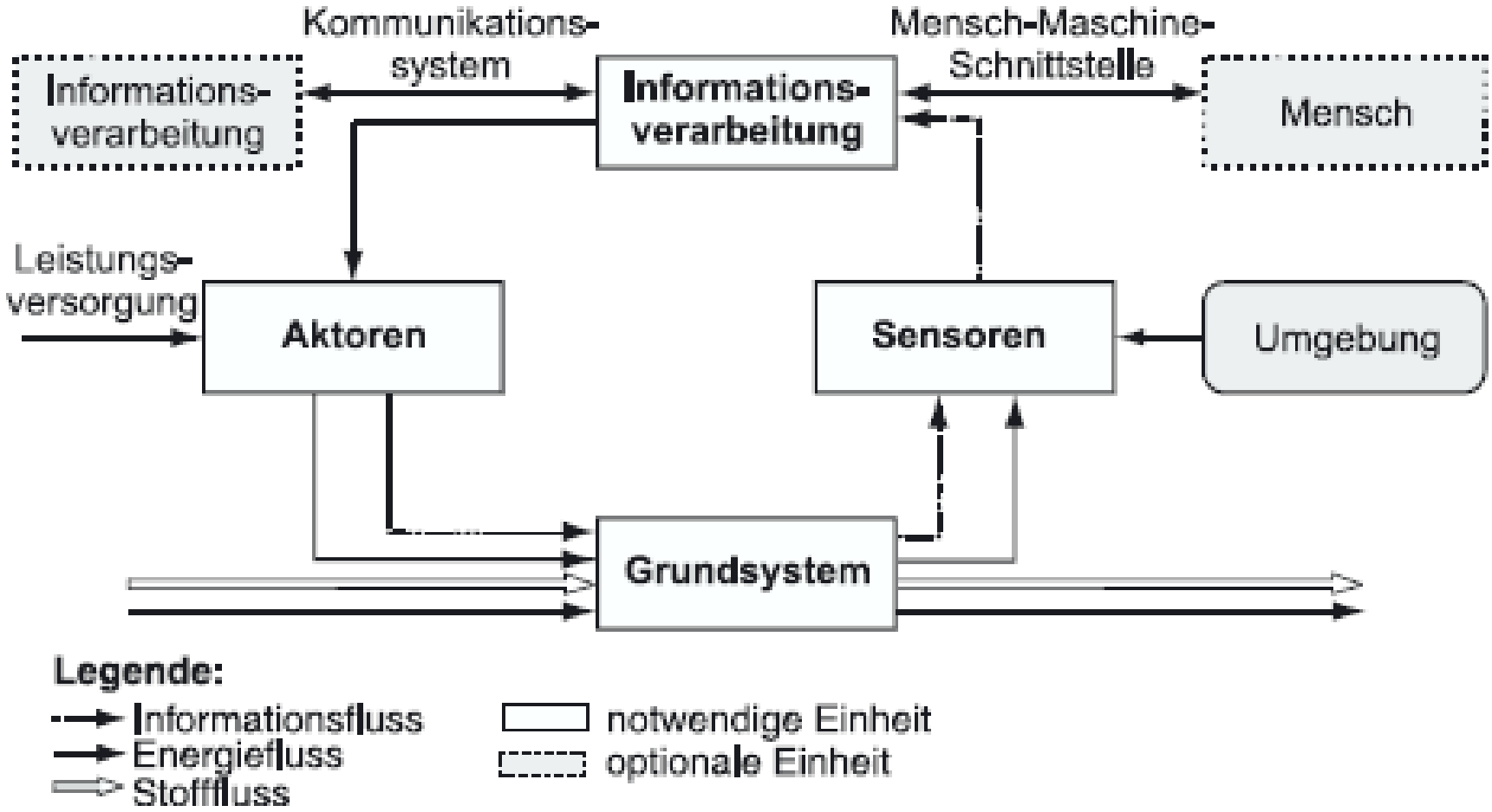
2. Was versteht man unter einem Prozess?
   - Flussgrößen und Elementarfunktionen von Prozessen
   - Beispiele

3. Modellierung von Systemen
   - Systemgrenzen; offene und geschlossen Systeme
   - Prinzipien der Systemmodellierung
   - Methoden der Systemmodellierung
   - Strukturen: aufbauorientierte und verhaltensorientierte Modelle

4. Zusammenfassung

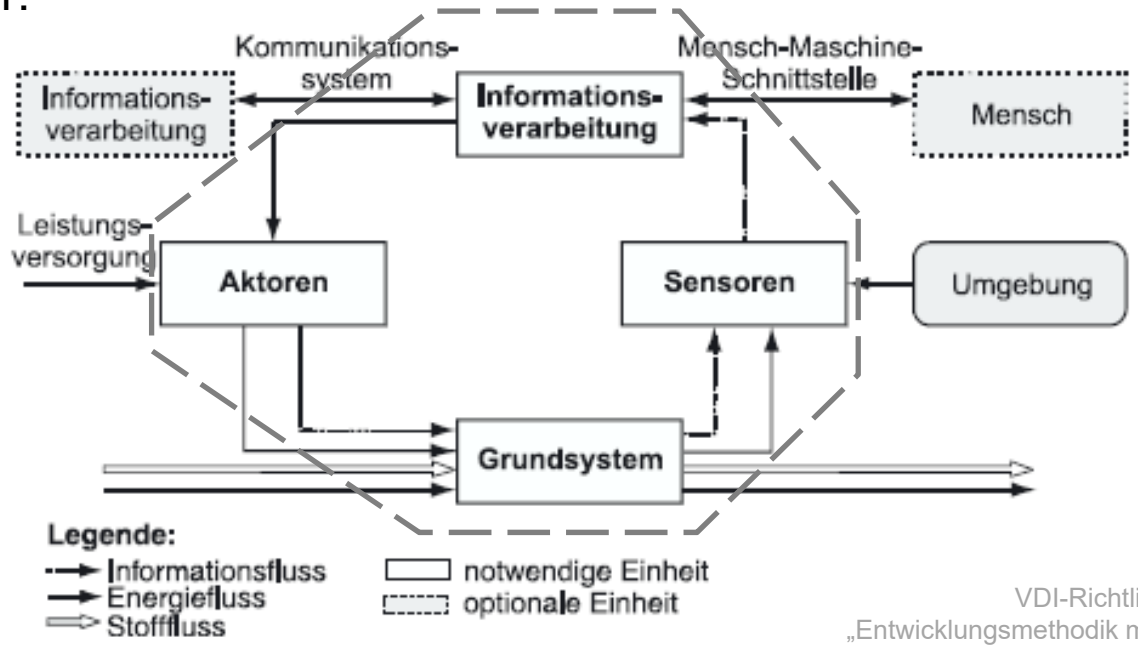
5. Literatur

Grundstruktur:



Quelle:
VDI-Richtlinie 2206, Juni 2004
„Entwicklungsmethodik mechatronischer Systeme“, S. 14 ff.

Grundstruktur:



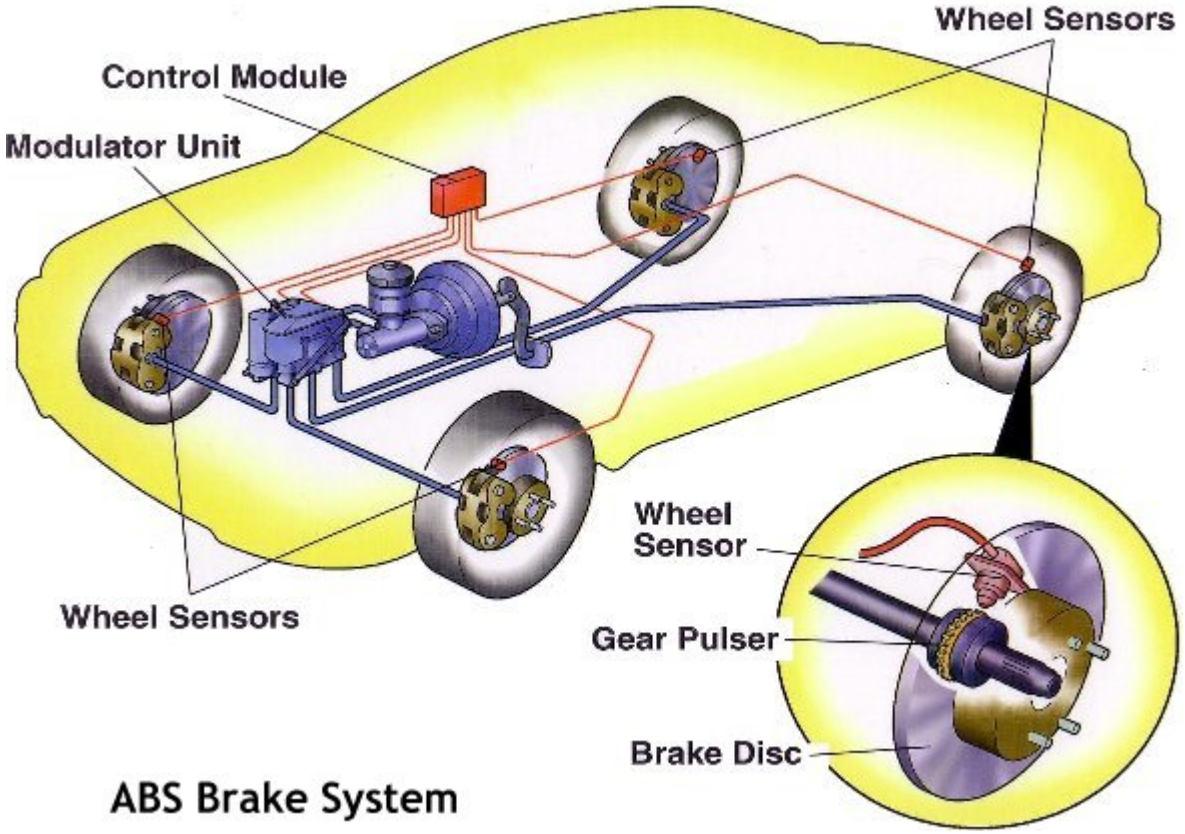
Quelle:
VDI-Richtlinie 2206, Juni 2004
„Entwicklungsmethodik mechatronischer Systeme“, S. 14 ff.

4 Hauptkomponenten:
- Grundsystem
- Sensorik
- Aktorik
- Informationsverarbeitung

! Nicht gleichbedeutend mit „Informationstechnik“ lt. Folie 2. „Informationsverarbeitung“ ist hier nur ein mechatronisches Element.

Schnittstellen zu:
- Umgebung
- (ggf.) Mensch
- (ggf.) andere Informationsverarbeitungseinheiten

Bsp. 1: Bremssystem mit ABS



Quelle: www.auto-repair-help.com/automotive_maintenance/brake_system.php

Fragen:
- Welche sind die 4 Hauptkomponenten?
- Welche Schnittstellen gibt es?
- Welche Flüsse treten auf?

Bsp. 2: Kompressorstation

Prinzip eines Druckluft-kompressors

Quelle: www.kaeser.de/Images/P-651-23-D-tcm6-12414.pdf

Fragen:
- Welche sind die 4 Hauptkomponenten?
- Welche Schnittstellen gibt es?
- Welche Flüsse treten auf?

Bsp. 2: Kompressorstation

Geregelte Kompressor-station

Quellen:
www.kaeser.de/Images/P-651-23-D-tcm6-12414.pdf
www.kaeser.de/Images/P-791-D-tcm6-16946.pdf

Fragen:
- Welche sind die 4 Hauptkomponenten?
- Welche Schnittstellen gibt es?
- Welche Flüsse treten auf?

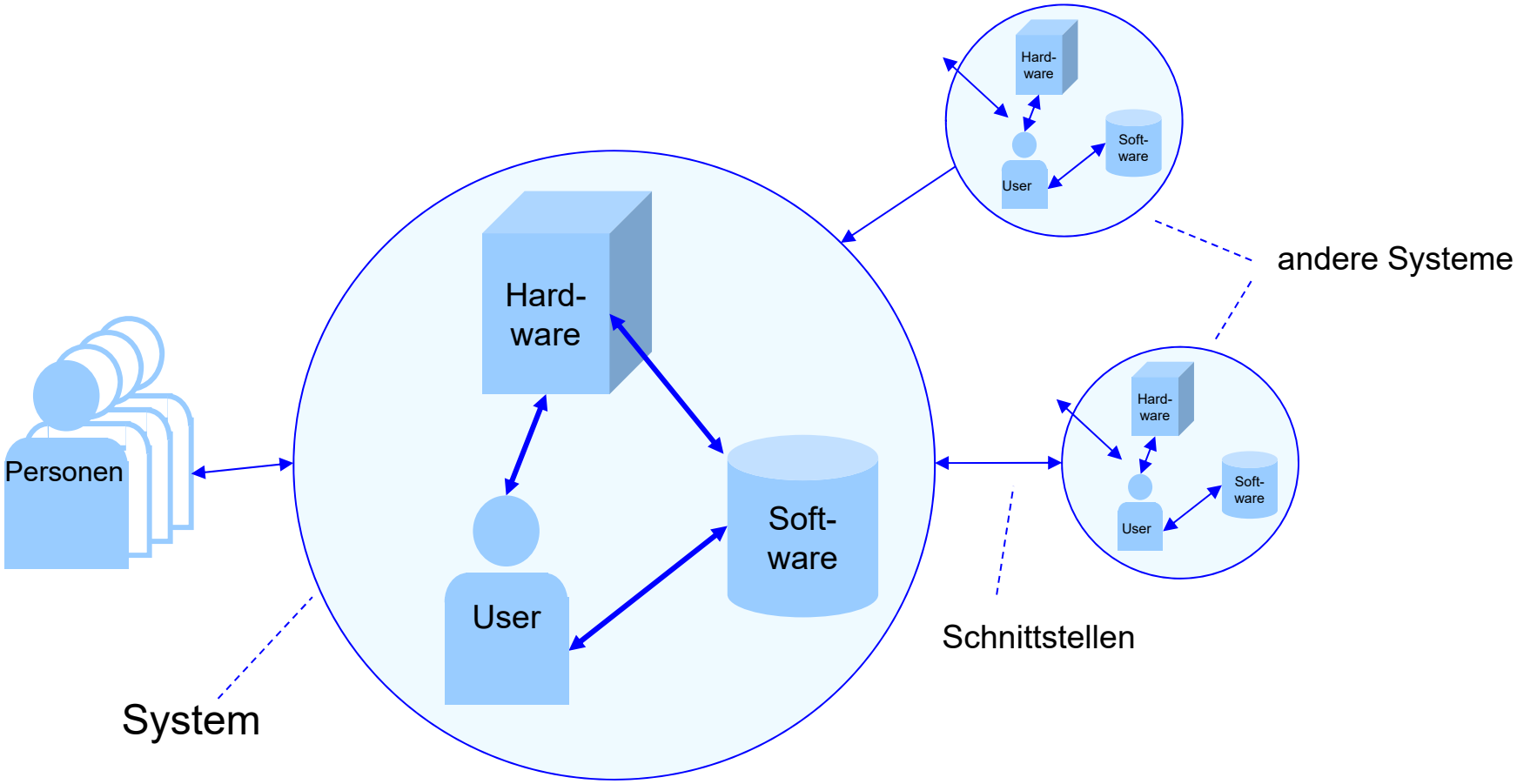
Bsp. 2: Kompressorstation

Geregelte Kompressor-station

Quellen:
www.kaeser.de/Images/P-651-23-D-tcm6-12414.pdf
www.kaeser.de/Images/P-791-D-tcm6-16946.pdf

Fragen:
- Welche sind die 4 Hauptkomponenten?
- Welche Schnittstellen gibt es?
- Welche Flüsse treten auf?

System mit Umgebung:



Fragen:
- Wie passen hier die Komponenten des mechatron. Systems nach VDI 2206 hinein?
- Gibt es strukturelle Unterschiede?

Was ist ein System (DIN 19226) ?

- System: Eine abgegrenzte Anordnung von aufeinander einwirkenden Gebilden. Solche Gebilde können sowohl Gegenstände als auch Denkmethoden und deren Ergebnisse [...] sein. Diese Anordnung wird durch eine Hüllfläche von ihrer Umgebung abgegrenzt oder abgegrenzt gedacht. Durch die Hüllfläche werden Verbindungen des Systems mit seiner Umgebung geschnitten. Die mit diesen Verbindungen übertragenen Eigenschaften und Zustände sind die Größen, deren Beziehungen untereinander das dem System eigentümliche Verhalten beschreibt. Durch zweckmäßiges Zusammenfügen und Unterteilen von solchen Systemen können größere und kleinere Systeme entstehen.

  (DIN 19226: Leittechnik – Regelungstechnik und Steuerungstechnik)

Bestandteile des Systembegriffs

1. Ein System besteht aus verschiedenen Elementen, die auch Komponenten, Teile, Bausteine usw. genannt werden.
2. Diese Elemente weisen Eigenschaften auf, die als Attribute, Funktionen oder Merkmale angegeben werden können.
3. Zwischen den Elementen bestehen Beziehungen, die die funktionelle Verknüpfung der Elemente wiedergeben.
4. Die Elemente mit ihren Eigenschaften und Beziehungen bilden eine abgegrenzte Anordnung.

Ein System besteht aus einer Menge von Elementen, welche Eigenschaften besitzen und welche durch Relationen miteinander verknüpft sind.

Allgemeine technische Definition: System

Ein System ist eine in einem betrachteten Zusammenhang gegebene Anordnung von Gebilden, die miteinander in Beziehung stehen. Diese Anordnung wird aufgrund bestimmter Vorgaben gegenüber ihrer Umgebung abgegrenzt.

[DIN V 19233: Leittechnik – Begriffe]

Elementarmodell zur Beschreibung der Systemeigenschaften

SM 1

Quelle: Patzak

Zustand:

- Ein System hat einen materiellen, energetischen und informationellen Zustand (räumlich und zeitlich)

Räumliches Verhalten / Schnittstellen, Systemgrenzen:

- Ein System tauscht räumlich Materie, Energie und Information über die Systemgrenzen aus

Zeitliches Verhalten / Dynamik:

- Ein System ändert zeitlich seinen Zustand

Diese eigenschaftsbasierte Definition eines Systems entspricht der Definition eines **Prozesses** nach DIN V 19233.

Ein **Prozess** ist die Gesamtheit aller aufeinander einwirkenden Vorgänge in einem System, bei denen

	Materie,	Energie,	Information
	Erhaltungsgrößen		KEINE Erhaltungsgröße
(E) erzeugt,			X
(U) umgeformt,	X	X	X
(T) transportiert,	X	X	X
(S) gespeichert,	X	X	X
(V) vernichtet			X

werden.

Beispiele mechatronischer Prozesse / Systeme

	Materie	Energie	Information
Kompressorstation	(T)	U T S	E U T S V
ICE (fahrender Zug)	T	U (S)	E U T S V
Hochregallager	T S	U	E U T S V
Kaffeeautomat	(U) T S	U T S	E U T S V
ABS-Bremssystem		U	E U T S V

Erzeugen **U**mformen **T**ransportieren **S**peichern **V**ernichten

Elementarfunktionen

Elementar-funktionen / Fluss-größen	Transportieren	Bewahren	Wandeln	Trennen	Kombinieren			Erzeugen / Vernichten
					Materie	Energie	Information	
Materie	Materie transportieren	Materie speichern isolieren konservieren	Materie wandeln umformen	Materie zerlegen	Materie verbinden	Materie mit Energie laden	Materie mit Information versehen	–
Energie	Energie transportieren	Energie speichern isolieren	Energie wandeln	Energie zerlegen	Energie auf Materie laden	Energie mit Energie vereinigen	Energie mit Information versehen	–
Information	Daten übertragen	Daten speichern archivieren	Daten verarbeiten	Daten sortieren	Daten auf Materie codieren	Daten durch Energie codieren	Daten mit Daten verknüpfen	Informationen erzeugen oder vernichten

- Bei der Modellierung eines Systems ist der 1. Schritt die Bestimmung der Systemgrenzen
- Die Systemgrenzen bilden die Schnittstelle zur Umwelt
- Man unterscheidet offene und geschlossene Systeme
- Offenes System:
  - tauscht über die Systemgrenzen Materie, Energie u./o. Information mit der Umwelt aus
- Geschlossenes System:
  - kein Austausch von Materie, Energie und/oder Information mit der Umwelt über die Systemgrenzen
  - ist in einem materiellen, energetischen und informationellen Gleichgewicht
  - geht langfristig in den Zustand minimaler freier Energie bzw. maximaler Entropie über

Fragen:

- Welche Systemgrenzen bieten sich bei der Modellierung der o.g. Beispiele an?
- Handelt es sich jeweils um ein offenes oder um ein geschlossenes System?

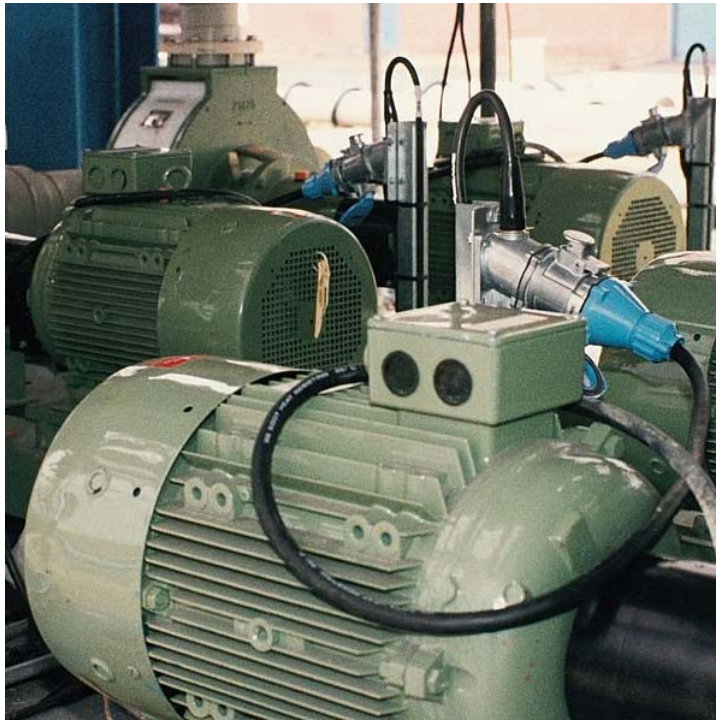
- In der Systemtechnik werden in der Regel offene Systeme betrachtet

! Unterschied zu „Systemen“ aus der Thermodynamik (hier nur Teil d. Hardware): „offen“: Austausch von Materie u. Energie; „geschlossen“: nur Aust. v. Energie; Information nicht betrachtet.

- **Modell**: geeignete Beschreibung eines Systems in Bezug auf
  - seinen **Aufbau** (→ aufbauorientiertes Modell)
  - sein **Verhalten** (→ ablauforientiertes Modell)

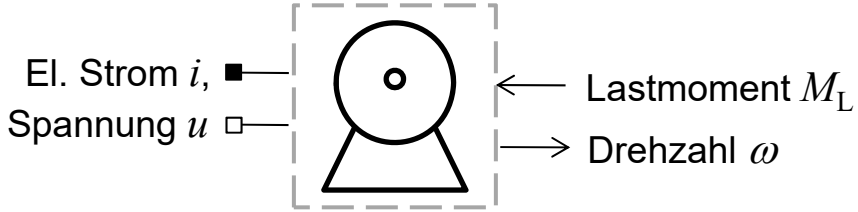
Welche Beschreibung „geeignet“ ist, hängt vom Zweck des Modells ab

Bsp.: Industrieantrieb



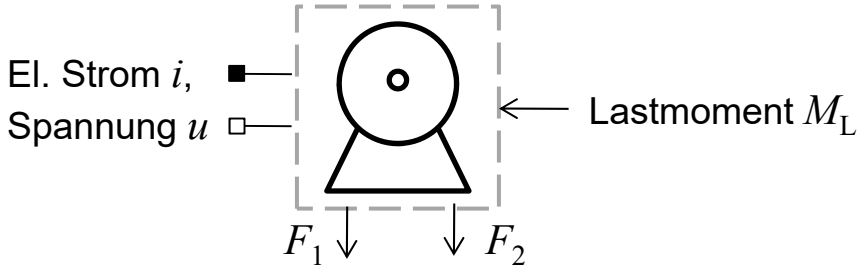
Quelle: http://news.directindustry.de/press/marechal-electric/stillstandszeitkosten-mit-elektrischem-marechal-verringern-9284-35062.html

- E-Motor-Modell des Regelungsingenieurs:



Fragestellung: Zus.hang zw. i, u, M_L, ω?

- E-Motor-Modell des Bauingenieurs:



Fragestellung: Welche Kräfte wirken auf das Fundament?

- **Prinzipien** bei der Systemmodellierung:

Abstraktion

- Verallgemeinerung durch die Vernachlässigung von Eigenschaften
- Konzentration auf Flüsse von Materie, Energie, Daten/Information
- Reduktion der Komplexität durch Weglassen „unwichtiger“ Eigenschaften;
  aber: Entscheidung über „wichtig“ und „unwichtig“ stark von Problemstellung abhängig

Dekomposition

- Strukturierung durch Zerlegung in Bestandteile
- Parallele Bearbeitung der Bestandteile möglich
- Sinnvolle Strukturierung durch Modularisierung, Hierarchisierung

Komposition

- Strukturierte Synthese des Ganzen aus *vorhandenen* Einzelteilen
- Sinnvolle Strukturierung ebenfalls durch Modularisierung, Hierarchisierung
- Verwendung vorhandener, geprüfter Komponenten kann Aufwand reduzieren und
  Fehler vermeiden; aber: sehr frühe Detailentscheidungen nötig

- **Methoden** der Systemmodellierung:

Top-down-Methode

Ausgangspunkt: abstrakte, umfassende Zielvorgabe für das Modell als vollständige Beschreibung

- Zerlegung in Teilsysteme und Konkretisierung in jedem Entwicklungsschritt
- Konsistenz zwischen abstrakten und konkreten Ebenen muss gewährleistet werden

Vorteile:

✓ Benutzerbereich steht im Mittelpunkt der Modellentwicklung
✓ Vollständige, exakte Spezifikationen führen genau zum gewünschten System
✓ Die Gesamtübersicht erleichtert die Bildung geeigneter Abstraktionsebenen und führt zu einer guten Modellstruktur

Probleme:

✗ Gute Fähigkeiten zum abstrakten Denken erforderlich
✗ Realisierungsprobleme werden erst spät erkannt

Bottom-up-Methode

Ausgangspunkt:

Konkrete Teillösungen (Einzelelemente, Teilsysteme) mit bekannter Funktionalität

- Schrittweise Abstraktion und Kombination bekannter Einzelelemente zu größeren Modelleinheiten
- Letzter Schritt: Realisierung des Gesamtsystems

Vorteile:

✓ Konkreter Ausgangspunkt garantiert solide Basis für die Modellentwicklung

✓ Wiederverwendung von vorhandenen Teillösungen wird begünstigt

Probleme:

✗ Schwierigkeiten Ziel tatsächlich zu erreichen

✗ Notwendigkeit einer breiten Basis auf unteren Ebenen

✗ Frühe Festlegung von Implementierungsdetails

Yo-Yo-Methode

Pragmatische Verbindung von **Top-Down- und Bottom-up**-Methode

- Grundsätzliches Vorgehen Top-Down, bei Bedarf Umschalten auf Bottom-Up
- Grund für Wechsel:
  - abstrakte Beschreibung ohne Kenntnis der konkreteren Eigenschaften nicht möglich / sinnvoll
  - aufgrund der abstrakten Beschreibung konnte ein vorhandenes Modul identifiziert werden

Vorteile:

✓ Flexible Anpassung der Vorgehensweise an die sich ergebende Situationen

✓ Reduzierte Anforderungen an das Abstraktionsvermögen

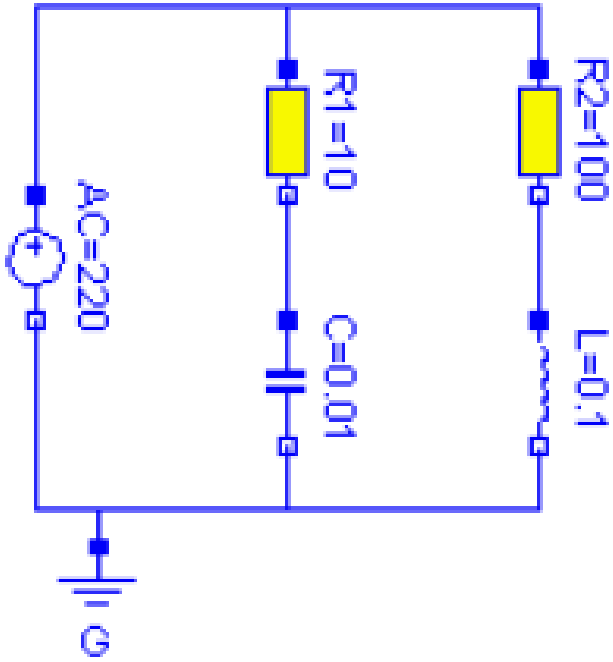
Problem:

✗ Willkürliche Entscheidung, wann Wechsel der Bearbeitungsrichtung

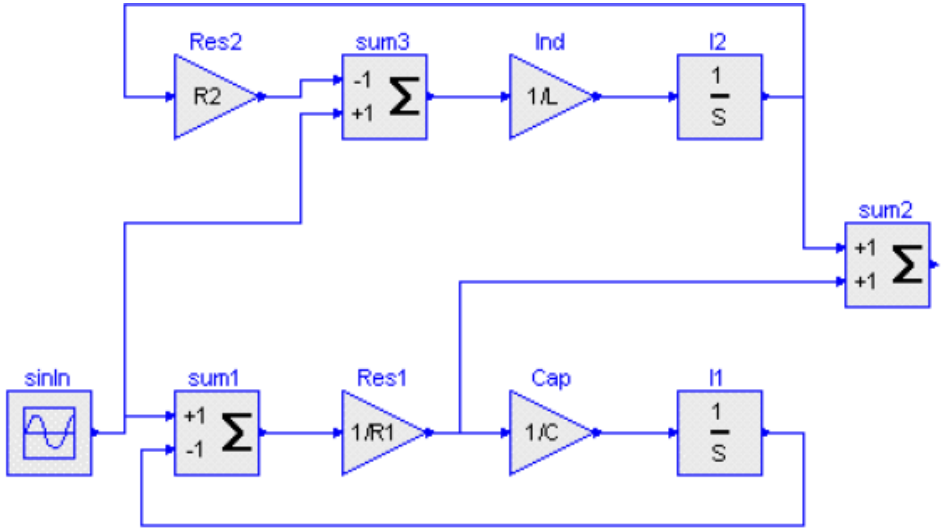
Struktur: Vergleich zwischen Aufbau- und Ablaufstruktur eines Modells

Strukturart / Charakteristika	Aufbaustruktur	Ablaufstruktur
Definition	Darstellung des **Systeminhalts** gegliedert nach sachlichen Zusammenhängen der in einem System enthaltenen Komponenten (Struktur der Komponenten)	Darstellung der **Systemfunktion** als zeitlich/logische Anordnung der in einem System enthaltenen Komponenten (Funktion der Komponenten)
Betrachtungsweise	statisch	dynamisch
Darstellungsform	• Baumstruktur • Hierarchiestruktur • Matrix • *Modelica*-Diagramm	• Gerichteter Graph (Netzplan, Entscheidungsbaum, Petrinetz) • Programmablaufplan • Flussdiagramm • *Simulink*-Blockschaltbild
Analyseverfahren	Aufbauanalyse Systemgliederung	Ablaufanalyse Funktionsanalyse
Charakteristische Analyse- bzw. Synthesebegriffe	„zerfällt in“ „setzt sich zusammen aus“	„liegt nach“, „folgt“ „liegt vor“, „erfordert“, „hängt ab von“

Aufbaumodell
(*Modelica*-Diagramm)



Verhaltensmodell
(*Blockschaltbild*)

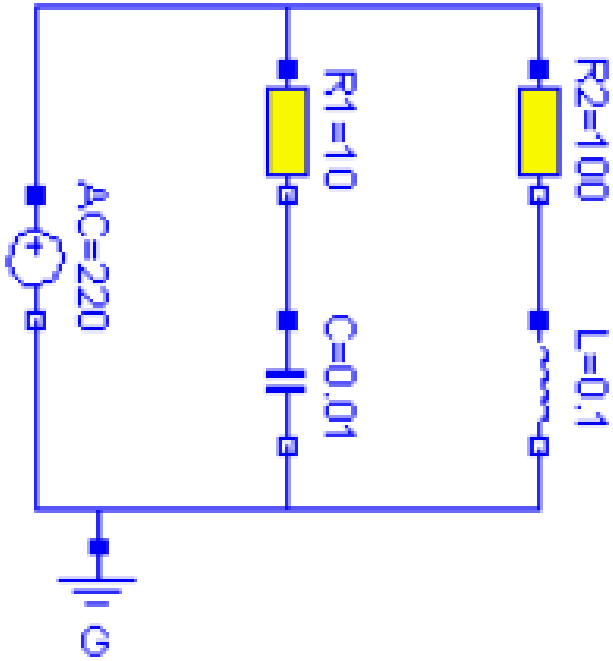


- Zeigt statische Verknüpfung der physischen Komponenten des Schaltkreises

- Mathematische Konstruktion des dynamischen Verhaltens des Gesamtstromes

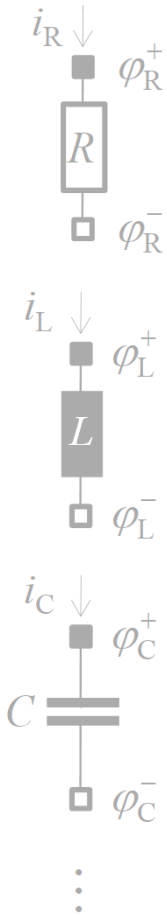
Aufbaumodell

(*Modelica*-Diagramm)



Verhaltensmodell

(Differential-algebraisches Gleichungssystem)



Widerstand 1: $R_1 i_{R1} = \varphi_{R1}^+ - \varphi_{R1}^-$

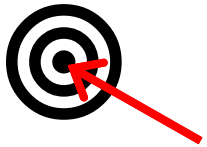
Widerstand 2: $R_2 i_{R2} = \varphi_{R2}^+ - \varphi_{R2}^-$

Spule: $L \dfrac{\mathrm{d}i_L}{\mathrm{d}t} = \varphi_L^+ - \varphi_L^-$

Kondensator: $C \dfrac{\mathrm{d}(\varphi_C^+ - \varphi_C^-)}{\mathrm{d}t} = i_C$

$\vdots$ $\vdots$ $\vdots$

Zusammenfassung und Lernkontrolle



- Den Begriff *System* kennen:

  Spezifisch mechatronischer Systembegriff, allgemeiner technischer Systembegriff

- Technischer *Prozess*:

  Flussgrößen und Elementarfunktionen kennen und anwenden können

- *Modellierung* von Systemen:

  Allgemeine Prinzipien und Methoden kennen,
  Unterschied zwischen Aufbau- und Ablaufstruktur eines Modells verstehen

Systemmodellierung
Teil 1: Ereignisdiskrete Systeme

Kapitel 2
Ereignisdiskrete Systeme

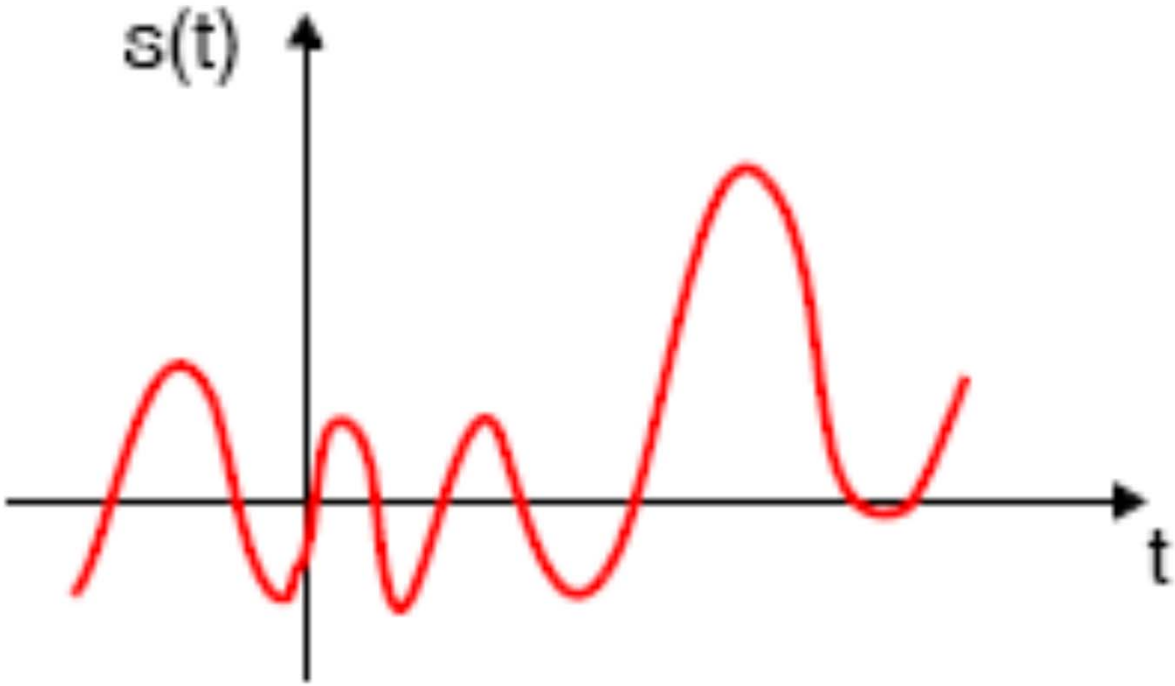
UNIVERSITÄT DES SAARLANDES

AES
Automatisierungs- und Energiesysteme

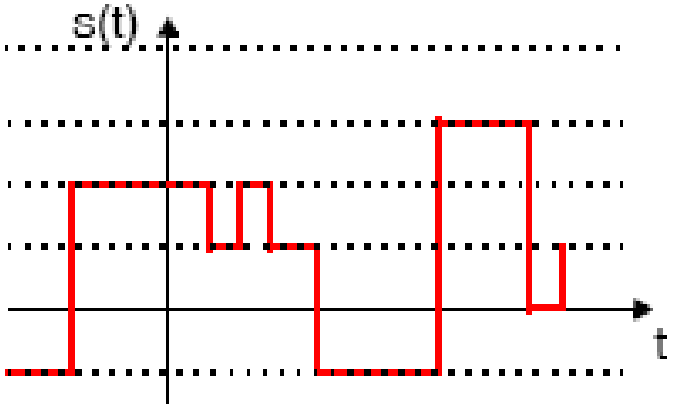
Kapitel 2: Ereignisdiskrete Systemmodelle

- Ereignisbegriff
  - Beispiele
  - Unterscheidung kontinuierlicher und ereignisdiskreter Systeme
- Modellierung mit Automaten
  - Autonome Automaten
  - Determinismus → nichtdeterministische Automaten
  - Kopplung → E/A-Automaten
  - Erweiterung zu Statecharts
  - Umsetzung in Stateflow (*Matlab*®/*Simulink*® Toolbox)

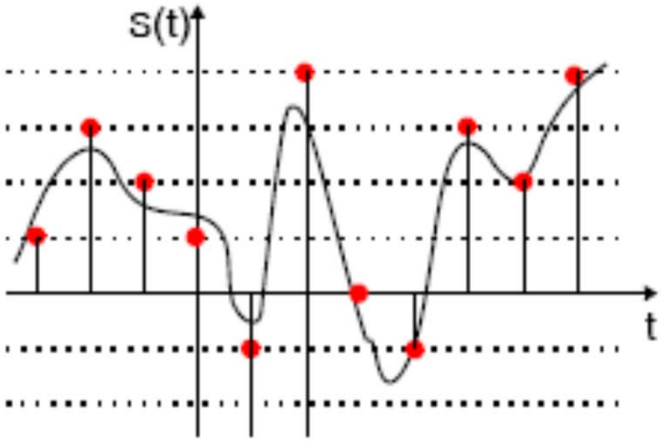
- Kontinuierlich in Wert und Zeit (Inhalt von Teil 2 der VL)
  - Sowohl Eingaben als auch Ausgaben
  - Beispiele: Temperatur, Position, Geschwindigkeit



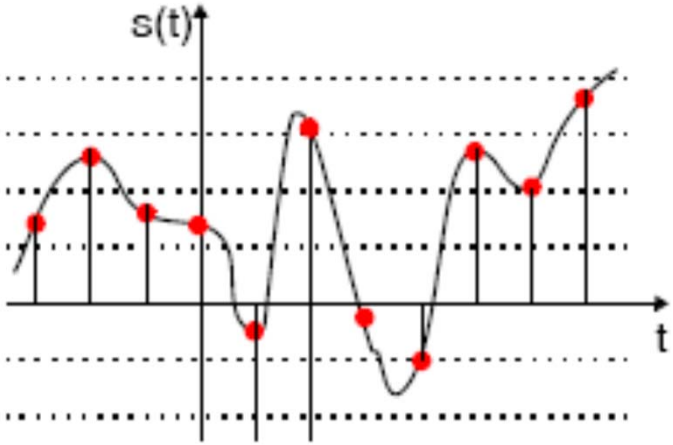
quantisiert (wertdiskret)



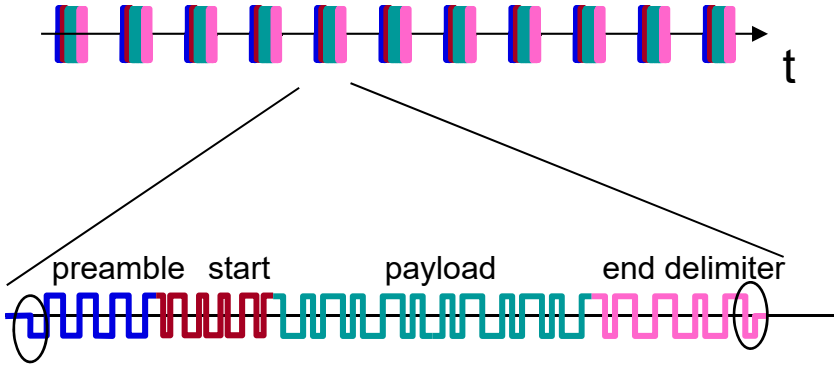
digital: quantisiert und abgetastet



abgetastet (zeitdiskret)



Nachricht: kodiert und serialisiert



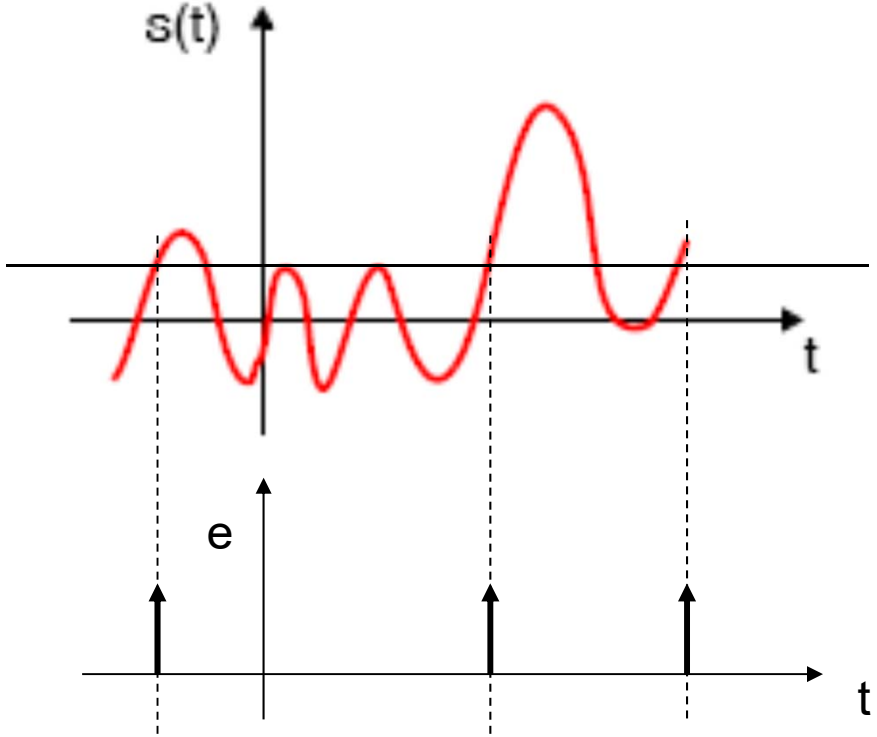
Was ist ein Ereignis?

- Signale (diskret oder kontinuierlich) sind zu jeder Zeit definiert
- Dies gilt auch für zeitdiskrete und serialisierte Darstellungen
- **Die Information liegt immer in der Amplitude des Signals**

- Ein Ereignis passiert
- Ein Ereignis hat keine Dauer
- **Die Information liegt im Auftreten des Ereignisses**
- Bei Systemen mit vielen Ereignissen ist meist nur die logische, nicht aber die exakte temporale, Abfolge interessant

- Ereignisse können in wertdiskrete Signale transformiert werden (wenn Zeitinformation vorhanden ist)
- Signale (ihre Änderungen) können in Ereignisse transformiert werden
  - Am natürlichsten: Binär-Signal B1 wird zu einer Menge von zwei Ereignissen konvertiert: E1= Wechsel auf Null, E2 = Wechsel auf Eins

Ereignisse in mechatronischen Systemen



Maschinenbau
- Steuerung von Maschinen
- Fertigungstechnische Anlagen

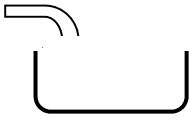
Verfahrenstechnik
- Batch-Prozesse

Elektro- und Informationstechnik
- Energieversorgungssysteme
- Verkehrstechnik

Informatik
- Kommunikationsnetze
- Embedded Systems

- Entstehung von Ereignissen
  - Abstraktion eines physikalischen Systems (schnelle Übergänge)
  - Ökonomische Gründe bei der technischen Implementierung (Grenzwertschalter)
- Ereignisdiskrete System sind i.d.R. „man-made“-Systems
  - Das ereignisdiskrete System entsteht oft erst durch die Informationsverarbeitung
  - Modellierung des unbeeinflussten Systems daher schwierig

- Systeme (genauer: System-Modelle) werden nach dem Charakter der darin vorkommenden Signale als ereignisdiskrete oder kontinuierliche Systeme bezeichnet.
- Ob ein System als diskretes oder kontinuierliches System behandelt wird, entscheidet sich oft erst bei der Modellierung – abhängig vom Modellierungszweck.
- Beispiel:
  - Füllstand *h* im Behälter



a) manchmal interessiert der genaue Wert des Füllstands

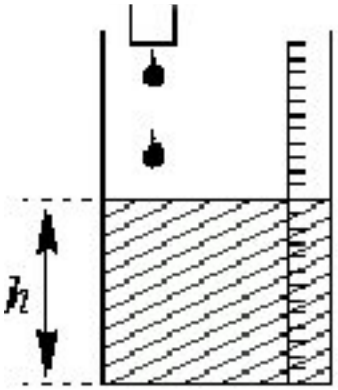
→ *h* als kontinuierliche Größe

b) manchmal interessiert nur eine Unterscheidung („voll“ / „nicht voll“)

→ *h* als diskrete Größe

- In jedem Fall ist *h* eine <u>Zustandsgröße</u> des Systems.

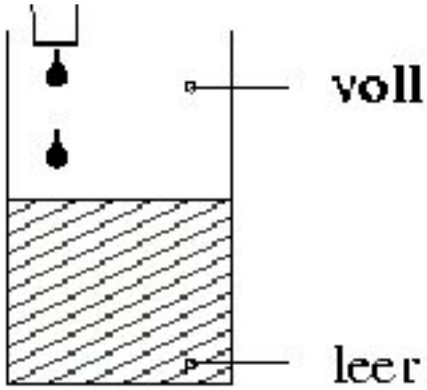
- Beispiel: Zustandsgröße „Füllstand“ des Behälters



Realisierung als Schwimmer mit elektro-mechanischem Wandler

kont. Zustandsvariable

$$0\ \text{cm} \leq h \leq 100\ \text{cm}$$



Realisierung als Schwimmer mit Schalter (Öffner und Schließer)

diskrete Zustandsvariable

$$h \in \{\text{voll, leer, gefüllt, fehler}\}$$

mit

$$\text{gefüllt} = \text{NOT(leer) AND NOT(voll)}$$
$$\text{fehler} = \text{leer AND voll}$$

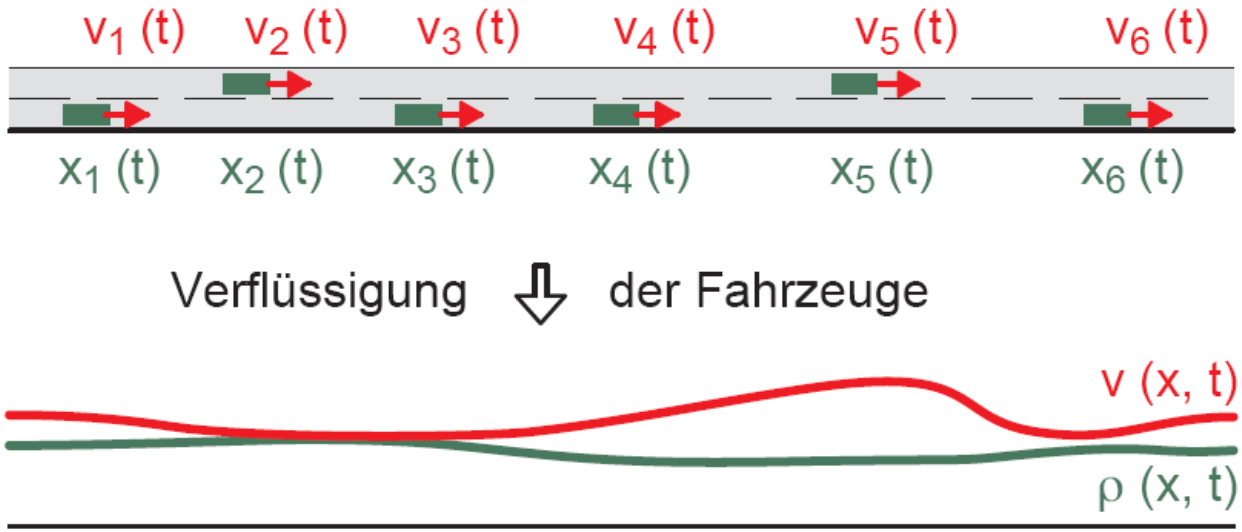
- Beispiel: Modellierung des Verkehrsflusses mit der Zustandsgröße „Fahrzeugdurchsatz“

a) manchmal interessiert nur der durchschnittliche Wert

→ *v* als kontinuierliche Größe

b) manchmal interessiert die genaue Zahl der vorbeifahrenden Fahrzeuge

→ *v* als diskrete Größe



mikroskopische Verkehrsmodellierung

makroskopische Verkehrsmodellierung

Systeme: diskret oder kontinuierlich

- Beispiel: Modellierung eines Raumes in Bezug auf seine Temperatur
- Relevante physikalische Größen und Signalcharakter

➢ Raumtemperatur	kontinuierlich
➢ Außentemperatur	kontinuierlich
➢ Temperatur im Heizkessel	kontinuierlich
➢ Vorlauftemperatur	kontinuierlich
➢ Heizkörperventilöffnung	kontinuierlich
➢ Vorlaufpumpenstellung	diskret
➢ Brennerleistung	kont. oder diskret
➢ Fensteröffnung	diskret
➢ Personen im Raum	diskret

→ Modellierung als kontinuierliches System sinnvoll, einzelne diskrete Signale integrieren.

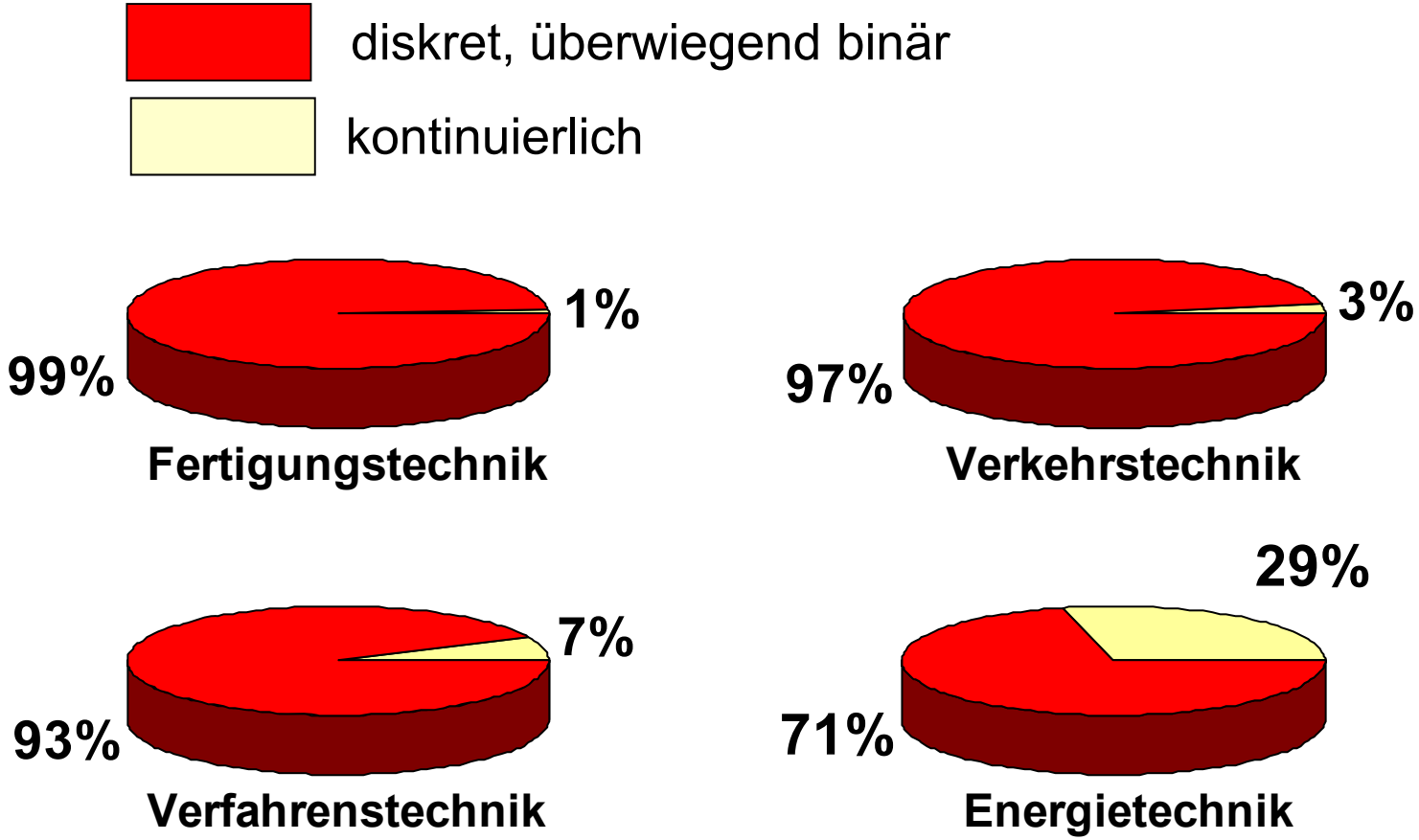
- Beispiel: Modellierung eines Personenaufzugs

- Relevante physikalische Größen und Signalcharakter

➢ vertikale Position	diskret (Etagen)
➢ Türöffnung	diskret
➢ Türblockierung	diskret
➢ Fahrkorbgewicht	diskret (Überlast)
➢ Motortemperatur	diskret (Überlast)
➢ Antriebsleistung	diskret oder kont.
➢ Fahrtrichtung	diskret
➢ Bremsleistung	diskret

→ Modellierung als diskretes System sinnvoll, ggf. Motor als kontinuierliches Teilsystem modellieren.

- Anteil binärer und kontinuierlicher Messgrößen in verschiedenen **technischen (i.S.v. automatisierten)** Systemen



→ Achtung: Anzahl der Signale bestimmt nicht notwendigerweise die Komplexität der beschriebenen Funktion.

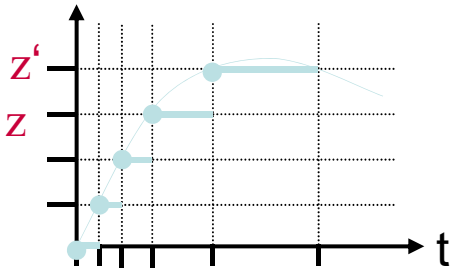
Quelle: Schnieder, Methoden der Automatisierung, S. 17

Systeme: ereignisdiskret oder kontinuierlich

- Gemeinsamkeiten der betrachteten Systeme: sie sind
  - dynamisch,
  - kausal,
  - zeitinvariant.
- Unterschiede ergeben sich hinsichtlich
  - Modellbeschreibung
    - Diskrete Systeme werden durch Zustandsübergangsrelationen beschrieben, kontinuierliche z.B. mit Differentialgleichungen.
  - Linearität
    - Kontinuierliche Systeme sind oft linear (oder können linearisiert werden), diskrete nicht.
  - Nichtdeterminismus
    - Diskrete Systeme verhalten sich häufig nichtdeterministisch, dies muss im Modell abgebildet werden.
      Kontinuierliche Systeme werden i.d.R. deterministisch modelliert.

Zustände in ereignisdiskreten Systemen

- Ein ereignisdiskretes System wechselt seinen Zustand, wenn ein Ereignis eintritt.
- Für die Abläufe und das Verhalten ist die Reihenfolge der Ereignisse und damit die Reihenfolge der Zustandsübergänge entscheidend.
- Die zeitlichen Abstände zwischen den Ereignissen spielen keine (oder nur eine untergeordnete) Rolle.



ereignis-diskret

- Automaten sind eine gut geeignete Form, um ereignisdiskrete Systeme zu beschreiben.
- Ein (autonomer, endlicher) Automat besteht aus
  - einer (endlichen) Menge von Zuständen $Z = \{z_1, z_2, z_3, \ldots\}$,
  - einem Anfangszustand $z_0 = z(0)$,
  - einer Zustandsübergangsfunktion f,
    die beschreibt, aus welchem Zustand ein Übergang in welchen anderen Zustand möglich ist.

$$f: \quad Z \rightarrow Z$$

$$z(k+1) = f(z(k)) \quad k = 0,1,\ldots$$

Beispiel:

$$z(k+1) = f(z(k)) = \begin{Bmatrix} z_2 \ \text{für}\ (z(k) = z_1) \\ z_3 \ \text{für}\ (z(k) = z_2) \\ z_1 \ \text{für}\ (z(k) = z_3) \end{Bmatrix}$$

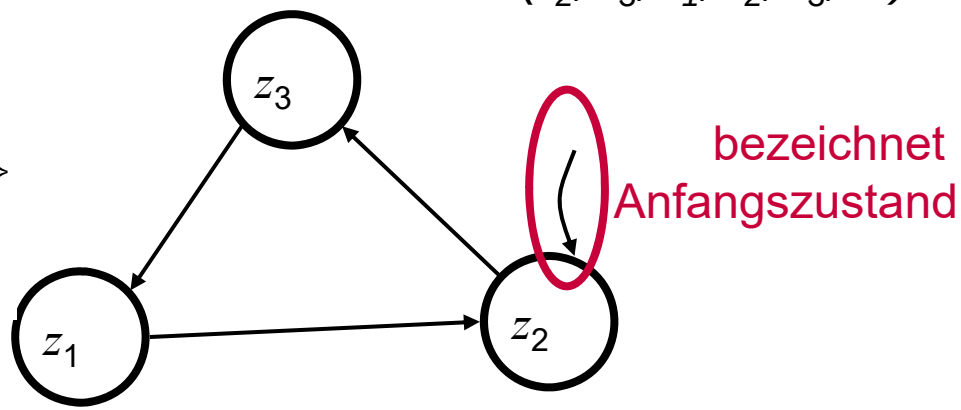
- alternative Darstellungen:
  - als Automatengraph
  - als Tabelle
  - als Matrix

- Automat in Graphendarstellung
- Ein Automat besteht aus
  - einer (endlichen) Menge von Zuständen $Z = \{z_1, z_2, z_3, \ldots\}$,
  - einem Anfangszustand $z_0 = z(0)$,
  - einem Graphen, bei dem jede gerichtete Kante einen erlaubten Zustandsübergang beschreibt.
- *$Z = (z(k=0), z(k=1), z(k=2), z(k=3), \ldots)$ ist die Folge von Zuständen, die durchlaufen werden.*

Beispiel:

$$z(k+1) = \begin{Bmatrix} z_2 \ \text{für}\left(z(k) = z_1\right) \\ z_3 \ \text{für}\left(z(k) = z_2\right) \\ z_1 \ \text{für}\left(z(k) = z_3\right) \end{Bmatrix}$$

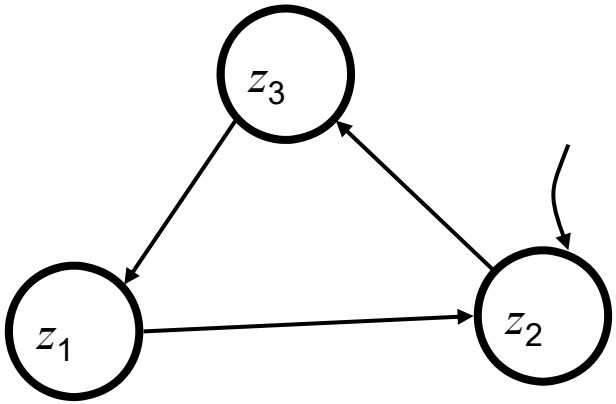
$Z = (z_2, z_3, z_1, z_2, z_3, \ldots)$



- Automat in Tabellendarstellung
- Ein Automat besteht aus
  - einer (endlichen) Menge von Zuständen $Z = \{z_1, z_2, z_3, \ldots\}$,
  - einem Anfangszustand $z_0 = z(0)$,
  - einer Tabelle,
    die beschreibt, aus welchem Zustand ein Übergang in welchen anderen Zustand möglich ist.

Beispiel:

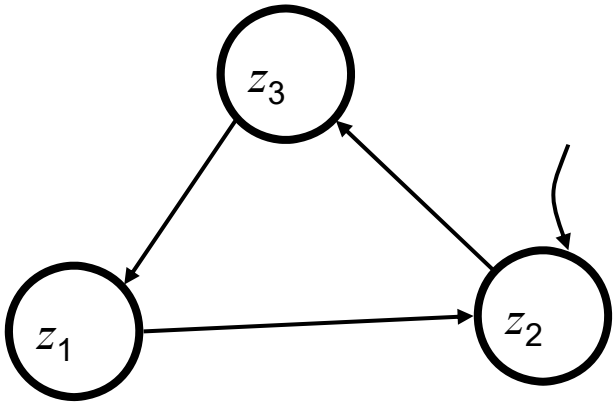
$z(k)$	$z(k+1)$
z_1	z_2
z_2	z_3
z_3	z_1



- Automat in Matrixdarstellung
- Ein Automat besteht aus
  - einer (endlichen) Menge von Zuständen $Z = \{z_1, z_2, z_3, \ldots\}$,
  - einem Anfangszustand $z_0 = z(0)$,
  - einer Zustandsübergangsmatrix F,
    die beschreibt, aus welchem Zustand ein Übergang in welchen anderen Zustand möglich ist:
    - Wenn ein Übergang von z_n nach z_m möglich ist, steht in der Matrix in der n-ten Spalte und der m-ten Zeile eine „1“, ansonsten eine „0“.

Beispiel:

$z(k)$	$z(k+1)$
z_1	z_2
z_2	z_3
z_3	z_1



$$F = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

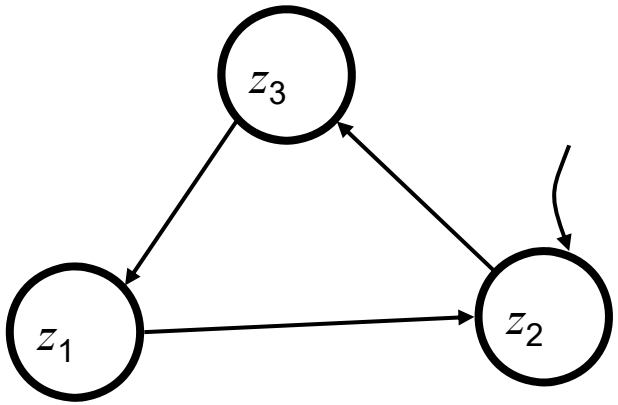
- Der Vektor p hat so viele Komponenten wie der Automat Zustände.
  - Die Komponente $p_i(k)$ gibt an, ob sich der Automat zum Zeitpunkt k
    - im Zustand i befindet ($p_i(k) = 1$)
    - oder nicht ($p_i(k) = 0$).
- Die Zustandsübergangsfunktion F ist eine Matrix.
  - Das Element f_{ij} ist genau dann gleich eins, wenn der Automat vom Zustand j in den Zustand i übergeht.
  - Alle anderen Elemente sind gleich null.
- In der Matrixdarstellung ergibt sich
  - der Zustand $p(k+1)$
  - aus der Multiplikation von $p(k)$
  - mit der Matrix F:

$$p(k + 1) = F\, p(k), \qquad (k = 0, 1, 2,...)$$

- Der Anfangszustand p_0 enthält bis auf die zu z_0 gehörende Stelle nur Nullen und bei z_0 eine Eins.
- Die Trajektorie *z(k)* des Automaten wird dann durch die Folge der Vektoren *p(k)* beschrieben, die man durch wiederholte Multiplikation des Vektors *p(k)* mit der Matrix *F* erhält: **Z** = *(p(0), p(1), p(2), ...)*

Beispiel:

$$F = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \quad p(0) = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \qquad p(1) = F\ p(0) = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$



$$\mathbf{Z} = \left(\begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \ldots \right)$$

Modellbildung: Beispiel Stanze

- Systembeschreibung:
  - In einer Stanze werden Blechteile in mehreren aufeinander folgenden Arbeitsgängen gestanzt, wobei
    - beim ersten Arbeitsschritt das Blech von einer Blechrolle abgeschnitten wird, danach gedreht wird,
    - und im zweiten Arbeitsschritt noch einmal gestanzt wird.
    - Danach wird das Blechteil abtransportiert.
- Die Stanze soll als Zustandsautomat in modelliert werden.

- Festlegung der Zustände

Zustand	Beschreibung
1	Die Stanze steht in der Ausgangsposition.
2	Das Blech befindet sich unter der Stanze.
3	Das Blech ist abgetrennt.
4	Die Stanze ist nach dem ersten Arbeitsgang geöffnet.
5	Das Werkstück ist neu ausgerichtet.
6	Das Werkstück ist umgeformt.
7	Die Stanze ist nach der Umformung des Werkstückes geöffnet.

- Festlegung der Zustandsübergänge

- Festlegung des Anfangszustands

- Festlegung der Zustände

Zustand	Beschreibung
1	Die Stanze steht in der Ausgangsposition.
2	Das Blech befindet sich unter der Stanze.
3	Das Blech ist abgetrennt.
4	Die Stanze ist nach dem ersten Arbeitsgang geöffnet.
5	Das Werkstück ist neu ausgerichtet.
6	Das Werkstück ist umgeformt.
7	Die Stanze ist nach der Umformung des Werkstückes geöffnet.

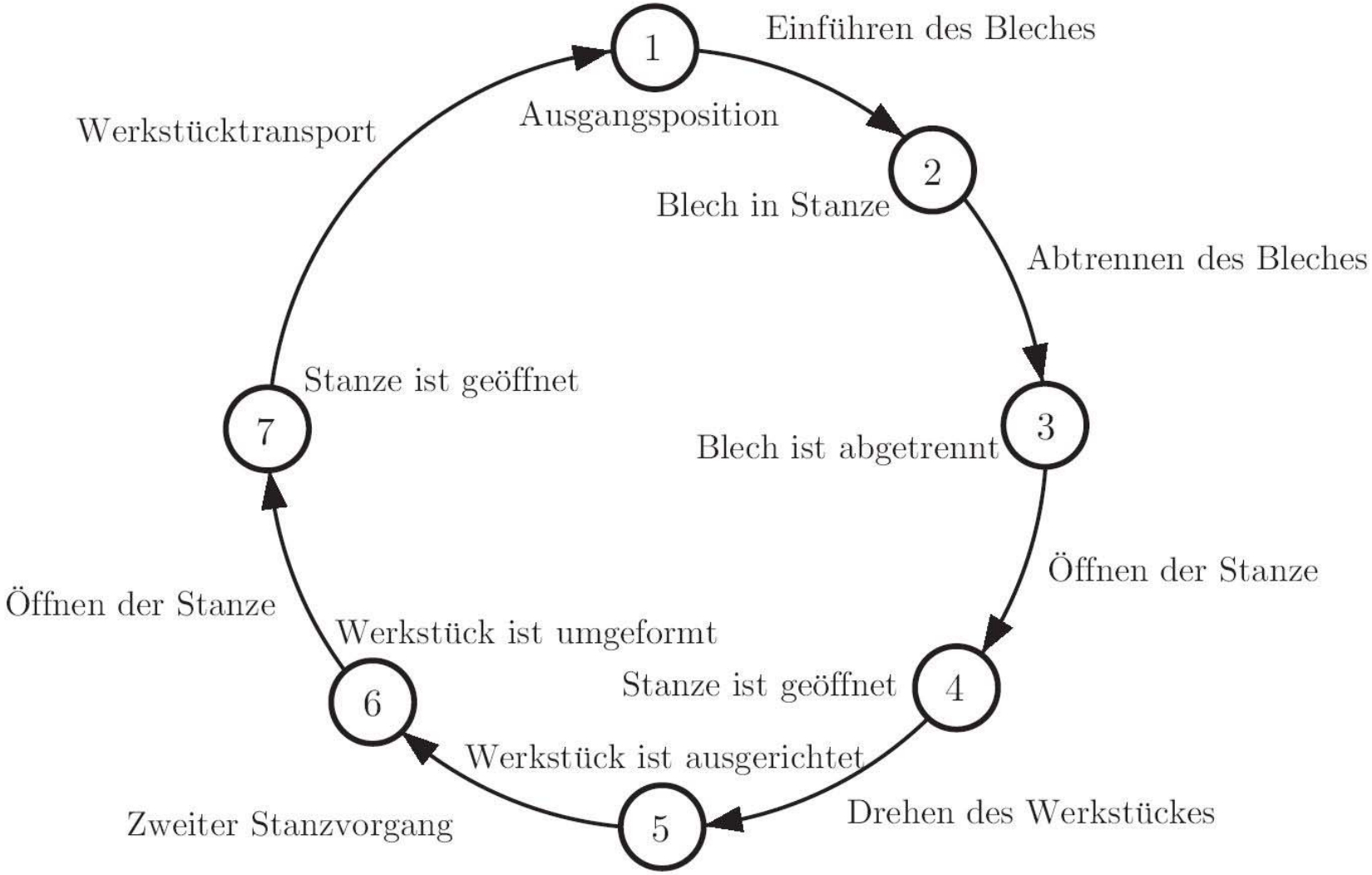
- Festlegung der Zustandsübergänge

z'	z	Beschreibung
2	1	Einführen des Bleches
3	2	Erster Stanzvorgang (Abtrennen des Bleches)
4	3	Öffnen der Stanze
5	4	Drehen des Werkstückes
6	5	Zweiter Stanzvorgang (Umformen des Bleches)
7	6	Öffnen der Stanze
1	7	Transport des Werkstückes aus der Stanze in das Lager

- Festlegung des Anfangszustands

$z(0) = z_1$

- Modell als Automatengraph



- Der Automat, der die Stanze beschreibt, lautet in Matrixdarstellung:

$$\begin{pmatrix} p_1(k+1) \\ p_2(k+1) \\ p_3(k+1) \\ p_4(k+1) \\ p_5(k+1) \\ p_6(k+1) \\ p_7(k+1) \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} p_1(k) \\ p_2(k) \\ p_3(k) \\ p_4(k) \\ p_5(k) \\ p_6(k) \\ p_7(k) \end{pmatrix}, \qquad \begin{pmatrix} p_1(0) \\ p_2(0) \\ p_3(0) \\ p_4(0) \\ p_5(0) \\ p_6(0) \\ p_7(0) \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

- Daraus erhält man die Zustandsfolge *z(k)*:

$$\mathbf{z} = \left(\begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \ldots \right)$$

- Ein Roboter nimmt abwechselnd Teile von Band 1 und Band 2 und legt sie auf das Ablageband.
- Vom Roboter sind folgende Zustände relevant:
  - Roboter über Band 1
  - Roboter über Band 2
  - Roboter über Ablageband
- Vom Greifer sind folgende Zustände relevant:
  - Greifer geöffnet
  - Greifer geschlossen (hält Werkstück)
- Modellieren Sie das Gesamtsystem „Roboter + Greifer“ als Automatengraph.

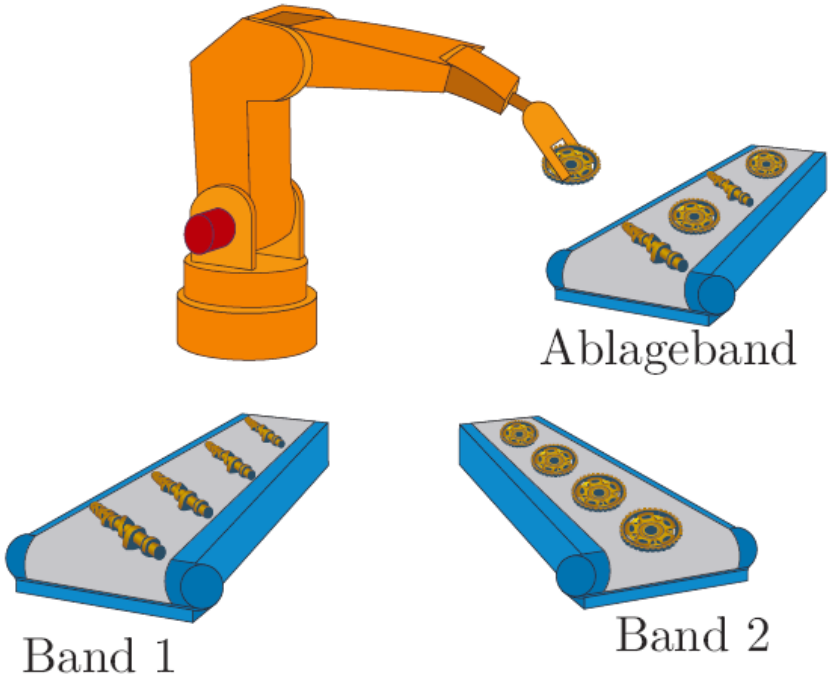
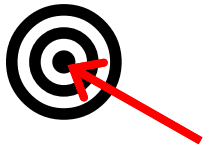


Abb. 1.3: Sortieren von Werkstücken durch einen Roboter

J. Lunze: Ereignisdiskrete Systeme, Oldenbourg-Verlag 2006

Zusammenfassung



- Begriff der ereignisdiskreten Systeme
- Unterscheidung kontinuierliche und ereignisdiskrete Systeme
  - Gemeinsamkeiten Unterschiede
- Wahl der „richtigen“ Modellform
- Autonome, endliche Automaten
  - Definition
  - Darstellungsmöglichkeiten
  - Beispiele

Aufgabe zur Nachbereitung: Ampel

Modellieren Sie eine Ampel als Automat

a) Ampel normal für Autos (nur eine Fahrtrichtung, keine Kreuzung)
b) Ampel plus Fußgängerampel (nur eine Fahrtrichtung, keine Kreuzung)
c) Ampel-Kreuzung für Autos (keine Fußgänger)

Systemmodellierung
Teil 1: Ereignisdiskrete Systeme

Kapitel 2
Ereignisdiskrete Systeme (II)

Modellierung mit autonomen Automaten

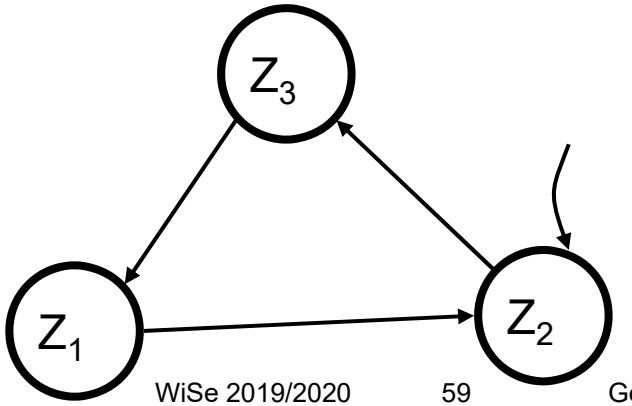
UNIVERSITÄT DES SAARLANDES

AES
Automatisierungs- und Energiesysteme

- Bei den bisher betrachteten Beispielen handelte es sich um deterministische Automaten,
  d.h. in jedem Zustand war eindeutig beschreibbar, welcher Zustand als nächstes eingenommen wird.
  - Im Automatengraph führt von jedem Knoten genau eine Kante weg.
  - In der Tabelle gibt es für jeden Zustand genau eine Zeile mit genau einem Folgezustand.
  - In der Matrix steht in jeder Spalte genau eine „1“.

Beispiel:

$z(k)$	$z(k+1)$
z_1	z_2
z_2	z_3
z_3	z_1

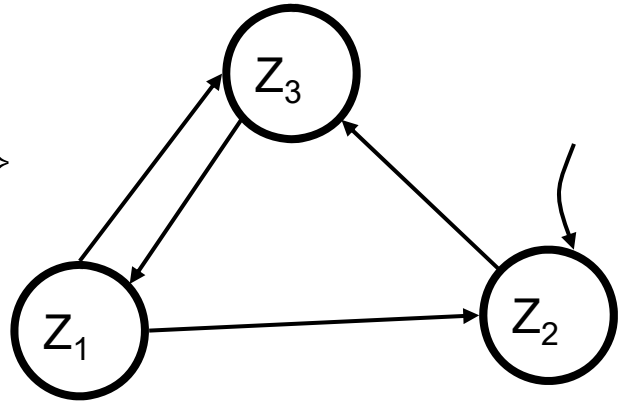


$$F = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

- Bei realen Systemen ist oft nicht eindeutig angebbar, welcher Zustand als nächstes eingenommen wird.
- Solche Systeme lassen sich als <u>nichtdeterministische</u> Automaten beschreiben.
  - Aus der Zustandsübergangs<u>funktion</u> wird eine Relation.
  - Im Automatengraph führt von (mindestens) einem Knoten mehr als eine Kante weg.
  - In der Matrix steht in (mind.) einer Spalte mehr als eine „1“.
- Eine eindeutige Zustandsfolge ist nicht mehr angebbar.

Beispiel:

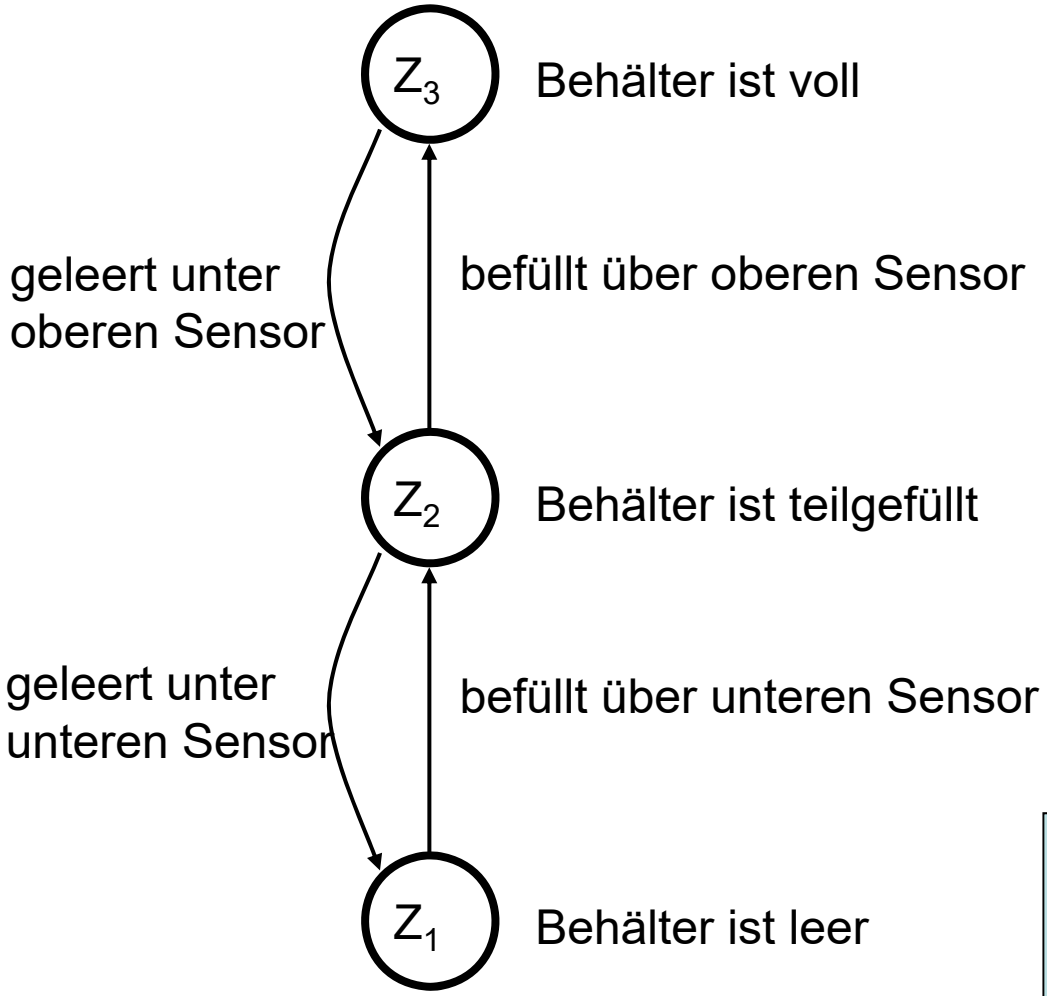
$$z(k+1) = \begin{Bmatrix} Z_2 \vee Z_3 \ \textit{für}\left(z(k) = Z_1\right) \\ Z_3 \ \textit{für}\left(z(k) = Z_2\right) \\ Z_1 \ \textit{für}\left(z(k) = Z_3\right) \end{Bmatrix}$$

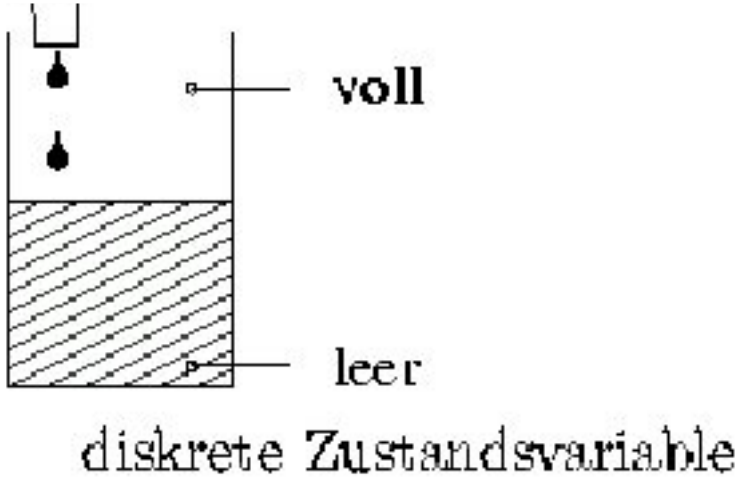


$$F = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \end{pmatrix}$$

Beispiel: Zustände eines Behälters

- Modellierung als Automatengraph





Von Zustand Z_2 aus sind zwei Zustandsübergänge möglich

→ nichtdeterministisch

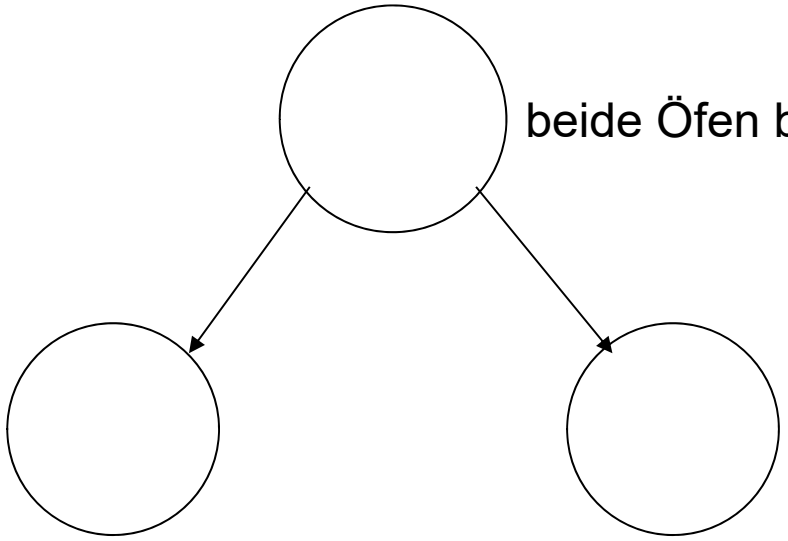
- Nichtdeterminismus wird in ereignisdiskreten technischen Systemen verursacht
  a) durch Unkenntnis über interne Abläufe:
     Beispiel: Werkstückvorwärmung

linker Ofen

Ablaufdauer = xx s

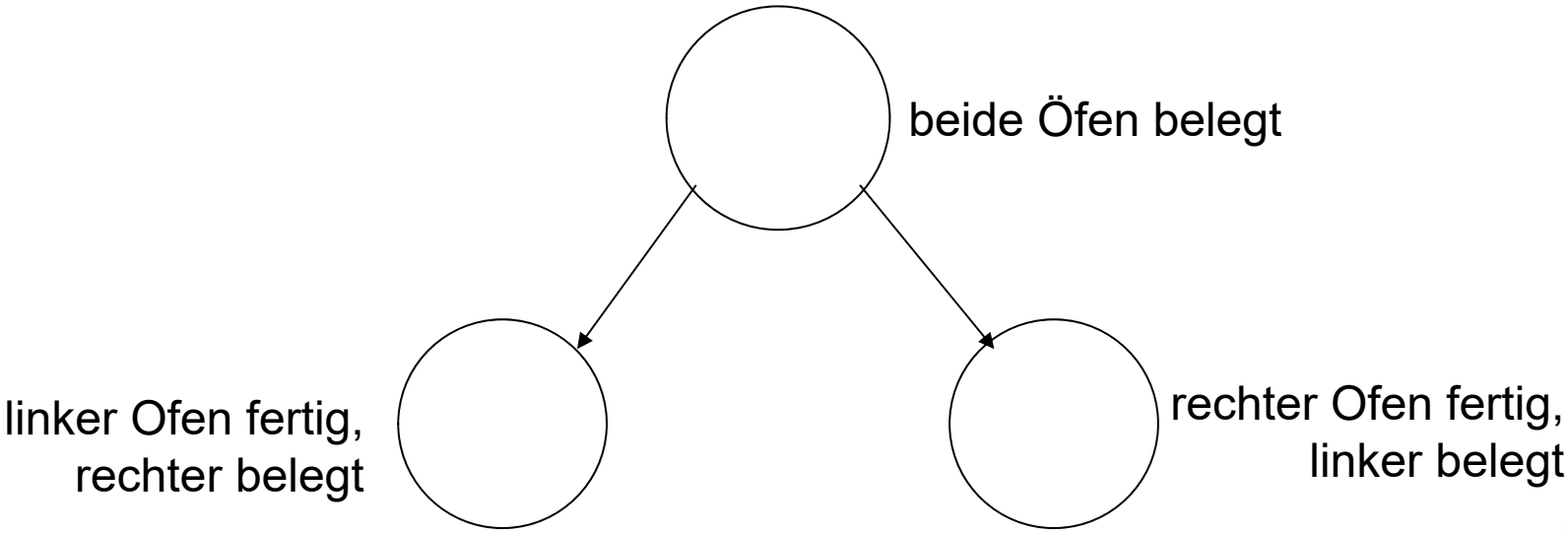
rechter Ofen

Ablaufdauer = xx s



- Nichtdeterminismus wird in ereignisdiskreten technischen Systemen verursacht
  b) durch kleine Schwankungen bei parallelen Abläufen:
  Beispiel: Werkstückvorwärmung

linker Ofen	rechter Ofen
Ablaufdauer = 23,8 s	Ablaufdauer = 24,0 s
Ablaufdauer = 24,2 s	Ablaufdauer = 24,1 s



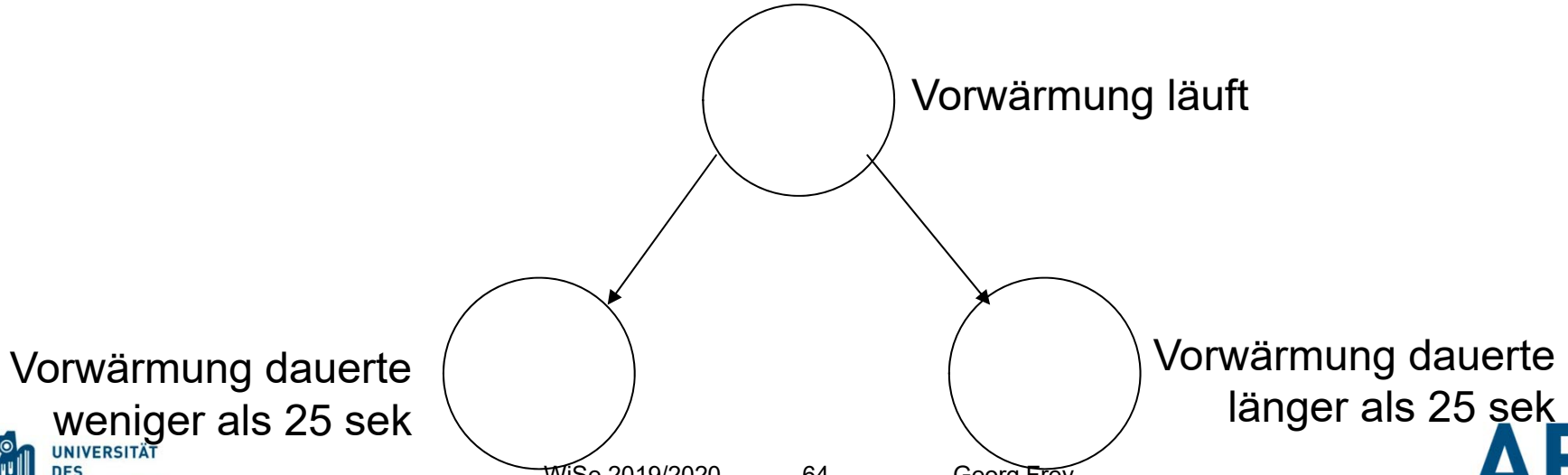
- Nichtdeterminismus wird in ereignisdiskreten technischen Systemen verursacht
  b) durch Störungen: Beispiel: Werkstückvorwärmung

linker Ofen

Ablaufdauer = 23,8 s

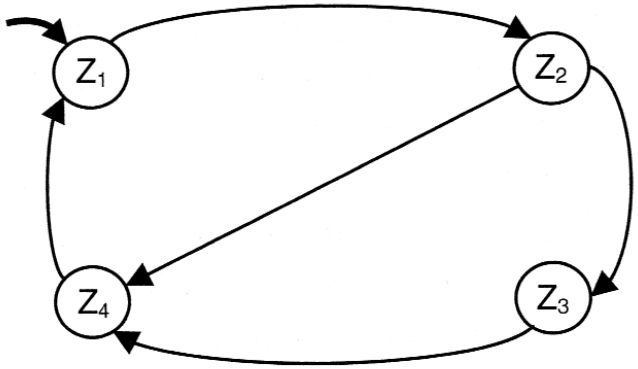
Ablaufdauer = 24,7 s

Ablaufdauer = 25,1 s



- Die Beschreibung eines nichtdeterministischen Automaten über die Zustandsfolge ist nicht hilfreich
  - Für dieses Beispiel:

$z(0) = Z_1$	$z(1) = Z_2$	$z(2) = ?$



→ Verhaltensvorhersage:

→ Man kann nur sagen, dass es für einen bestimmten endlichen Horizont von Schritten eine endliche Menge von möglichen Zustandsfolgen gibt.

→ Überwachung von technischen Systemen:

→ Man kann nur überwachen, ob die beobachtete Zustandsfolge zu dieser Menge möglicher Zustandsfolgen gehört, oder ob ein unvorhergesehener Fehler aufgetreten ist.

- Das Automatenmodell der Stanze wird ergänzt um die Modellierung von Fehlerfällen:
  - Fehlendes Rohmaterial
  - Fehlerhafte Ablage des fertig bearbeiteten Werkstücks

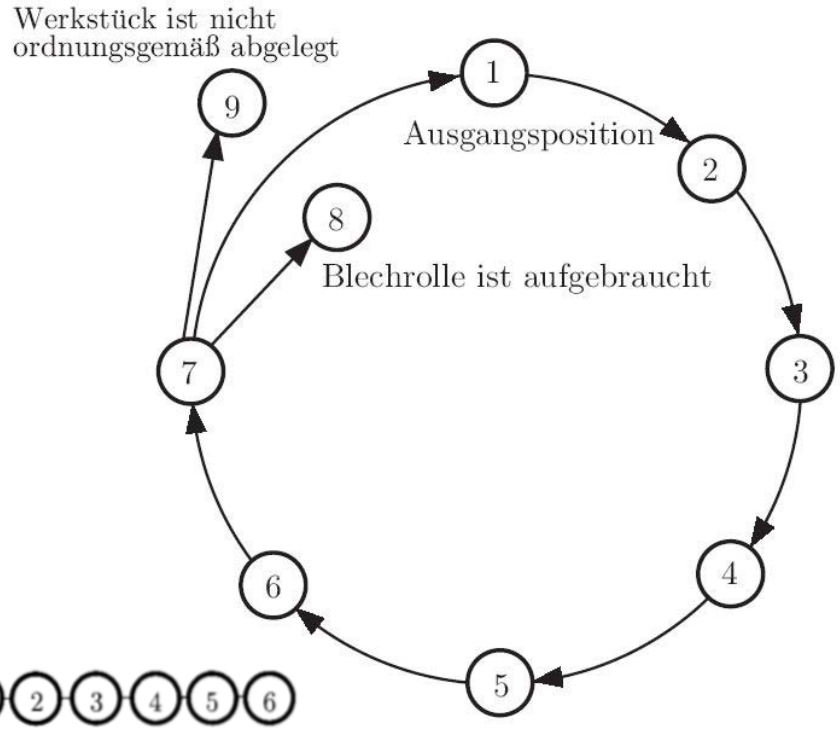
→ Änderung der Festlegung der Zustände:

Zustand	Beschreibung
1	Die Stanze steht in der Ausgangsposition und es ist noch Blech vorhanden.
8	Die Stanze steht in der Ausgangsposition, aber die Blechrolle ist aufgebraucht.
9	Das Werkstück ist nicht ordnungsgemäß abgelegt.

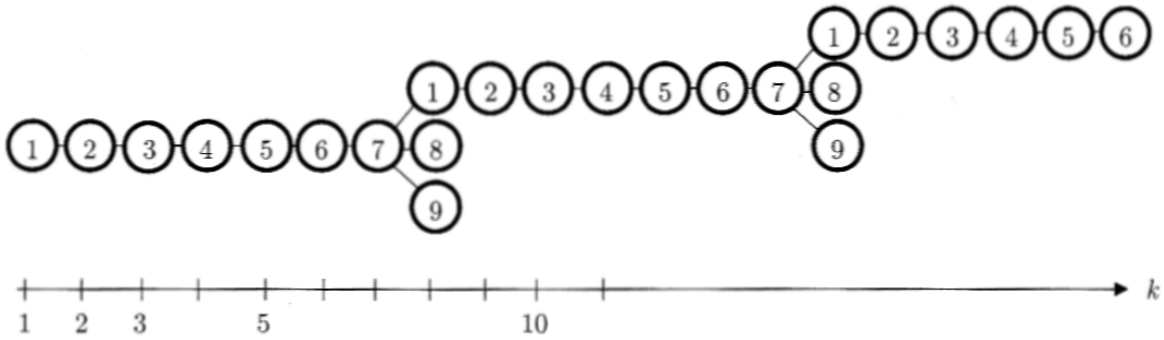
→ Aus Zustandsüberführungs-Funktion wird -Relation:

z'	z	Beschreibung
1	7	Stanze kehrt in die Ausgangsposition zurück
8	7	Ordnungsgemäßer Transport des Werkstücks, aber Blechrolle ist leer
9	7	Fehlerhafter Transport des Werkstückes aus der Stanze in das Lager

- Entsprechend erweiterter Automatengraph:



- Mögliche Zustandsfolgen (in graphischer Darstellung):



- Werden Automaten verwendet zur Beschreibung nichtdeterministischer Systeme, so muss die Modellform modifiziert werden:
  - ➢ Darstellung als Matrix:
    - Die Zustandsübergangs-Matrix *F* enthält in mindestens einer Spalte mehr als eine „1“.

    → Damit kann p(k+1) mehr als eine Komponente $p_i(k+1) = 1$ enthalten.

    → Da *F* aber auch in einer Zeile mehr als eine „1“ enthalten kann, kann die Matrixmultiplikation auf Ergebnisse > 1 führen.
    - Die Matrixoperation $p(k+1) = F \circ p(k)$ wird daher um eine „Abrundung auf 1“ ergänzt,
      so dass in p(k+1) keine Komponente $p_i(k+1) > 1$ auftritt.
    - Die Folge p(0), p(1), ..., p(k), p(k+1) ist eindeutig.

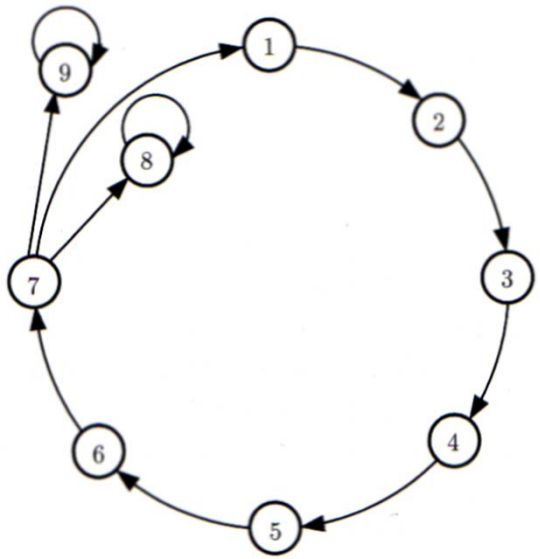
    → Damit kann aus p(k) eindeutig ermittelt werden, welche Zustände das System im Schritt k+1 einnehmen kann.

- Automat der Stanze in Matrixdarstellung

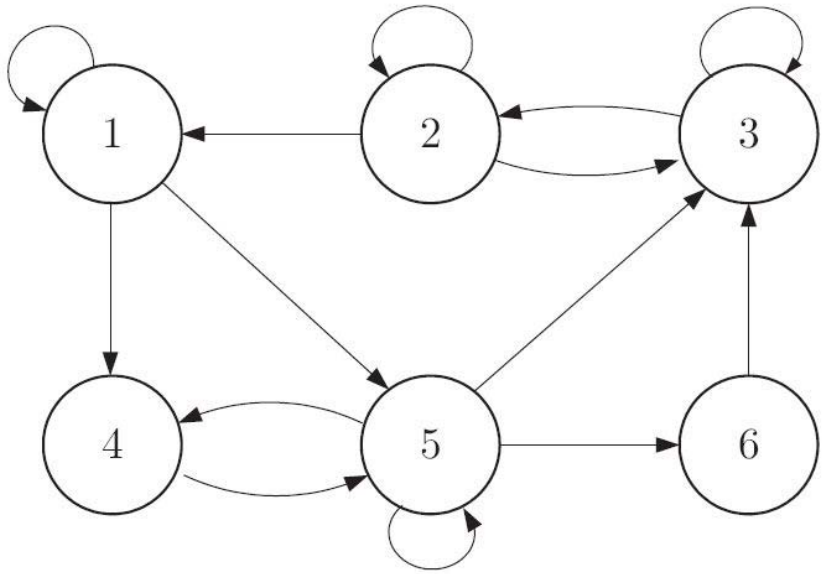
$$\begin{pmatrix} p_1(k+1) \\ p_2(k+1) \\ p_3(k+1) \\ p_4(k+1) \\ p_5(k+1) \\ p_6(k+1) \\ p_7(k+1) \\ p_8(k+1) \\ p_9(k+1) \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \end{pmatrix} \circ \begin{pmatrix} p_1(k) \\ p_2(k) \\ p_3(k) \\ p_4(k) \\ p_5(k) \\ p_6(k) \\ p_7(k) \\ p_8(k) \\ p_9(k) \end{pmatrix}, \quad \begin{pmatrix} p_1(0) \\ p_2(0) \\ p_3(0) \\ p_4(0) \\ p_5(0) \\ p_6(0) \\ p_7(0) \\ p_8(0) \\ p_9(0) \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

- Im Zustand $z(k) = Z_8$ oder $z(k) = Z_9$ ist $z(k+1)$ nicht definiert und damit die Gleichung ungültig.

→ Daher werden Selbstschleifen eingefügt.



- Nichtdeterministischer Automat:



- Wie lautet *F* für diesen Automaten?
- Wie lautet die Folge der ersten 5 Zustandsvektoren, wenn $z(0) = z_1$?
- Ab welchem Schritt kann sich das System in jedem beliebigen Zustand aufhalten?

- Nichtdeterministische Automaten erlauben von einem Zustand ausgehend verschiedene Zustandsübergänge.
  - Sie erlauben keine Aussage darüber, welcher Zustandsübergang als nächstes voraussichtlich vollzogen wird.
- In vielen technischen Systemen treten manche Zustandsübergänge häufiger auf als andere.
  - Zum Beispiel treten Übergänge in Fehlerzustände i.a. seltener auf als Übergänge in erwünschte Zustände.
  - Dies kann mit Hilfe von Ereignis-Auftritts-Wahrscheinlichkeiten
    = Zustands-Übergangs-Wahrscheinlichkeiten modelliert werden.
  - Dies führt auf die <u>stochastischen Automaten</u>.

- Bei einem stochastischen Prozess kann aus der Kenntnis des Anfangszustands z_0 nicht eindeutig der nachfolgende Zustand $z_p(1)$ vorhergesagt werden, weil der Zustandsübergang vom Zufall abhängt.
- Die Wahrscheinlichkeit, dass auf den Zustand z der Zustand z‘ folgt, wird beschrieben durch

$$p_{z',z} = \mathrm{P}(z(k+1) = z' \mid z(k) = z)$$

- Die einzelnen Zustandsübergangs-Wahrscheinlichkeiten summieren sich zu eins:

$$\sum_{z'=1}^{N} p_{z'z} = 1$$

- Da sich das System immer in irgendeinem Zustand befindet, gilt:

$$\sum_{z=1}^{N} p_z(k) = 1 \quad \text{für alle } k$$

mit $p_z(k)$ als Wahrscheinlichkeit, dass das System sich im Schritt k im Zustand z befindet.

- Der stochastische Automat ist beschrieben durch
  - seine Zustandsmenge N_z
  - seine Zustandsübergangswahrscheinlichkeitsverteilung F
  - und die Wahrscheinlichkeitsverteilung des Anfangs-zustandes p_0

$$F : N_z \times N_z \rightarrow [0,1] \qquad\qquad p_0 : N_z \rightarrow [0,1]$$

- F setzt sich aus den einzelnen Zustandsübergangswahrschein-lichkeiten zusammen:

$$F(z' \mid z) = p_{z',z} = \mathrm{P}(z(k+1) = z' \mid z(k) = z)$$

- Der Automat kann jede Zustandsfolge durchlaufen,
  - die in einem Zustand *z(0)* beginnt, den der Automat mit einer nicht verschwindenden Wahrscheinlichkeit annimmt,
  - und bei der aufeinander folgende Zustände *z(k), z(k+1)* eine nicht verschwindende Übergangswahrscheinlichkeit haben:

$$p_0(z(0)) > 0$$

$$p_{z',z} = \mathrm{P}(z(k+1) = z' \mid z(k) = z) > 0$$

- Mit welcher Wahrscheinlichkeit befindet sich der Automat im Schritt k+1 im Zustand z‘? $\mathrm{P}(z(k+1)=z')$
- Dies kann iterativ bestimmt werden:
  - Der Automat kann (mit einer gewissen Wahrscheinlichkeit) im Schritt k im Zustand z gewesen sein: $\mathrm{P}(\mathrm{z}(\mathrm{k})=\mathrm{z})$
  - Aus z kann er (mit einer gewissen Wahrscheinlichkeit) nach z‘ übergehen: $\mathrm{P}(\mathrm{z}(\mathrm{k}+1)=\mathrm{z}' \mid \mathrm{z}(\mathrm{k})=\mathrm{z})$
  - Die Wahrscheinlichkeit, dass der Automat im Schritt k in Zustand z gewesen ist und dann nach z‘ übergeht, ermittelt sich aus dem Produkt der Wahrscheinlichkeiten:

$$\mathrm{P}(\mathrm{z}(\mathrm{k}+1)=\mathrm{z}' \mid \mathrm{z}(\mathrm{k})=\mathrm{z}) \cdot \mathrm{P}(\mathrm{z}(\mathrm{k})=\mathrm{z})$$

  - Da es mehrere Zustände z geben kann, von denen aus der Automat nach z‘ übergegangen sein könnte, wird aufsummiert:

$$\mathrm{P}(z(k+1)=z')=\sum_{z=1}^{n} \mathrm{P}(\mathrm{z}(\mathrm{k}+1)=\mathrm{z}' \mid \mathrm{z}(\mathrm{k})=\mathrm{z}) \cdot \mathrm{P}(\mathrm{z}(\mathrm{k})=\mathrm{z})$$

  - In Kurzform: $p_{z'}(k+1)=\sum_{z=1}^{N} p_{z'z} \cdot p_z(\mathrm{k})$

- Matrixdarstellung
  - Der Vektor p(k) enthält die Zustandswahrscheinlichkeiten $p_z(k)$.
    - Für Komponenten $p_i(k)$ gilt:

$$p_i(k) = P(z(k) = i)$$

- Die Matrix F enthält die Zustandsübergangs-wahrscheinlichkeiten $p_{z'z}$.
  - Für Komponenten f_{ij} gilt: $$f_{ij} = P(z(k+1) = i \mid z(k) = j)$$
- Damit lässt sich p(k+1) iterativ berechnen:

$$p(k+1) = F \cdot p(k), \qquad k = (0,1,2,...)$$

- Ist der Vektor der Anfangs-Zustandswahrscheinlichkeiten $p_0 = p(0)$ bekannt, lässt sich p(k+1) auch direkt berechnen:

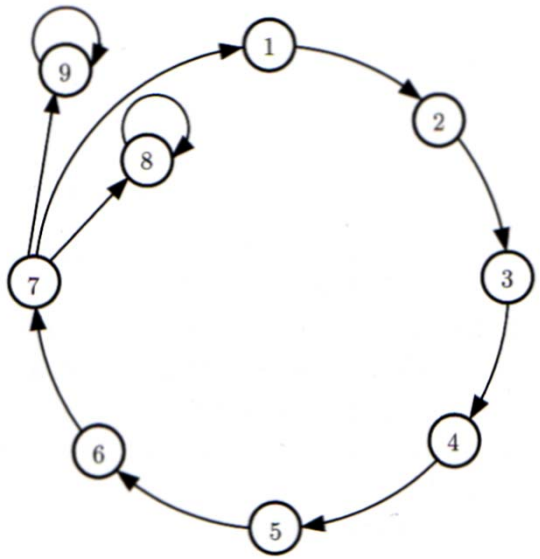
$$p(k+1) = F^{k+1} p_0$$

- Die nichtdeterministische Beschreibung der Zustandsübergänge

z'	z	Beschreibung
1	7	Stanze kehrt in die Ausgangsposition zurück
8	7	Ordnungsgemäßer Transport des Werkstücks, aber Blechrolle ist leer
9	7	Fehlerhafter Transport des Werkstückes aus der Stanze in das Lager

wird ergänzt um Zustandsübergangswahrscheinlichkeiten:

- p_{87}=0,001
  - d.h. eine Blechrolle reicht für durchschnittlich 1000 Teile;
- p_{97}=0,0005
  - d.h. bei durchschnittlich 1 von 2000 Zyklen wird ein Werkstück fehlerhaft transportiert.
- Daraus ergibt sich
  $p_{17} = 1 - p_{87} - p_{97} = 0{,}9985$



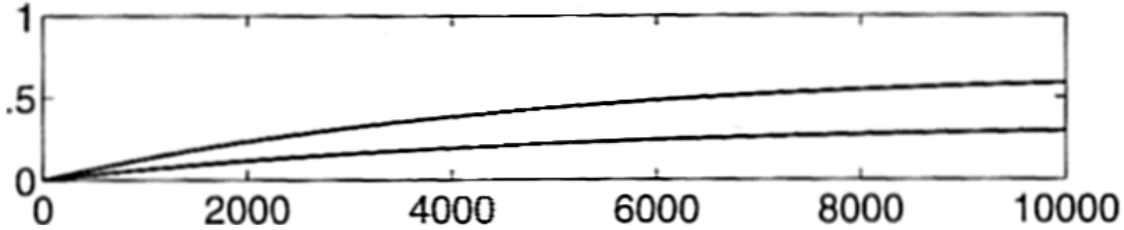
- Daraus ergibt sich die Zustandsübergangsmatrix *F*

$$\begin{pmatrix} p_1(k+1) \\ p_2(k+1) \\ p_3(k+1) \\ p_4(k+1) \\ p_5(k+1) \\ p_6(k+1) \\ p_7(k+1) \\ p_8(k+1) \\ p_9(k+1) \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0{,}9985 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0{,}0010 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0{,}0005 & 0 & 1 \end{pmatrix} \circ \begin{pmatrix} p_1(k) \\ p_2(k) \\ p_3(k) \\ p_4(k) \\ p_5(k) \\ p_6(k) \\ p_7(k) \\ p_8(k) \\ p_9(k) \end{pmatrix}, \quad \begin{pmatrix} p_1(0) \\ p_2(0) \\ p_3(0) \\ p_4(0) \\ p_5(0) \\ p_6(0) \\ p_7(0) \\ p_8(0) \\ p_9(0) \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

- In welchen Zuständen kann sich die Stanze im Schritt k befinden, und wie wahrscheinlich sind diese Zustände?

$$p(k) = F^k p_0$$

- Zum Beispiel k = 1500:
- $p(1500)=F^{1500}p_0=(0\ \ 0\ \ 0{,}7252\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0{,}1832\ \ 0{,}0916)^T.$
- Das heißt, die Wahrscheinlichkeit, dass die Stanze nach 1500 Schritten in einem fehlerhaften Zustand steht, liegt bei 27%.
- Die folgende Darstellung zeigt die Fehlerwahrscheinlichkeiten p_8 und p_9 aufgetragen über den Schritten k:



Zusammenfassung

- Nicht-deterministische Automaten
  - Gründe für Nichtdeterminismus
  - Deterministische vs. nicht-deterministische Automaten
  - Darstellung nicht-deterministischer Automat
  - Beispiele
- Stochastische Automaten
  - Gründe für stochastische Modellierung
  - Übergang vom nicht-deterministischen zum stochastischen Automaten
  - Darstellung stochastischer Automat
  - Beispiele

Aufgabe für die Nachbereitung

- Wie lautet die Matrixdarstellung für den Automaten auf der Folie 61 (Behälter) ?
- Beschreiben Sie diesen Automaten mit Hilfe seiner Zustandsübergangs<u>relation</u>.

Aufgabe für die Nachbereitung

- Wie sieht der Automatengraph des Roboters (Folie 55) aus, wenn nicht festgelegt ist, welches Werkstück als nächstes gegriffen werden soll? Woran kann man im Automatengraphen diesen Nichtdeterminismus erkennen?

- Modellieren Sie folgendes System als stochastischen Automaten:
  - Ein Fertigungszyklus bestehe aus 5 aufeinanderfolgenden Zuständen, die zyklisch durchlaufen werden.
  - Aus jedem dieser Zustände wird beim Auftreten eines Fehlers in den allgemeinen Fehlerzustand gewechselt.
  - Aus dem allgemeinen Fehlerzustand kann in keinen anderen Zustand als den Anfangszustand übergegangen werden.
  - Die Wahrscheinlichkeit, dass von einem Zustand im Fertigungszyklus korrekt zum nächsten gewechselt wird, betrage 99%.
  - Die Wahrscheinlichkeit des Übergangs vom Fehlerzustand in den Anfangszustand nennt man („Reparaturquote“).
- Wie groß ist die Wahrscheinlichkeit, dass sich das System nach 4 Schritten im Fehlerzustand befindet?
  - A) mit Reparaturrate 0
  - B) mit Reparaturrate 90%

Systemmodellierung
Teil 1: Ereignisdiskrete Systeme

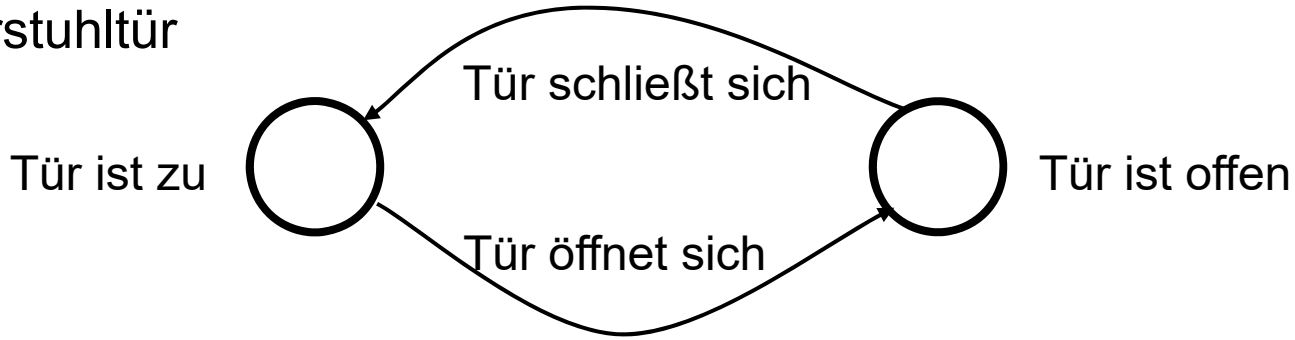
Kapitel 3
Ereignisdiskrete Systeme (III)

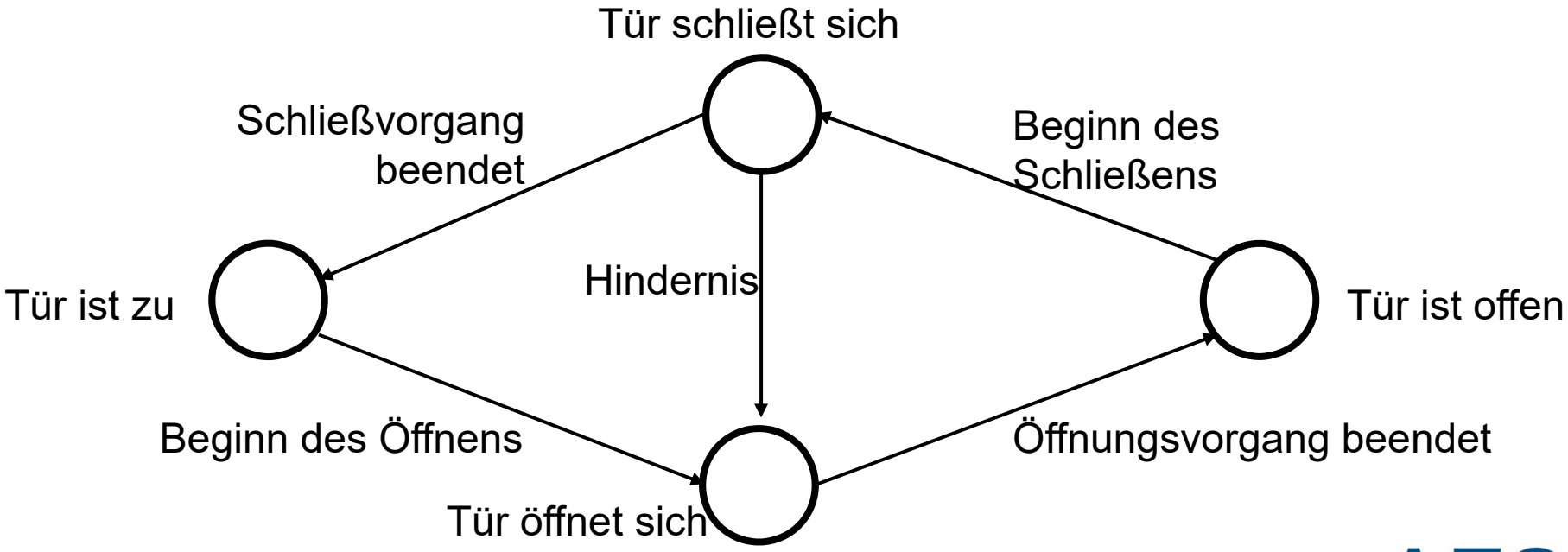
Modellierung mit E/A-Automaten

UNIVERSITÄT DES SAARLANDES

AES
Automatisierungs- und Energiesysteme

- Verfeinerte Modellbildung
  - Beispiel: Fahrstuhltür



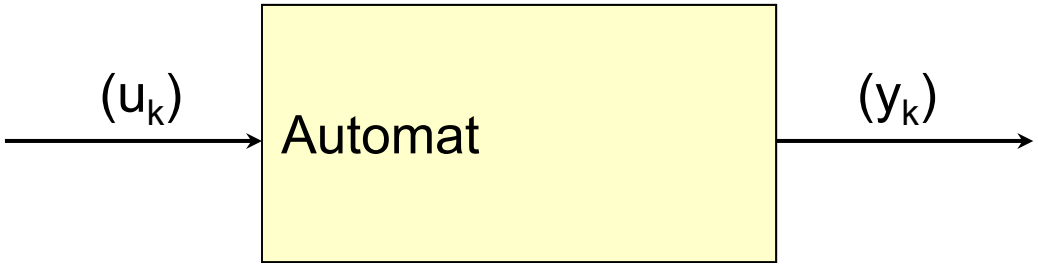


Automaten mit Eingaben und Ausgaben

- Bislang waren wir von Automaten ausgegangen, bei denen die Zustandsübergänge durch Ereignisse ausgelöst wurden, ohne dass wir diese Ereignisse näher spezifiziert hätten.
- Ereignisse können sein:
  - interne, von außen nicht sichtbare Ereignisse
  - ein (Zeit-)Taktsignal
    - intern (nicht sichtbar) oder extern (von außen vorgegeben)
  - <u>Eingaben, die von Eingangssignalen herrühren</u>

- Im Folgenden werden ereignisdiskrete Systeme als Automaten mit Eingaben und Ausgaben („E/A-Automaten“) modelliert.
- Die bisherigen „autonomen Automaten“ werden dazu ergänzt.

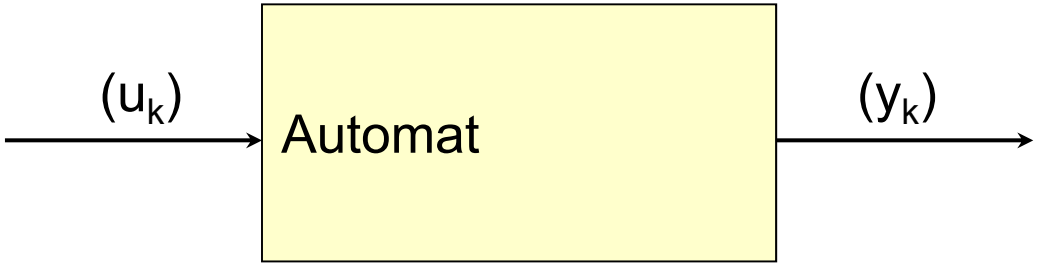
- Ein Automat ist ein dynamisches System, das eine Eingabefolge in eine Ausgabefolge abbildet



$$(u_k) = (u(0), u(1), u(2), \ldots) \qquad (y_k) = (y(0), y(1), y(2), \ldots)$$

- Bei einem technischen System
  - sind die Eingaben U oft Stellbefehle, die auf Aktoren im System wirken,
  - sind die Ausgaben Y oft Sensorsignale oder Kombinationen von Sensorsignalen.
    - Die Ausgaben sind eine teilweise oder vollständige Information darüber, in welchem Zustand sich das System befindet.

- Für die Darstellung eines Automaten mit Eingaben und Ausgaben gibt es drei geeignete Formen:

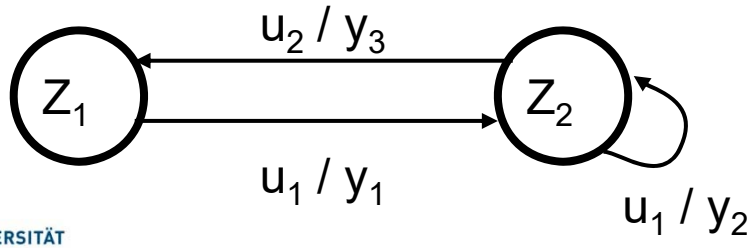


- Als Funktion

$$z(k+1) = f(z(k), u(k)) \qquad z(0) = z_0$$
$$y(k) = g(z(k), u(k))$$

- Als Automatengraph

Beispiel:



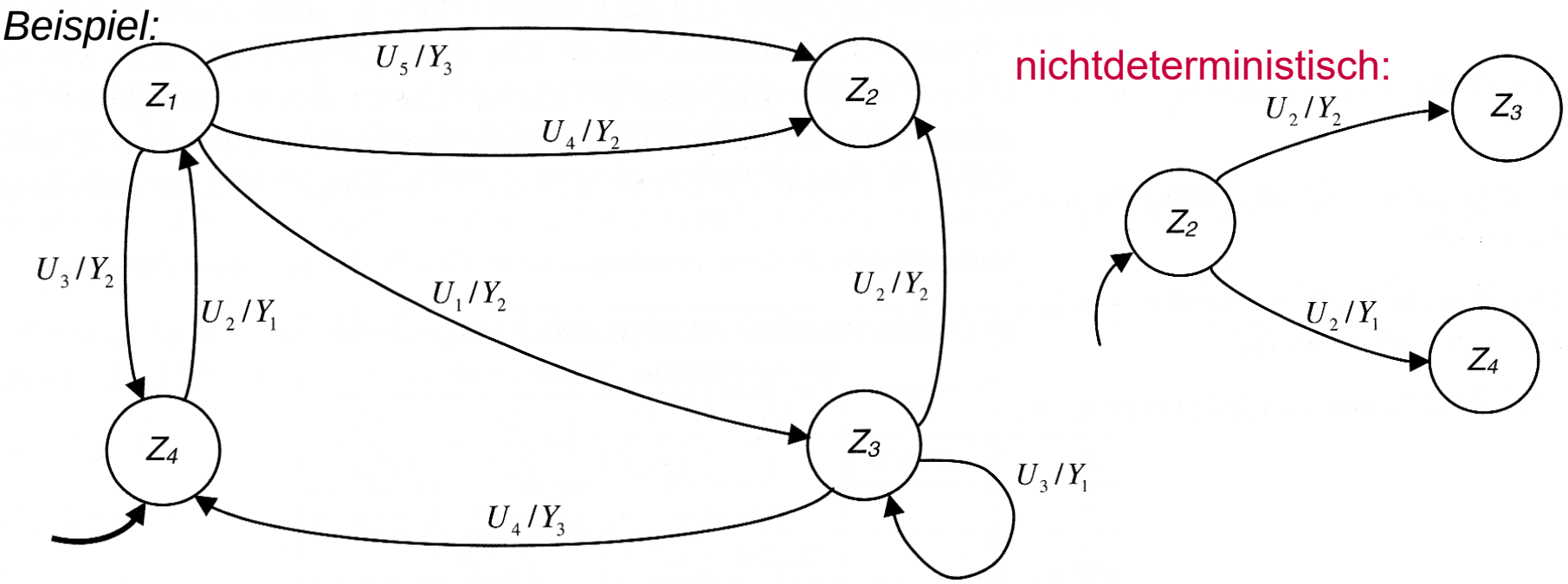
- Als Tabelle

Beispiel:

	z(k+1), y(k) bei u(k)	
z(k)	u_1	u_2
z_1	z_2, y_1	-
z_2	z_2, y_2	z_1, y_3

- Möglich: mehrere Kanten zwischen zwei Zuständen
- Möglich: Kante beginnt und endet im selben Zustand
- Kanten mit gleicher Eingabe führen von einem Knoten weg: Abbildung nichtdeterministischen Verhaltens

Beispiel:



- Die Automatentabelle stellt systematisch
  - für jeden Zustand
  - und jede Eingabe
  - die Ausgabe
  - und den Folgezustand dar.
- Hier gezeigt für das gleiche Beispiel

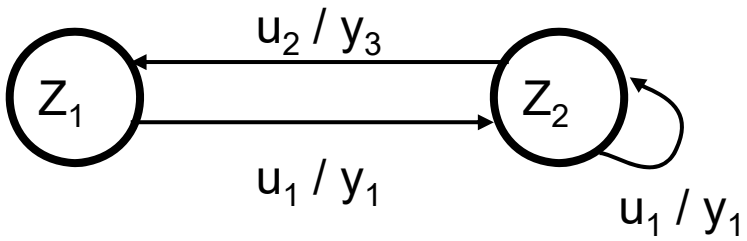
Beispiel:

Momentan-Zustand $z(k)$	(Folgezustand $z(k+1)$, Momentaner Ausgang $y(k)$) bei momentaner Eingabe $u(k)$				
	U_1	U_2	U_3	U_4	U_5
Z_1	(Z_3,Y_2)	–	(Z_4,Y_2)	(Z_2,Y_2)	(Z_2,Y_3)
Z_2	–	–	–	–	–
Z_3	–	(Z_2,Y_2)	(Z_3,Y_1)	(Z_4,Y_3)	–
Z_4	–	(Z_1,Y_1)	–	–	–

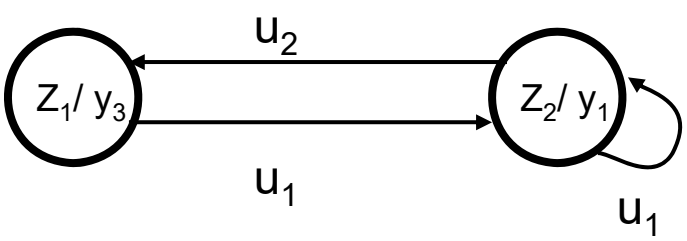
- Im Mealy-Automat sind die Ausgaben Zustandsübergängen zugeordnet
  - Die Ausgabe gibt an, dass ein Zustandsübergang stattfindet.

- Im Moore-Automat sind die Ausgaben Zuständen zugeordnet
  - Die Ausgabe gibt an, dass ein Zustandsübergang stattgefunden hat.

Beispiel:



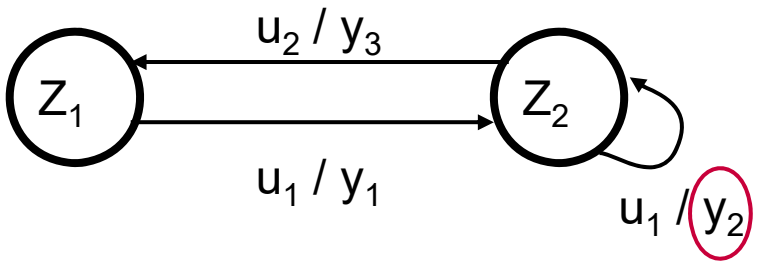
System als Mealy-Automat



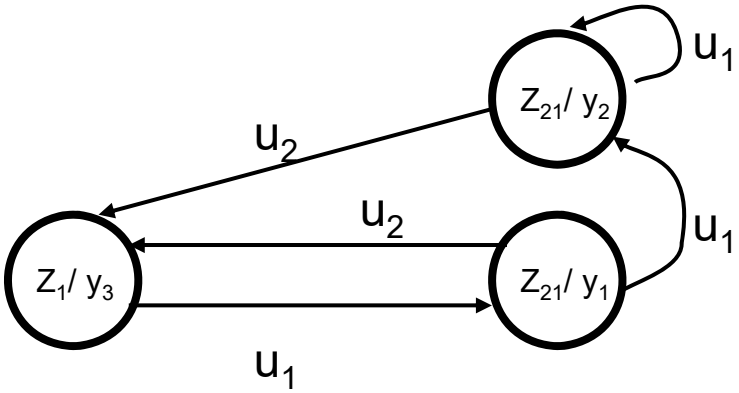
gleiches System als Moore-Automat

- Der Mealy-Automat auf der vorhergehenden Folie ließ sich direkt in den äquivalenten Moore-Automaten umwandeln. Dies ist nicht immer so möglich, wie folgendes Beispiel zeigt:

Beispiel:



System als Mealy-Automat



gleiches System als Moore-Automat

- Beim Moore-Automaten ist jedem Zustand eindeutig eine Ausgabe zugeordnet.

Beispiel Getränkeautomat (Tafel)

- Modellieren Sie den Getränkeautomaten im Foyer als Moore-Automat
  - Eingaben: Bedienhandlungen, …
  - Ausgaben: Getränkeausgaben, Anzeigen am Display
- Handelt es sich um einen deterministischen Automaten?
  - Überlegen Sie Beispiele für nichtdeterministisches Verhalten eines Getränkeautomaten.

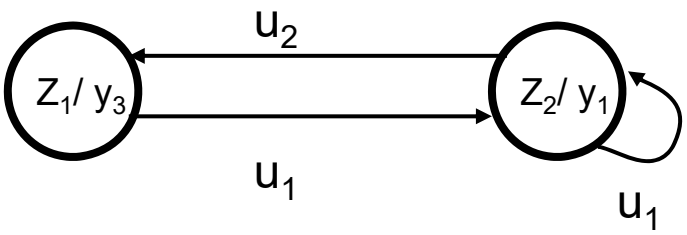
Aufgabe für die Nachbereitung

- Beschreiben Sie den Ablauf zum Starten eines PKW (vom Aufschließen bis zum Anfahren) als E/A-Automat.
  - Berücksichtigen Sie dabei, dass die Reihenfolge von manchen Tätigkeiten fest vorgegeben ist, andere hingegen in beliebiger Reihenfolge durchgeführt werden können.

Aufgabe für die Nachbereitung

- Modellieren Sie einen Personenaufzug (3 Etagen) als ereignisdiskretes System.
  - Bei der Modellierung sollen die Systemaspekte berücksichtigt werden, die für den Nutzer des Aufzugs sichtbar sind.
- Stellen Sie das Ergebnis als Automatengraph dar.
- Handelt es sich um ein deterministisches oder ein nichtdeterministisches System?
- Gibt es Systemaspekte, die man getrennt voneinander modellieren kann?
  Wie stellt sich das im Automatengraphen dar?

- Technische Systeme lassen sich oft gut mit Moore-Automaten modellieren:
  - Die Zustände des Automaten beschreiben Zustände des Systems, in denen sich die für die Betrachtung relevanten Zustandsgrößen nicht ändern.
  - Zustandsübergänge werden ausgelöst durch Eingaben U, welches oft Stellbefehle sind, die auf Aktuatoren im System wirken.
  - Das Erreichen eines neuen Zustands wird technisch oft durch Sensoren festgestellt und nach außen signalisiert durch die dem Zustand zugeordnete Ausgabe Y.
  - Die Zeit, die für den Zustandsübergang benötigt wird, sollte vergleichsweise kurz sein.
    - Sonst sollte ein Zwischenzustand eingeführt werden.

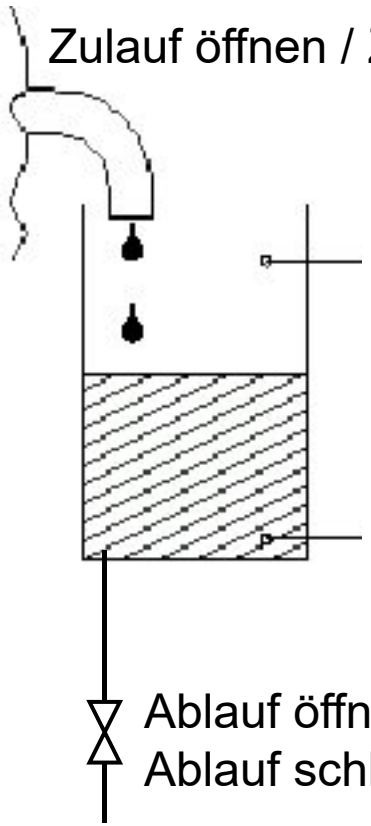


Beispiel: Behälter als Moore-Automat

- Zustände werden definiert durch Sensorsignale:
  - Z1: unten und oben Flüssigkeit
  - Z2: unten Flüssigkeit, oben nicht
  - Z3: unten und oben keine Flüssigk.
- mögliche Eingaben:

u	Ablauf	Zulauf
u_1	*zu*	*zu*
u_2	*zu*	*offen*
u_3	*offen*	*

Annahme:
Ablauf > Zulauf



- Ergebnis

	z(k+1) bei u(k)			
z(k)	u_1	u_2	u_3	u_4
z_1				
z_2				
z_3				

* = „don‘t care“

- Zustandsübergänge werden durch Ereignisse ausgelöst.

  a) Ein Zustandsübergang ist eindeutig an eine Eingabe gebunden.
  → Die Eingabe löst den Zustandsübergang aus.
  → Durch die Eingabe kann der Zustandsübergang erzwungen werden.
  → Durch Vermeiden dieser Eingabe kann der Zustandsübergang verhindert werden.

  → Dieses nennt man „steuerbares Ereignis“.

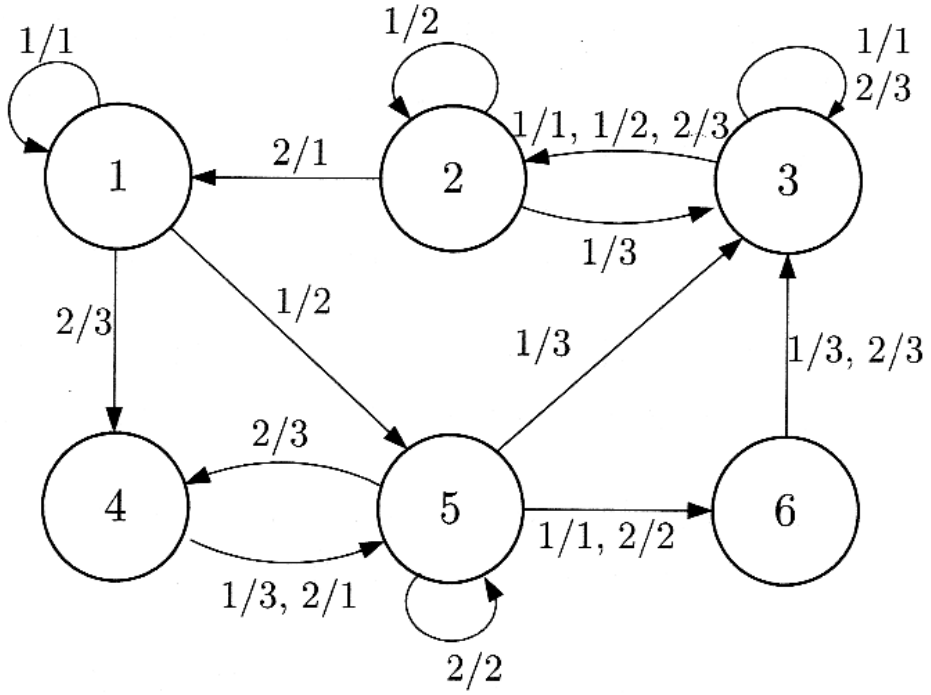
  b) Ein Zustandsübergang ist nicht (oder nicht eindeutig) an eine Eingabe gebunden.
  → Der Zustandsübergang erfolgt spontan.
  → Der Zustandsübergang kann nicht durch eine Eingabe erzwungen werden.
  → Der Zustandsübergang kann nicht durch eine Eingabe verhindert werden.

  → Dieses nennt man ein „nicht steuerbares Ereignis“.

  → Enthält ein Automat nicht steuerbare Ereignisse, so ist sein Verhalten nicht vorhersagbar = nichtdeterministisch

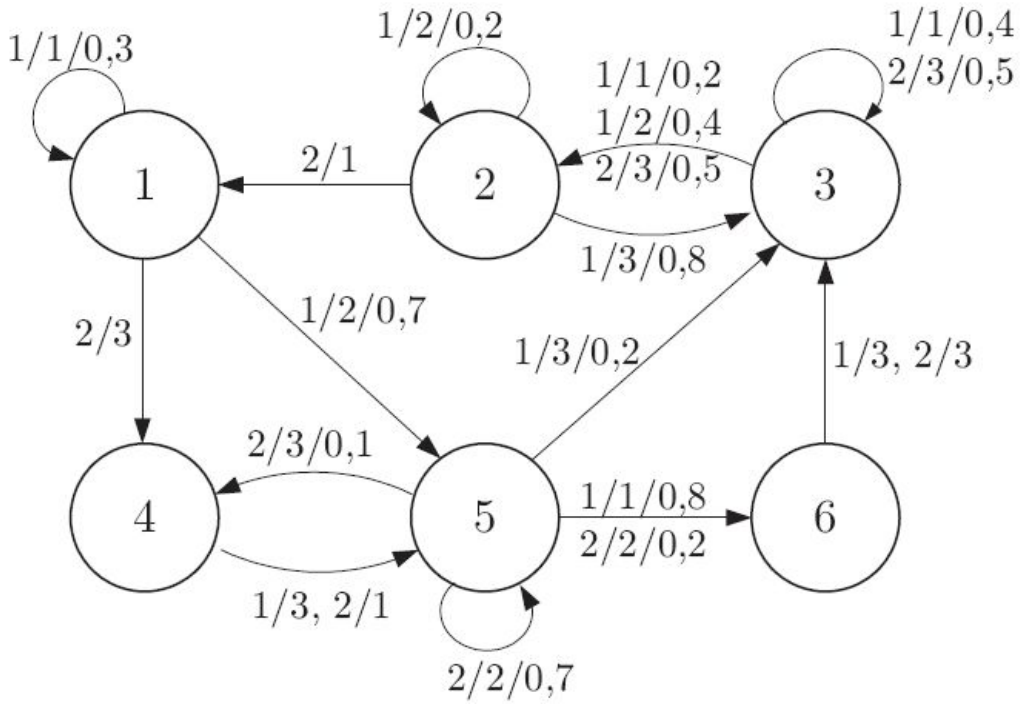
- Unterscheidung von Nichtdeterminismus im Moore-Automaten und Mealy-Automaten:
  - Im Moore-Automaten sind die Ausgaben Zuständen zugeordnet; nichtdeterministisch:
    - Bei gleicher Eingabe ist Übergang in verschiedene Zustände möglich
  - Im Mealy-Automaten sind die Ausgaben Zustands-übergängen zugeordnet, nichtdeterministisch:
    - Bei gleicher Eingabe ist Übergang in verschiedene Zustände möglich
    - Bei Übergang in gleichen Zustand sind verschiedene Ausgaben möglich

- Beispiel für nichtdeterministischen Mealy-Automaten:
  - Bei gleicher Eingabe ist Übergang in verschiedene Zustände möglich (im Beispiel von z_5 aus nach z_5 oder z_6).
  - Bei gleicher Eingabe verschiedene Ausgaben möglich (im Beispiel beim Übergang von z_3 nach z_2 bei Eingabe u_1 die Ausgaben y_1 oder y_2).
  - Und beides in Kombination (Beispiel: z_2 und u_1)



- Bei stochastischen E/A-Automaten werden die Zustandsübergangswahrscheinlichkeiten dafür angegeben, dass bei der Eingabe u ein Übergang von z nach z‘ stattfindet.

  - Schreibweise:
    an den Kanten steht
    u / y / Wahrscheinlichkeit;
    bei $p_{z'z} = 1$ wird die
    Wahrscheinlichkeit
    weggelassen.



- Vorhersage des Systemverhaltens:
  - Gegeben: Modell (als Automat)
    Anfangszustand z_0
    Eingabefolge (u(k))
  - Gesucht: Zustandsfolge (z(k)), Ausgabefolge (y(k))

- Interessante Fragestellung:
  - Kommt das System in einem Endzustand,
    den es auch bei weiteren Eingaben nicht verlässt?
    - Kann bedeuten: Fehler, System bleibt ungewollt stehen
    - Kann bedeuten: Ziel erreicht

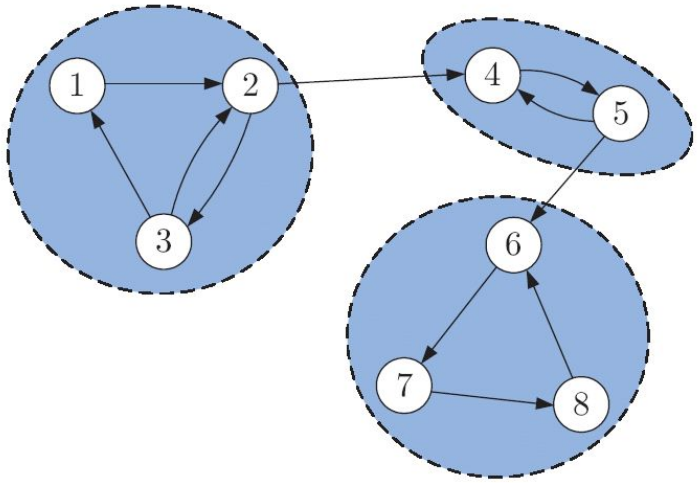
Endliche Automaten: Eigenschaften

- Der Automat befindet sich immer in genau einem Zustand
- Ein Automat bei dem in jedem Zustand die Reaktion auf alle möglichen Eingaben definiert ist heißt vollständig (oft wird implizit angenommen, dass eine nicht explizit angegebene Reaktion einer Schleife im Zustand entspricht)
- Anhand des Automatengraphen kann man streng zusammenhängende und zusammenhängende Automaten unterscheiden
  - zusammenhängend: jeder Zustand ist erreichbar
  - streng zusammenhängend: jeder Zustand ist von jeden anderen Zustand erreichbar

- Es gibt eine Vielzahl von Erweiterungen des Automatenbegriffs:
  - timed (mit Übergangszeiten)
  - stochastisch (es sind bei einer bestimmten Eingabe verschiedene Übergänge möglich)
  - hybrid (die Zustände des Automaten werden durch kontinuierliche Zustandsgleichungen beschrieben, Übergänge führen zu anderen Gleichungen und können von Werten der kontinuierlichen Zustandsgrößen anhängen)

- Überprüfung der Erreichbarkeit in Automaten:
  - Gegeben: Modell (als Automat)
    Anfangszustand z_0
    gewünschter Endzustand z_e
  - Gesucht: Zustandsfolge (z(k)) von z_0 nach z_e
    = im Automatengraph: Pfad von z_0 nach z_e

- Interessante Fragestellung:
  - Ist von einem gegebenen Anfangszustand z_0 aus ein bestimmter Endzustand z_e erreichbar?
  - Ist von jedem Zustand aus jeder andere Zustand erreichbar?
    - Wenn ja, ist Automatengraph „stark zusammenhängend“



- Bei allen hier behandelten Automaten,
  - ob autonom oder mit Eingaben und/oder Ausgaben,
  - ob deterministisch, nichtdeterministisch oder stochastisch,

- lässt sich der im nächsten Schritt eingenommene Zustand (bzw. die Menge der möglichen Zustände) beschreiben durch

$$\mathrm{p}(\mathrm{k}+1) = F \cdot \mathrm{p}(\mathrm{k}) \qquad \mathrm{k} = (0,1,2,\ldots)$$

- Das heißt, der aktuelle Zustand bietet dafür die ausreichende Informationsbasis.
- Insbesondere ist die Kenntnis in früheren Schritten eingenommener Zustände nicht erforderlich.
- Dies nennt man die <u>Markov-Eigenschaft</u>.

Zusammenfassung

- Ein endlicher Automat bildet eine Eingabefolge auf eine Ausgabefolge ab.
- Abhängig vom bisherigen Zustand und der aktuellen Eingabe wird ggf. ein neuer Zustand eingenommen.
- Nichtdeterminismus in ereignisdiskreten Systemen kann durch nichtdeterministische oder stochastische Automaten beschrieben werden.

Systemmodellierung
Teil 1: Ereignisdiskrete Systeme

Kapitel 4
Ereignisdiskrete Systeme (IV)

Petrinetze

UNIVERSITÄT DES SAARLANDES

AES
Automatisierungs- und Energiesysteme

Bewertung der Modellierung mit endlichen Automaten

- Engl.: finite automata (FA) auch finine state machines (FSM)
- Grundlegende Arbeiten: Mealy 1955 und Moore 1956
- Ziel: Beschreibung diskreter sequentieller Systeme
- Zugrunde liegende Mathematik: Theorie formaler Sprachen

- Endliche Automaten eignen sich zur Beschreibung dynamischer ereignisdiskrete Systeme (Zustand)
- Stark nebenläufige System führen bei endlichen Automaten zu Problemen (Zustandsexplosion).

Nebenläufigkeit fehlt → Petrinetze

- Engl.: Petri nets (PN)
- Grundlegende Arbeiten: Petri 1962
- Ziel: Beschreibung nebenläufiger Prozesse und deren Kommunikation
- Zugrunde liegende Mathematik: Graphentheorie
- Ein Petrinetz (PN) in seiner allgemeinen Definition ist ein autonomes System: *Es besteht keine Kopplung zur Umgebung*, d.h. im Gegensatz zum endlichen Automaten sind weder Eingaben noch Ausgaben definiert

Ein gewöhnliches Petrinetz ist ein Tripel PN = ($\boldsymbol{P}$, $\boldsymbol{T}$, $\boldsymbol{F}$) mit

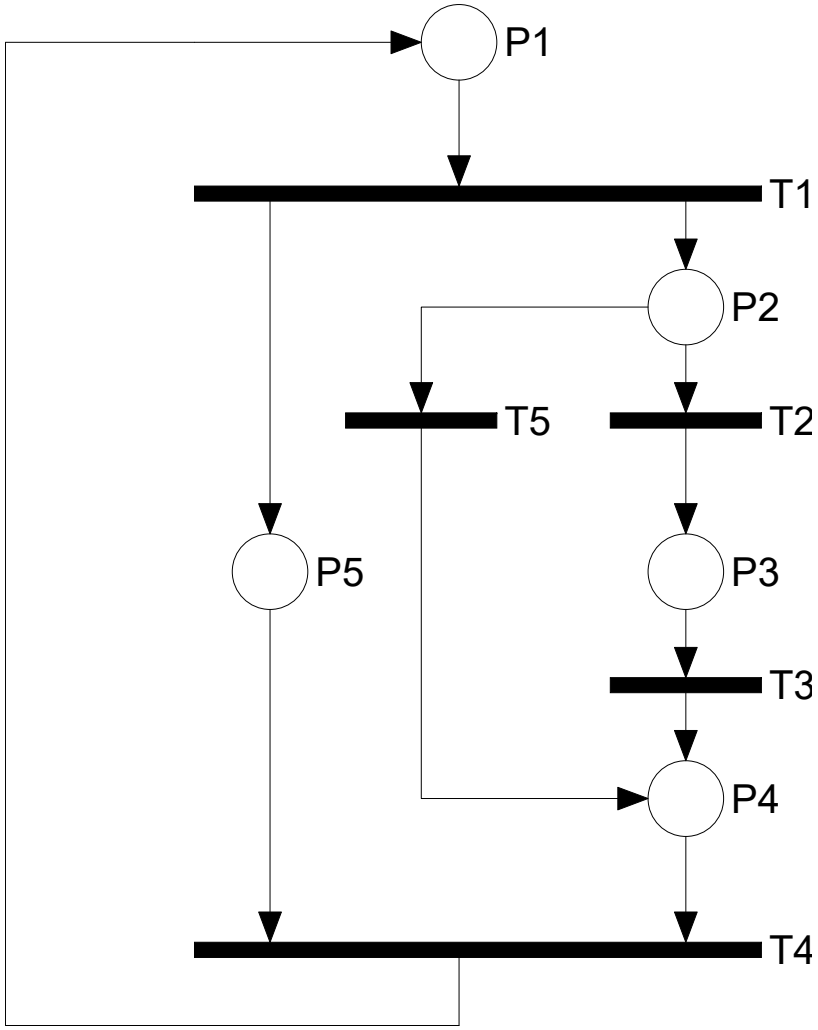
- $\boldsymbol{P} = \{p_1, \ldots, p_{|\boldsymbol{P}|}\}$ eine nichtleere, endliche Menge von Plätzen (Stellen),
- $\boldsymbol{T} = \{t_1, \ldots, t_{|\boldsymbol{T}|}\}$ eine nichtleere, endliche Menge von Transitionen mit $\boldsymbol{P} \cap \boldsymbol{T} = \emptyset$,
- $\boldsymbol{F} \subseteq \boldsymbol{P} \times \boldsymbol{T} \cup \boldsymbol{T} \times \boldsymbol{P}$ einer nichtleeren, endlichen Menge gerichteter Kanten. Die Kanten verbinden Plätze mit Transitionen oder Transitionen mit Plätzen jedoch nie zwei Plätze oder Transitionen miteinander (F steht für flow-relation bzw. Flussbeziehung)
  → bipartiter Digraph

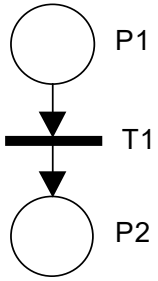
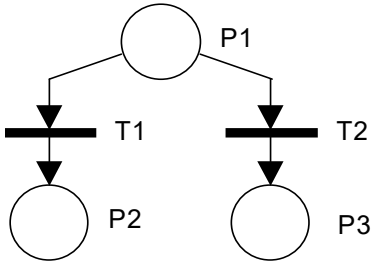
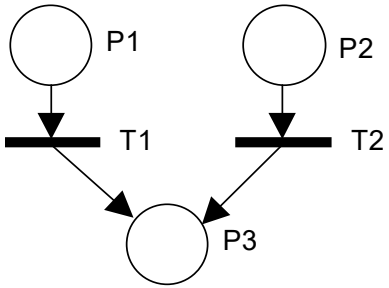
Grafische Darstellung:

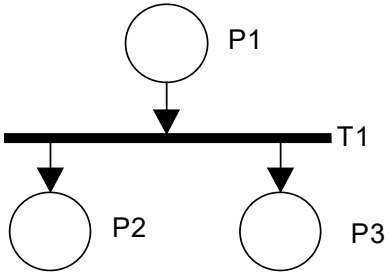
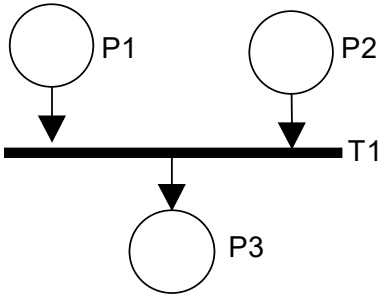
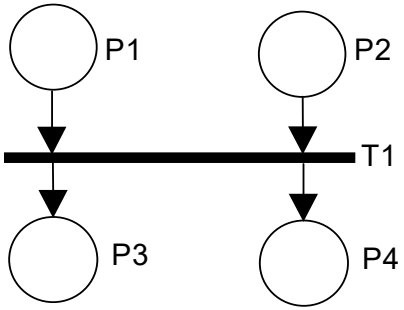
- Plätze → Kreise (manchmal auch Ovale)
- Transitionen → Balken, Rechtecke
- Kanten → Pfeile

Beispiel

- P = {P1, P2, P3, P4, P5}
- T = {T1, T2, T3, T4, T5}
- F = {(P1, T1), (T1, P2),
  (P2, T2), (T2, P3),
  (P3, T3), (T3, P4),
  (P4, T4), (T4, P1),
  (T1, P5), (P5, T4),
  (P2, T5), (T5, P4)}



	**Sequenz**
	**Auswahl-Verzweigung**
	**Zusammen-führung**

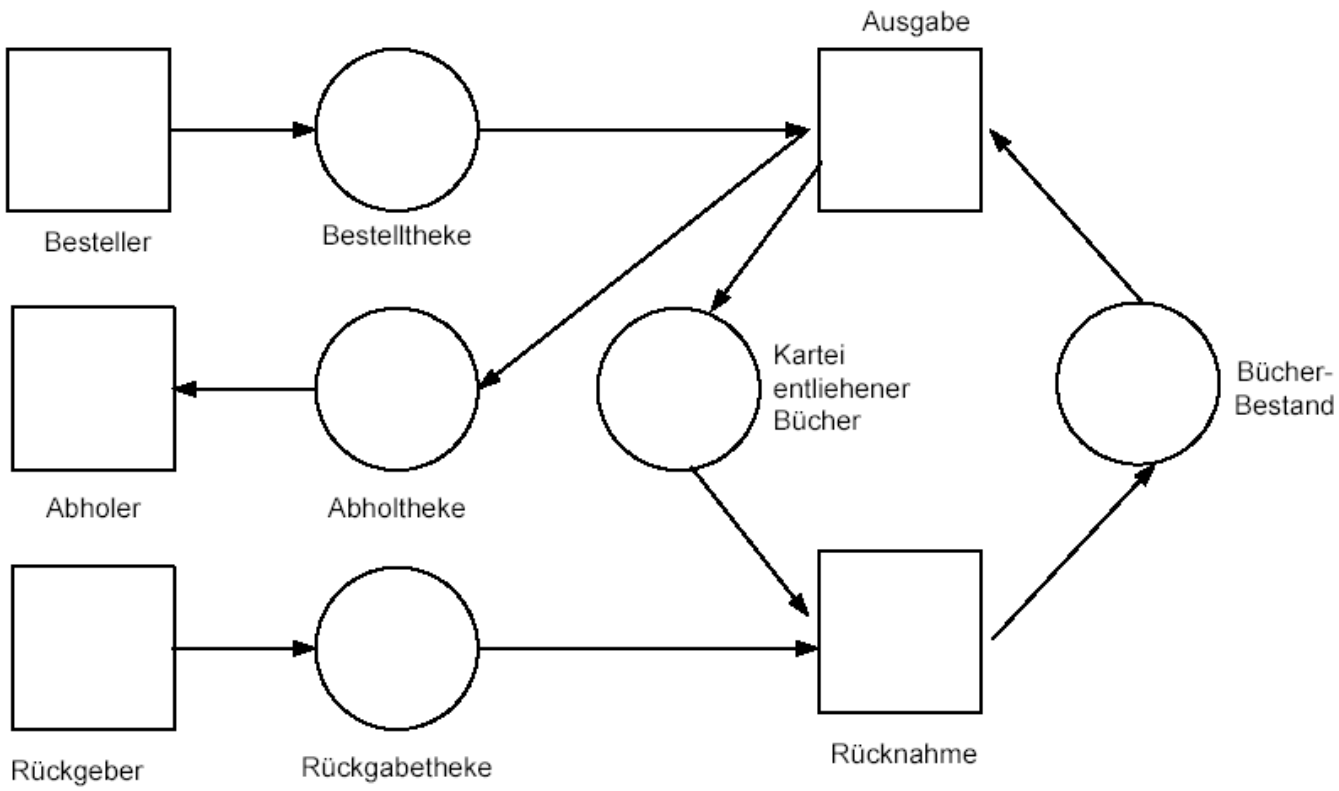
	**Aufspaltung (Vorwärts-synchronisation)**
	**Synchronisation (Rückwärts-synchronisation)**
	**Gleichzeitige Vorwärts- und Rückwärts-synchronisation**

Kanal-Agenten-Netz

- Das gewöhnliche Petrinetz besitzt keine Dynamik
- Gibt man den Netzelementen eine Bedeutung, so erhält man ein Kanal-Agenten-Netz
- auch Kanal-Instanzen-Netze (engl.: Channel Agency Nets)
  - Plätze = Kanäle = passive Systemkomponenten
    - Speicher
    - Annahme von Systemzuständen
    - Visualisierung von Objekten
  - Transitionen = Agenten = aktive Systemkomponenten
    - Herstellung, Verarbeitung, Transport von Objekten
    - Veränderung von Systemzuständen
  - Kanten = abstrakte, logische Beziehung zwischen Kanälen und Agenten, KEINE REALE SYSTEMKOMPONENTE
    - logischer Zusammenhang
    - räumliche Nähe
    - Zugriffsrechte
    - Verbindungen

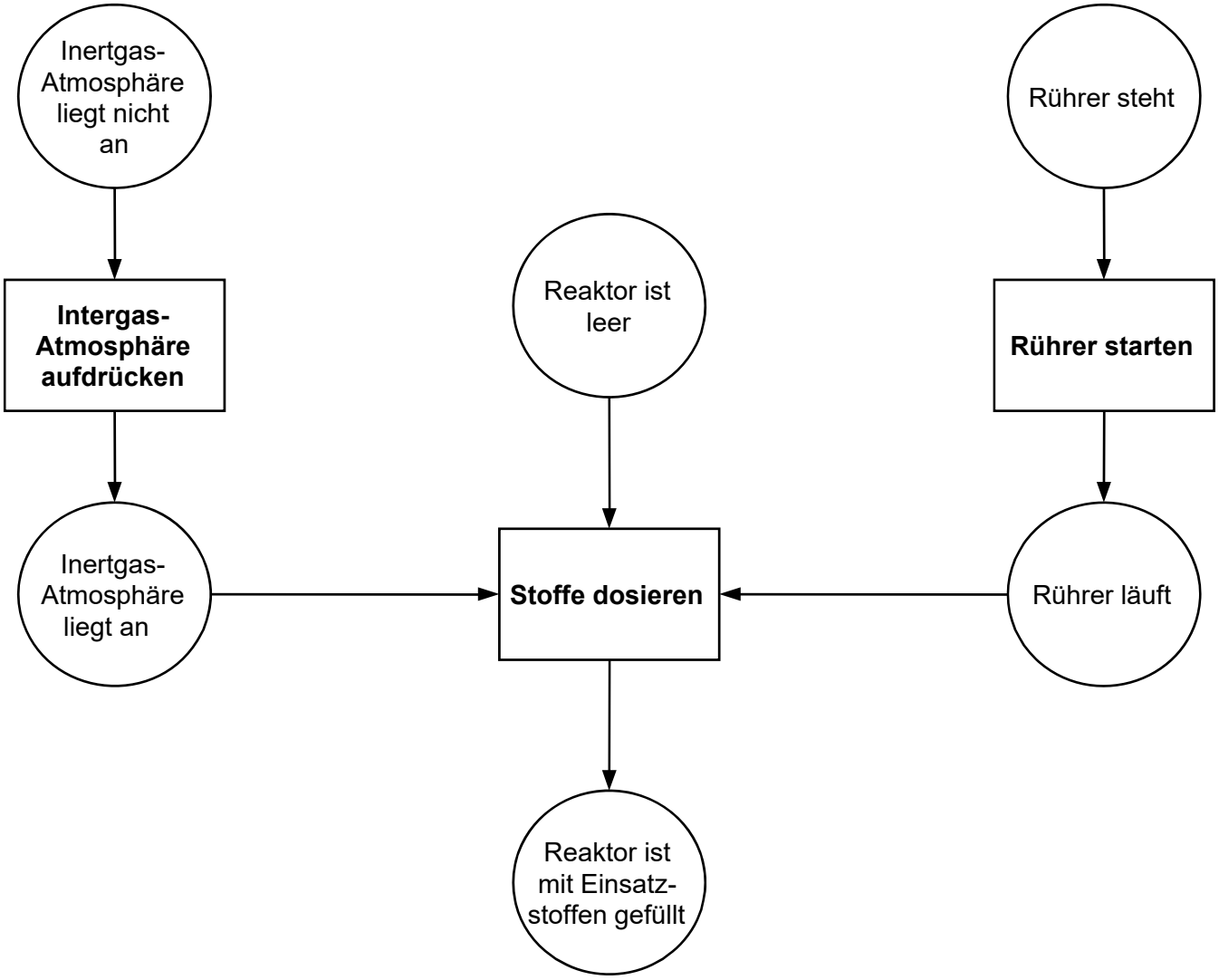
Beispiel: Kanal-Agenten-Netz Bibliothek

SM 1



Beispiel Kanal-Agenten-Netz chemischer Prozess

SM 1



- Petrinetze können auch zur Modellierung dynamischen Verhaltens verwendet werden.
- Dazu Erweiterung nötig:
  - Hinzufügen von Marken (tokens) auf den Plätzen
  - Definition einer Schaltregel zum Transport der Marken durch die Transitionen
- Abhängig vom gewählten Markierungskonzept entstehen verschiedene Arten von Petrinetzen
  - binäre Markierung = Entweder eine oder keine Marke auf einem Platz
    → Bedingungs-/Ereignisnetze BE-Netze bzw. Condition-Event Nets CE-Nets
  - Markierung mit positiven ganzen Zahlen = Keine oder mehrere nicht unterscheidbare (schwarze) Marken auf einem Platz
    → Platz-Transitions-Netze PT-Netze bzw. Place-Transition Nets (auch Stellen-Transitions-Netze, ST-Netze)
  - Markierung mit individuellen Marken = Mehrere unterscheidbare (farbige) Marken auf einem Platz
    → gefärbte Petrinetze bzw. coloured Petri nets (CP-Nets)
  - Markierung mit Prädikatenlogischen Ausdrücken
    → Prädikat-Transitionen-Netze PrT-Nets
  - Markierung mit reellen Zahlen → hybride Petrinetze

- **Zunächst nur BE-Netze und PT-Netze**
  ➔ Markierung lässt sich durch ganze Zahlen darstellen
- Eine *Markierung* eines Petrinetzes ist eine Abbildung
  $\mathbf{m}: \boldsymbol{P} \rightarrow \{0, 1, 2, \ldots\}^{|\boldsymbol{P}|}$, die jedem Platz eine nicht-negative, ganze Zahl von Marken zuweist.
- Die Markierung wird durch einen Vektor **m** beschrieben, dessen Dimension der Kardinalität der Menge der Plätze $\boldsymbol{P}$ entspricht.
- Ein Platz $p_i \in \boldsymbol{P}$ heißt *unmarkiert (frei, leer)* unter der Markierung **m** wenn das entsprechende Element des Vektors, bezeichnet mit $\mathrm{m}(p_i)$, Null ist.
- Ansonsten heißt der Platz *markiert* mit $\mathrm{m}(p_i)$ Marken.
- Für eine gegebene Markierung $\mathbf{m_a}$ ist die *Menge der markierten Plätze* $\boldsymbol{P}_\mathrm{m}$ gegeben durch $\boldsymbol{P}_\mathrm{m}(\mathbf{m_a}) = \{p \in \boldsymbol{P}: \mathrm{m_a}(p) > 0\}$.

- **Einschränkung für BE-Netze**
  ➔ Markierung lässt sich durch binäre Zahlen darstellen
- Eine *Markierung* eines Petrinetzes ist eine Abbildung
  $\mathbf{m}: \boldsymbol{P} \rightarrow \{0, 1\}^{|\boldsymbol{P}|}$, die jedem Platz eine oder keine Marke zuweist.
- Die Markierung wird durch einen Vektor **m** beschrieben, dessen Dimension der Kardinalität der Menge der Plätze $\boldsymbol{P}$ entspricht.
- Ein Platz $p_i \in \boldsymbol{P}$ heißt *unmarkiert (frei, leer)* unter der Markierung **m** wenn das entsprechende Element des Vektors, bezeichnet mit $m(p_i)$, Null ist.
- Ansonsten heißt der Platz *markiert*.
- Für eine gegebene Markierung $\mathbf{m_a}$ ist die *Menge der markierten Plätze* $\boldsymbol{P}_m$ gegeben durch $\boldsymbol{P}_m(\mathbf{m_a}) = \{p \in \boldsymbol{P}: m_a(p) = 1\}$.

Struktur

- Ein Bedingungs-Ereignis-Netz (BE-Netz) ist ein Quadrupel PN = ($\boldsymbol{P}$, $\boldsymbol{T}$, $\boldsymbol{F}$, $\mathbf{m}_0$) bestehend aus einem gewöhnlichen Petrinetz ($\boldsymbol{P}$, $\boldsymbol{T}$, $\boldsymbol{F}$) und einer binären Anfangsmarkierung $\mathbf{m}_0$: $\boldsymbol{P} \rightarrow \{0, 1\}^{|\boldsymbol{P}|}$.

Dynamik

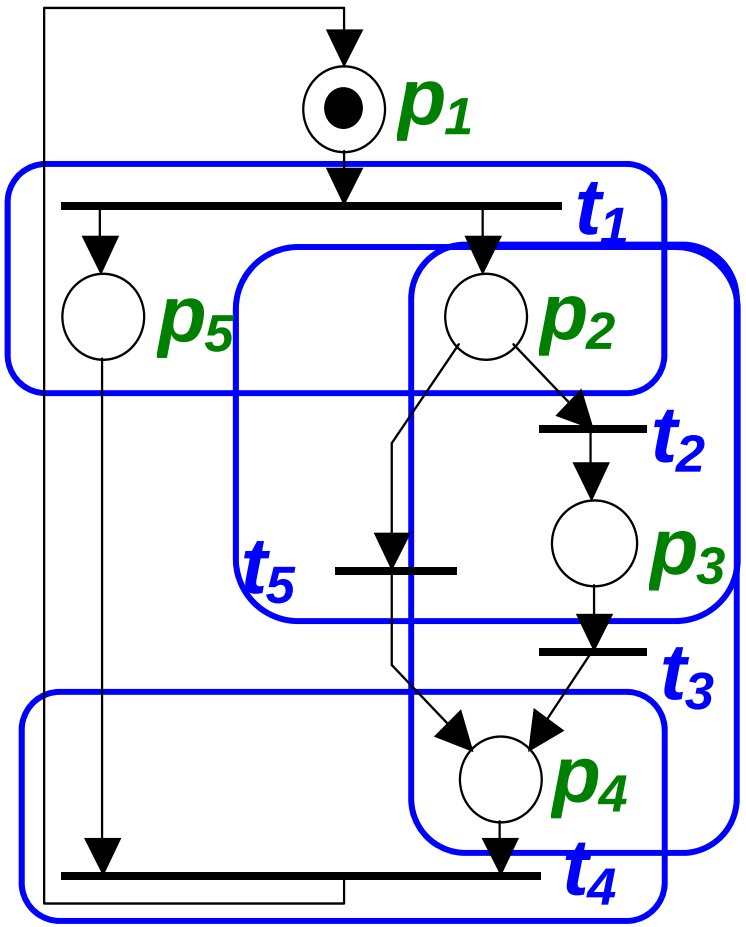
- Eine Transition t in einem BE-Netz ist konzessioniert (enabled) unter einer Markierung $\mathbf{m}$, En(t, $\mathbf{m}$) = True, wenn alle Plätze in ihrem Vorbereich markiert sind und alle Plätze in ihrem Nachbereich unmarkiert sind: $\mathrm{En}(t, \mathbf{m}) = \mathrm{True} \Leftrightarrow (\forall\, p \in \bullet t\colon \mathrm{m}(p) = 1) \wedge (\forall\, p \in t\bullet\colon \mathrm{m}(p) = 0)$
- Eine konzessionierte Transition **KANN** schalten (feuern).
- Mehrere gleichzeitig konzessionierte Transitionen schalten **EINZELN** nacheinander, niemals gleichzeitig.
- Beim Schalten wird allen Plätzen im Vorbereich die Marke entzogen, und alle Plätze im Nachbereich werden markiert.

Bezeichnung

- Die Plätze im Vorbereich einer Transition symbolisieren die *Bedingungen*, die erfüllt sein müssen, damit das *Ereignis* (symbolisiert durch die Transition) eintreten kann.

PN = (P, T, F, m_0)

- **Plätze** $P = \{p_1, \ldots, p_{|P|}\}$
- **Transitionen** $T = \{t_1, \ldots, t_{|T|}\}$
- **Kanten** $F \subseteq P \times T \cup T \times P$
- **Anfangsmarkierung** m_0: $P \rightarrow \{0, 1\}^{|P|}$

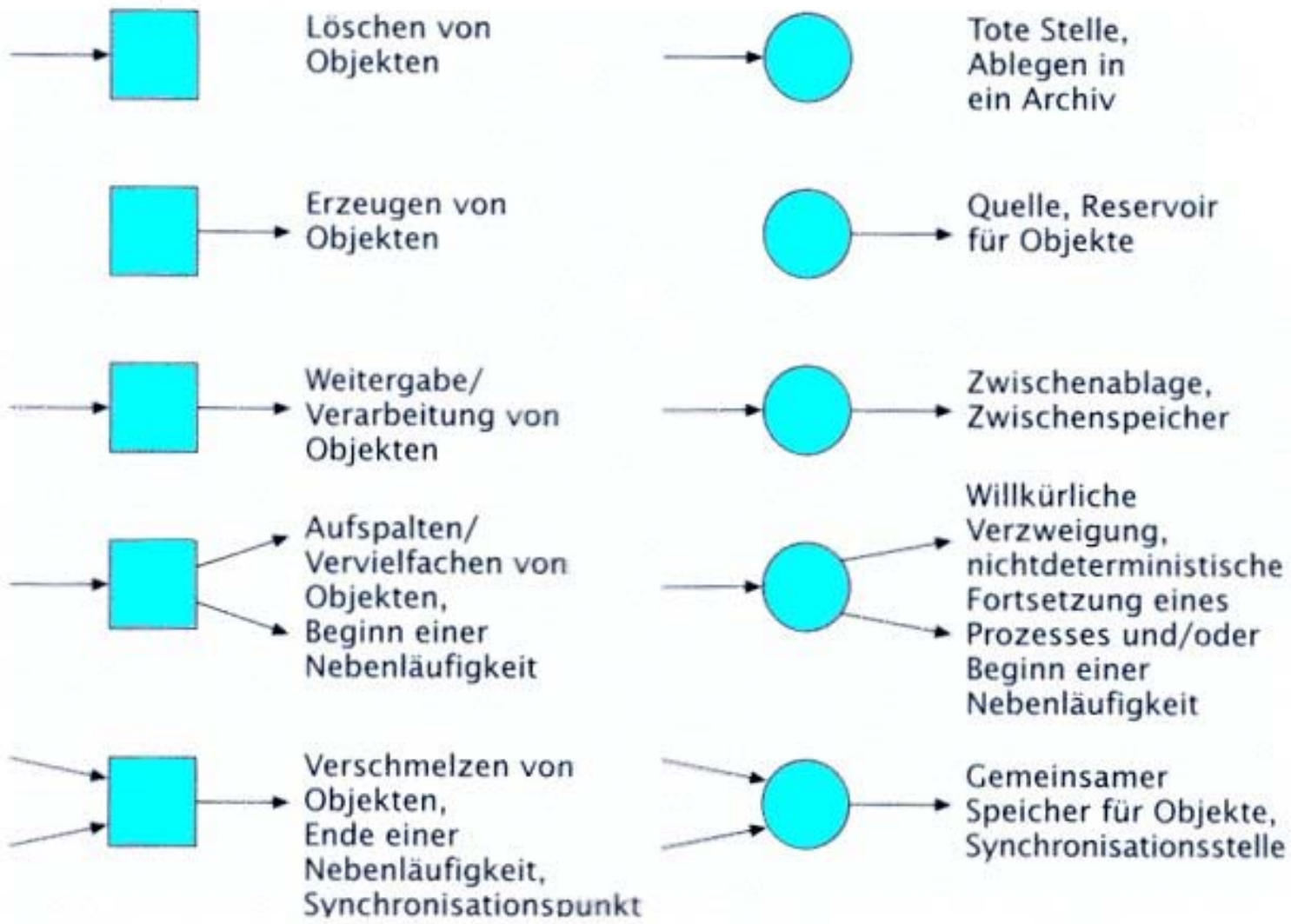


Modellierung von

- ⇨ Sequenz
- ⇨ Entscheidung
- ⇨ Nebenläufigkeit
- ⇨ Synchronisation

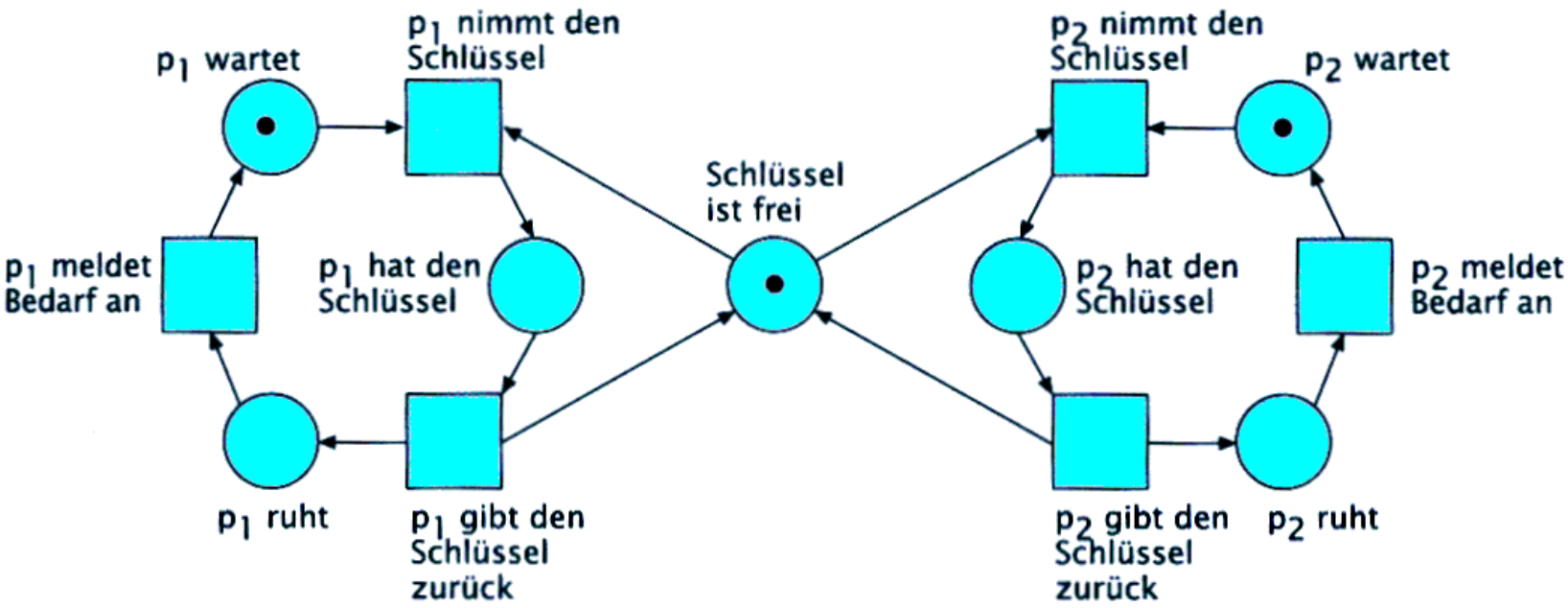
- Der *Vorbereich* $\bullet t_i$ einer Transition $t_i \in \boldsymbol{T}$ enthält alle Plätze die mit t_i über eine gerichtete Kante von dem jeweiligen Platz zu der Transition verbunden sind: $\bullet t_i = \{p \in \boldsymbol{P}: (p, t_i) \in \boldsymbol{F}\}$.
- Die Elemente von $\bullet t_i$ werden Pre-Plätze oder Eingangsplätze genannt.
- Der *Nachbereich* $t_i\bullet$ einer Transition $t_i \in \boldsymbol{T}$ enthält alle Plätze die mit t_i über eine gerichtete Kante von der Transition zu dem jeweiligen Platz verbunden sind: $t_i\bullet = \{p \in \boldsymbol{P}: (t_i, p) \in \boldsymbol{F}\}$.
- Die Elemente von $t_i\bullet$ heißen Post-Plätze oder Ausgangsplätze.
- *Vorbereich* $\bullet p_i$ und *Nachbereich* $p_i\bullet$ eines Platzes $p_i \in \boldsymbol{P}$ sind entsprechend definiert:
  $\bullet p_i = \{t \in \boldsymbol{T}: (t, p_i) \in \boldsymbol{F}\}$
  $p_i\bullet = \{t \in \boldsymbol{T}: (p_i, t) \in \boldsymbol{F}\}$

Strukturelemente von Petrinetzen



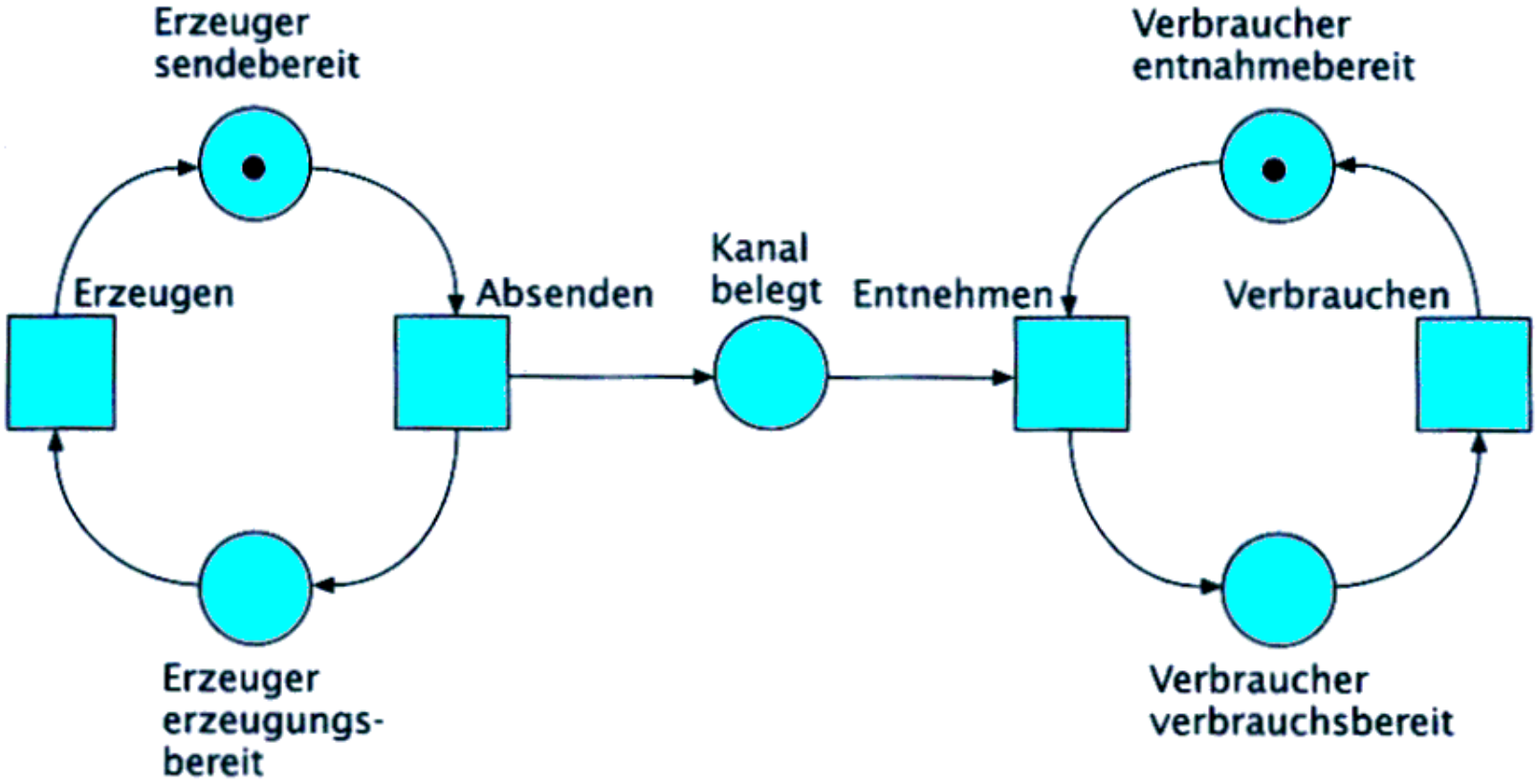
Quelle: Balzert

Beispiel: Leser und Schreiber im Petrinetz



Quelle: Balzert

Beispiel: Produzenten und Konsumenten



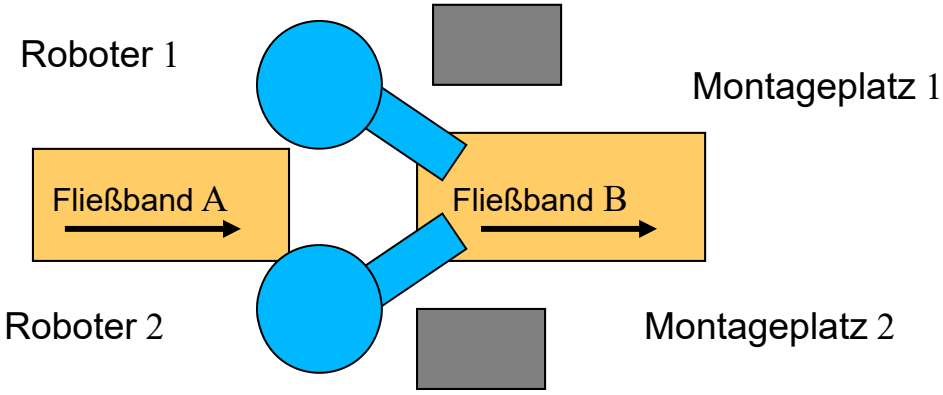
Quelle: Balzert

Beispiel: Montage

Zwei Roboter bestücken Leiterplatten mit elektronischen Bauelementen, die auf einem Fließband A antransportiert werden. Ist ein Roboter frei, dann kann er eine antransportierte Leiterplatte vom Fließband nehmen. Sind beide Roboter frei, dann wird nicht-deterministisch entschieden, welcher Roboter die Leiterplatte nimmt. Beide Roboter konkurrieren um die antransportierte Leiterplatte.

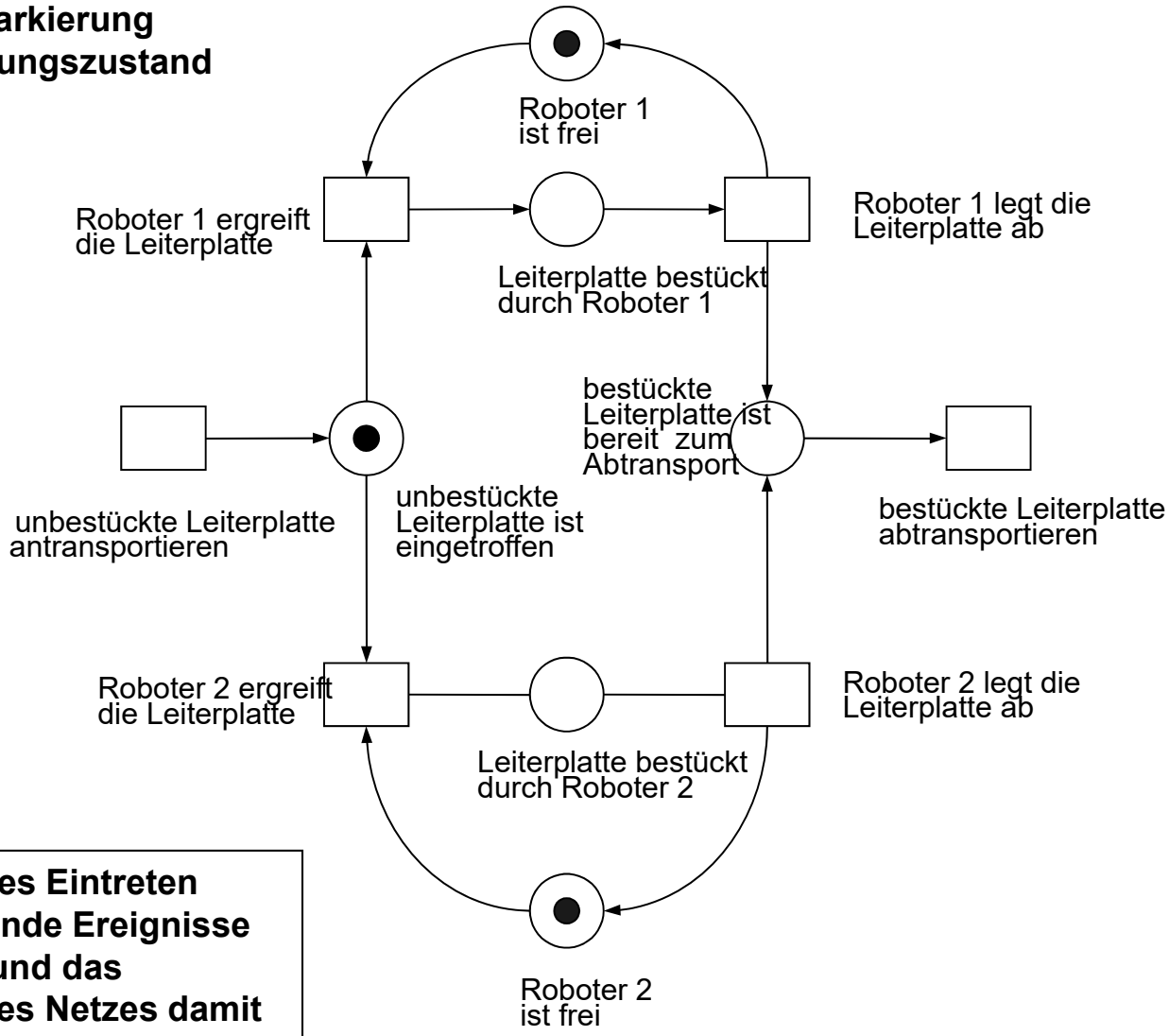
Sie arbeiten parallel, da die Bestückung mehr Zeit erfordert als die anderen Bearbeitungsschritte. Die Zeit zur Bestückung der Leiterplatte kann unterschiedlich lang sein. Ist eine Leiterplatte bestückt, wird sie von dem jeweiligen Roboter auf das Fließband B gelegt und abtransportiert.

Sind beide Roboter gleichzeitig fertig, dann wird nicht-deterministisch bestimmt, welche Roboter die Leiterplatte zuerst ablegt. Durch Lichtschranken gesteuert, werden die Fließbänder gestartet und gestoppt.



B/E-Netz des Bestückungsroboters vor dem Schalten

Anfangsmarkierung /Initialisierungszustand

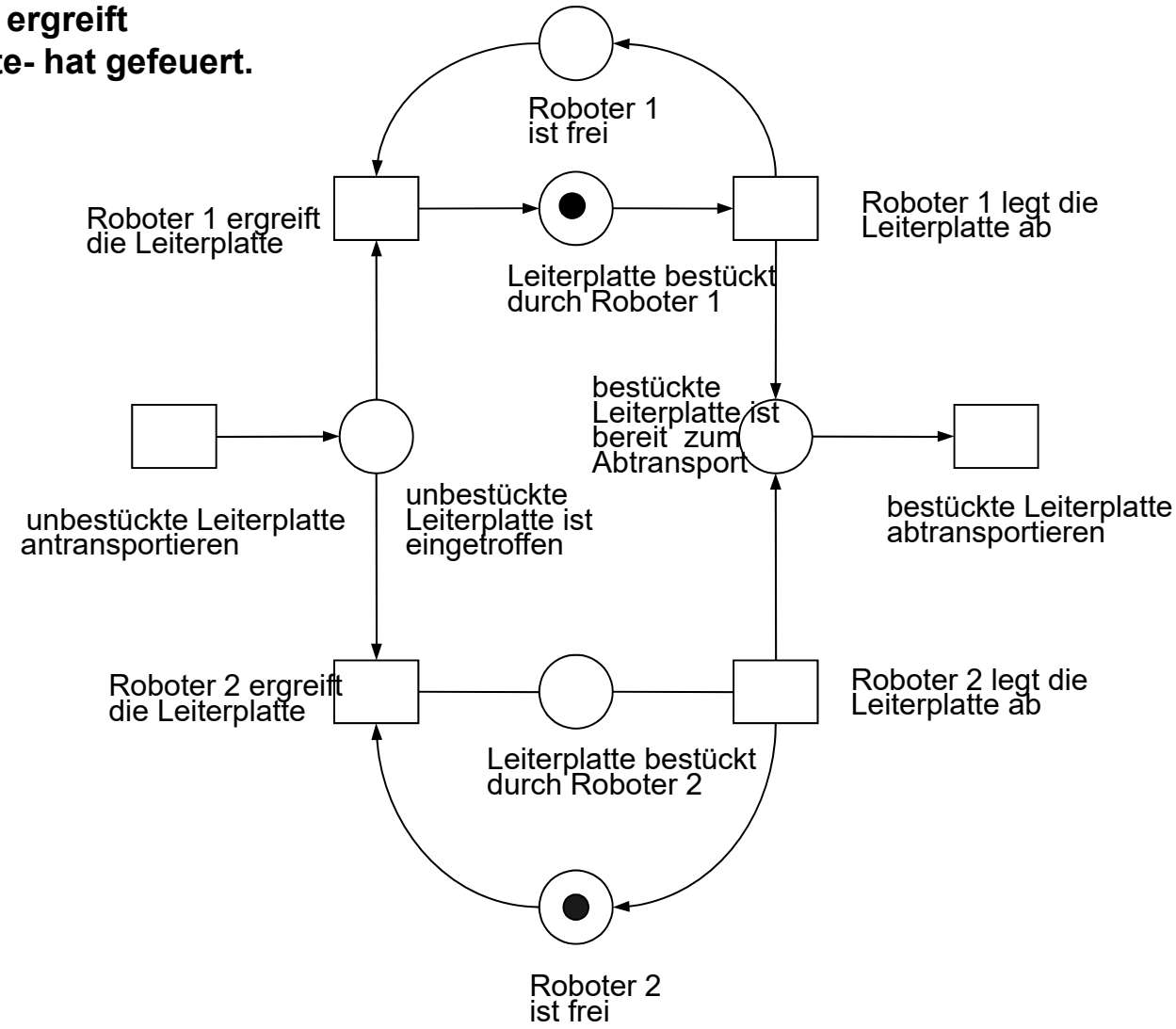


Tatsächliches Eintreten konkurrierende Ereignisse ist zufällig und das Verhalten des Netzes damit nicht mehr deterministisch.

Quelle: Balzert

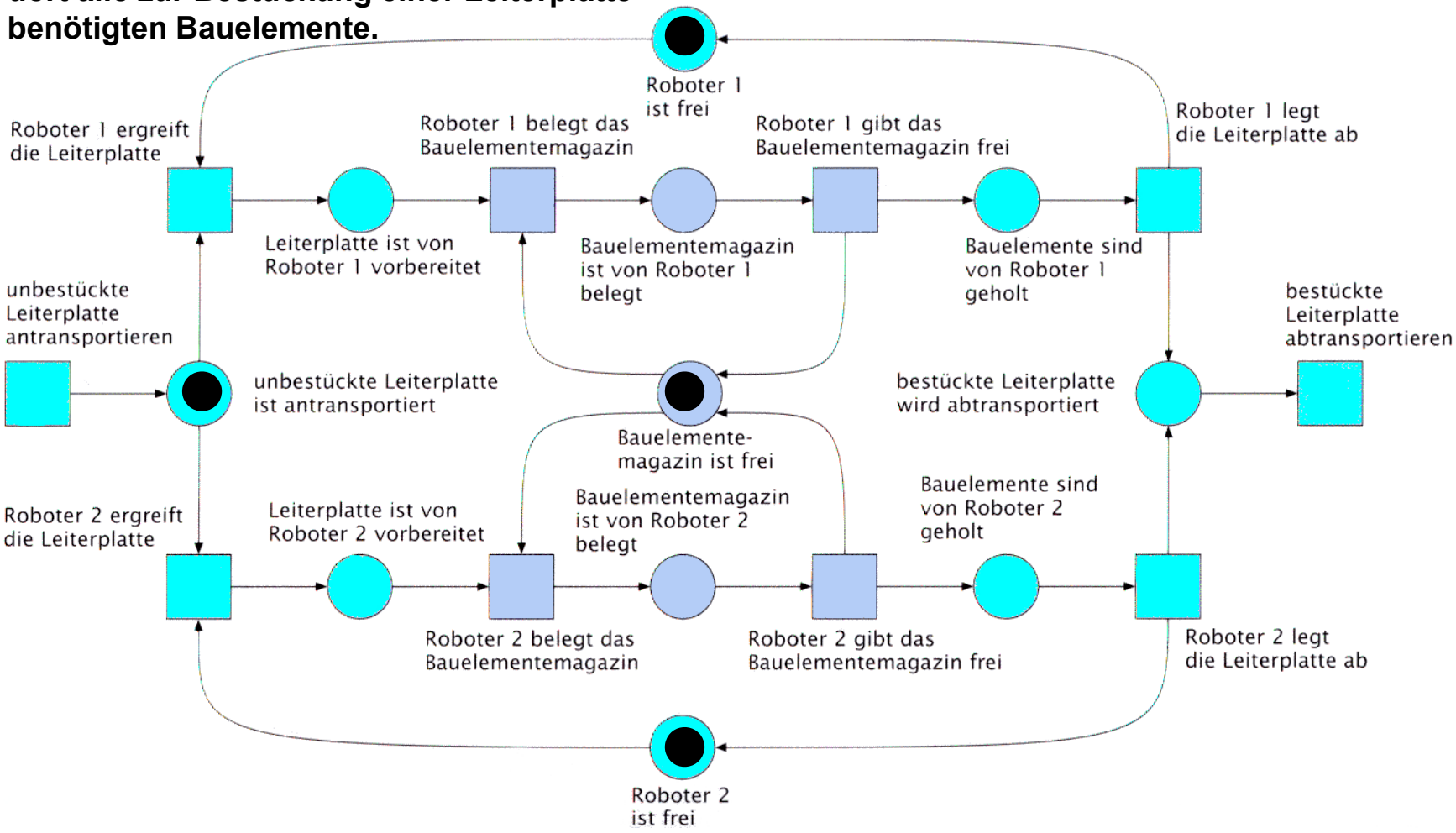
B/E-Netz des Bestückungsroboters nach dem Schalten

Roboter 1 ergreift Leiterplatte- hat gefeuert.



Quelle: Balzert

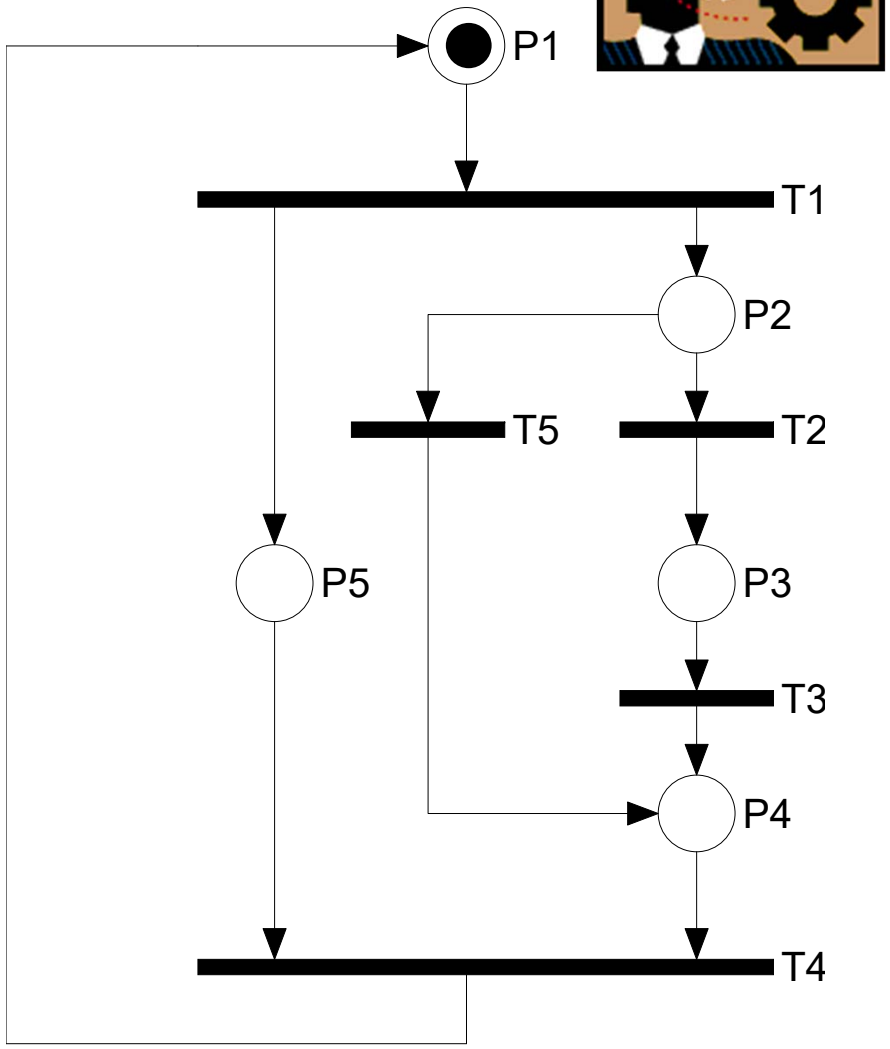
Beide Roboter greifen auf ein gemeinsames Bauteilemagazin zu, holen dort alle zur Bestückung einer Leiterplatte benötigten Bauelemente.



Quelle: Balzert

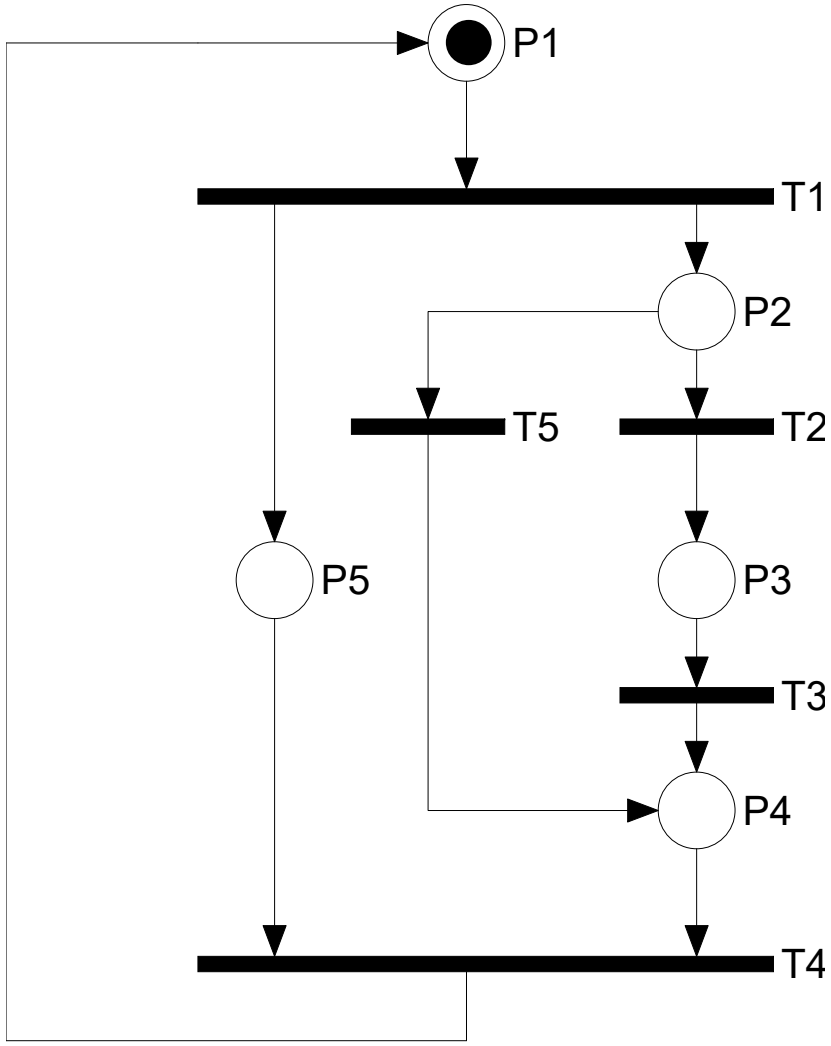
Beispiel einer Netzstruktur

SM 1



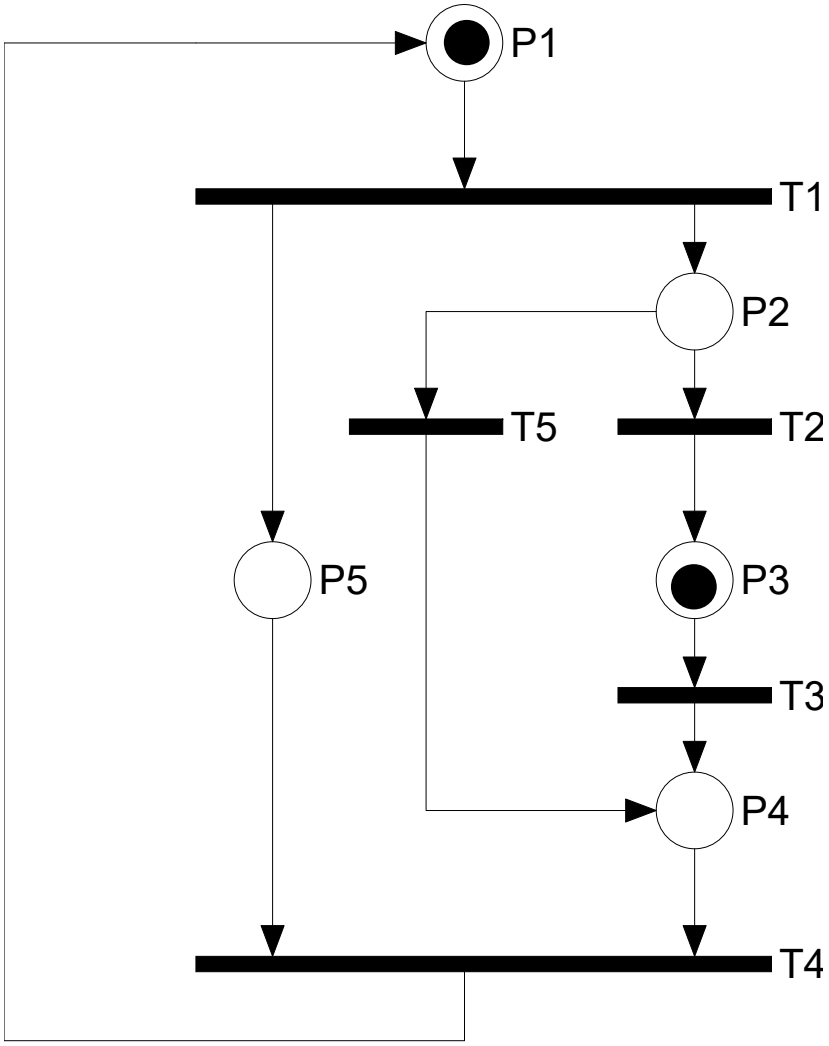
Beispiel mit Anfangsmarkierung 1

- P = {P1, P2, P3, P4, P5}
- T = {T1, T2, T3, T4, T5}
- F = {(P1, T1), (T1, P2),
  (P2, T2), (T2, P3),
  (P3, T3), (T3, P4),
  (P4, T4), (T4, P1),
  (T1, P5), (P5, T4),
  (P2, T5), (T5, P4)}
- $\mathbf{m}_0 = (1, 0, 0, 0, 0)$

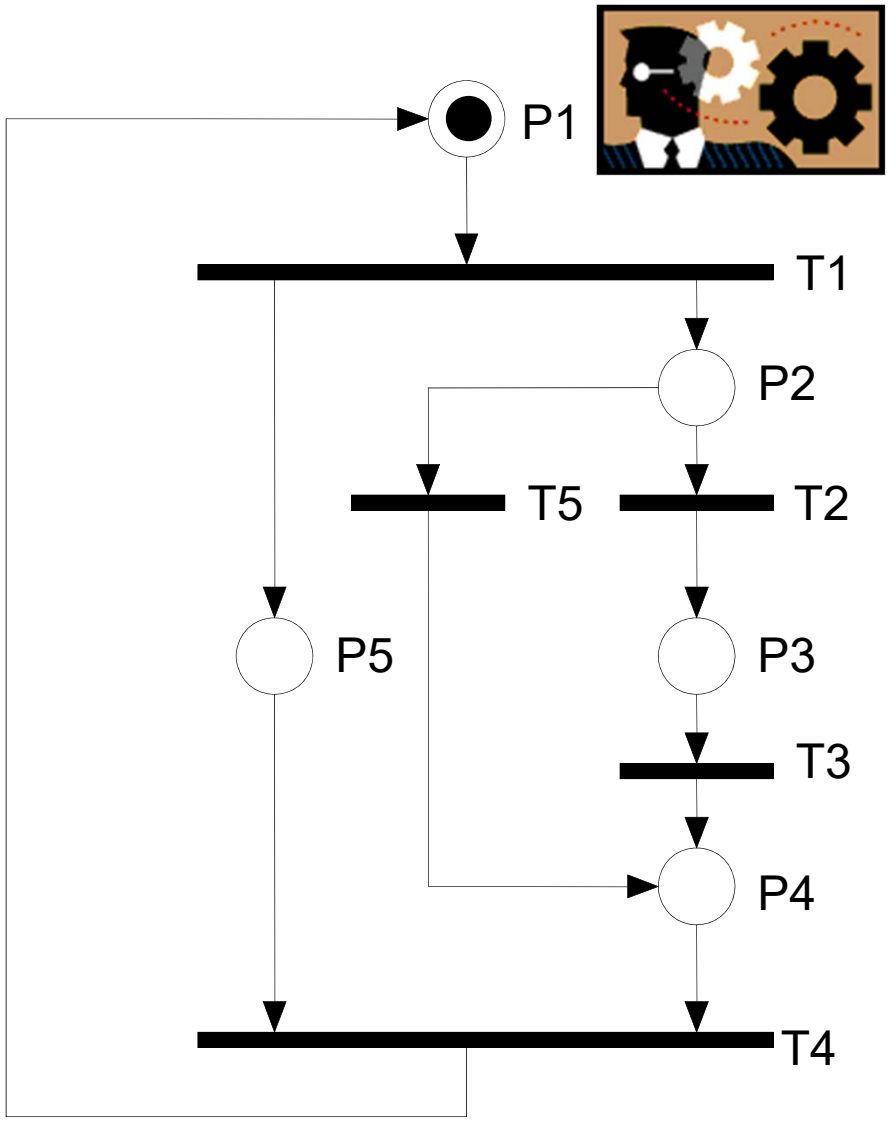


Beispiel mit Anfangsmarkierung 2

- P = {P1, P2, P3, P4, P5}
- T = {T1, T2, T3, T4, T5}
- F = {(P1, T1), (T1, P2),
  (P2, T2), (T2, P3),
  (P3, T3), (T3, P4),
  (P4, T4), (T4, P1),
  (T1, P5), (P5, T4),
  (P2, T5), (T5, P4)}
- $\mathbf{m}_0$ = (1, 0, 1, 0, 0)

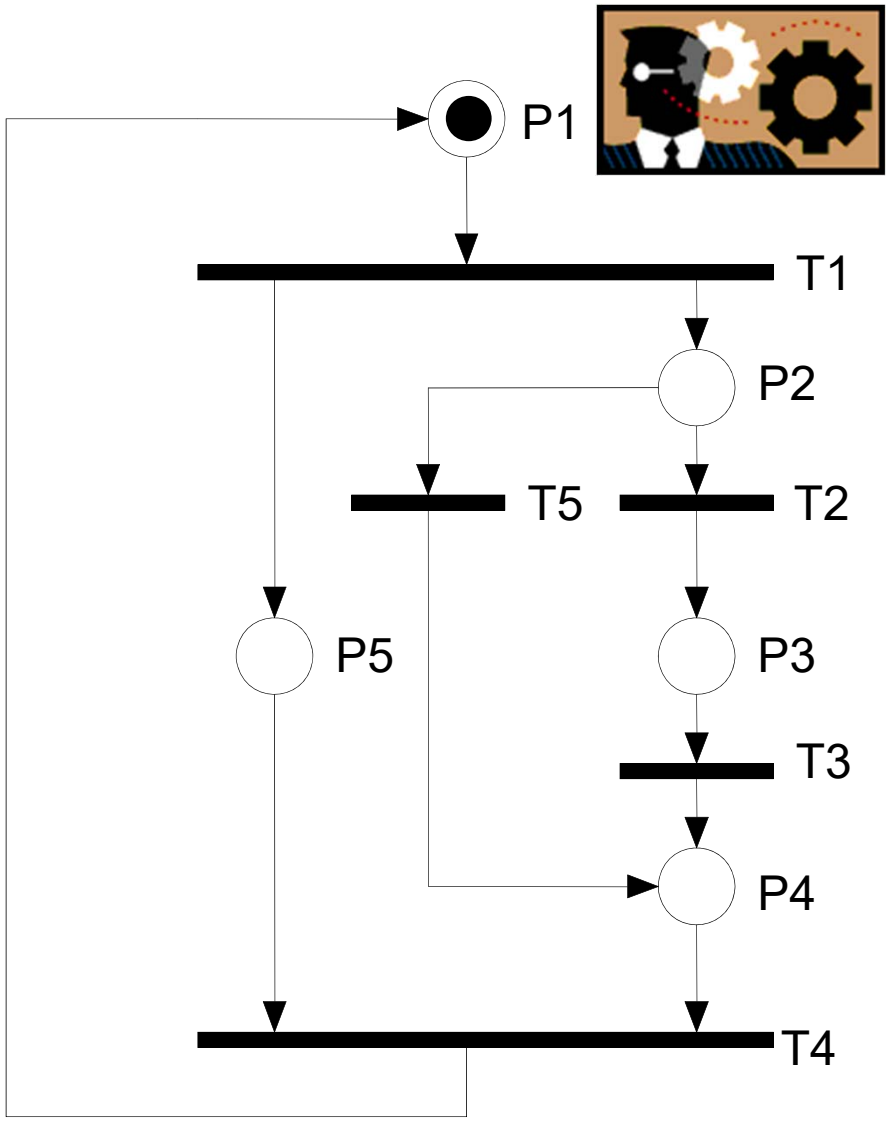


Erreichbarkeitsgraph und Erreichbarkeitsmenge zu Bsp. 1

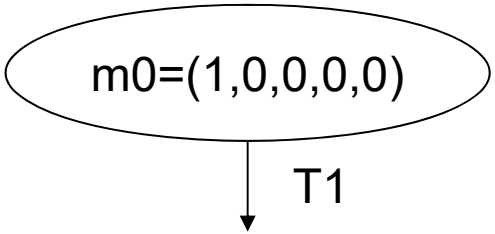


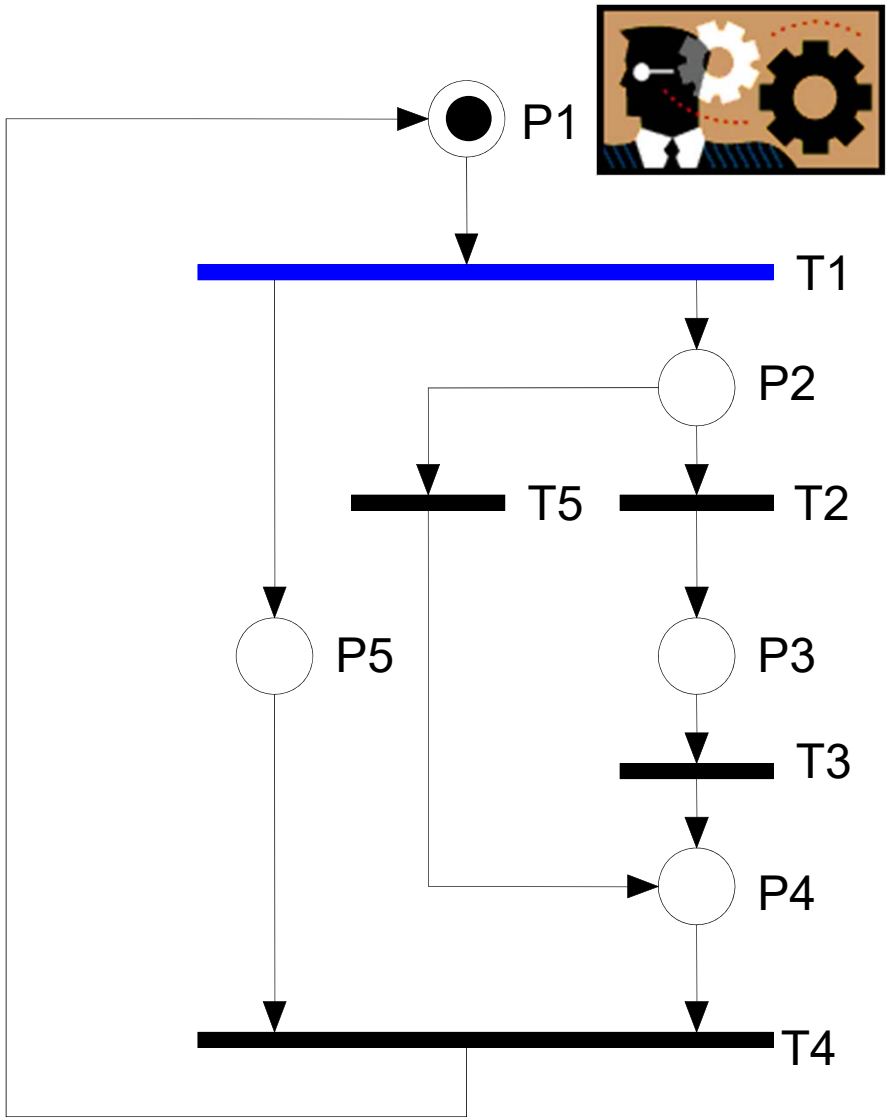
Erreichbarkeitsgraph und Erreichbarkeitsmenge zu Bsp. 1

m0=(1,0,0,0,0)

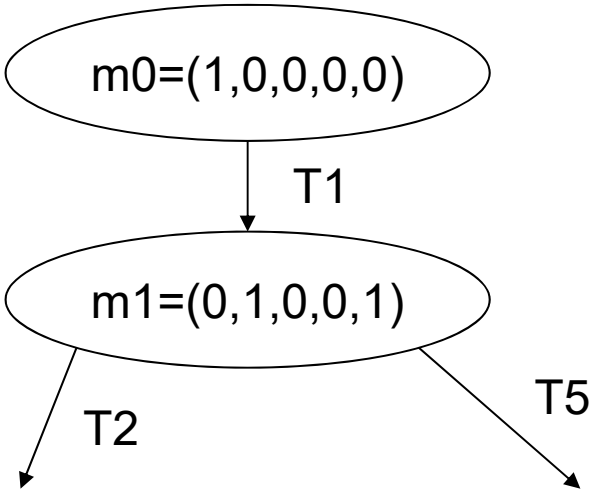


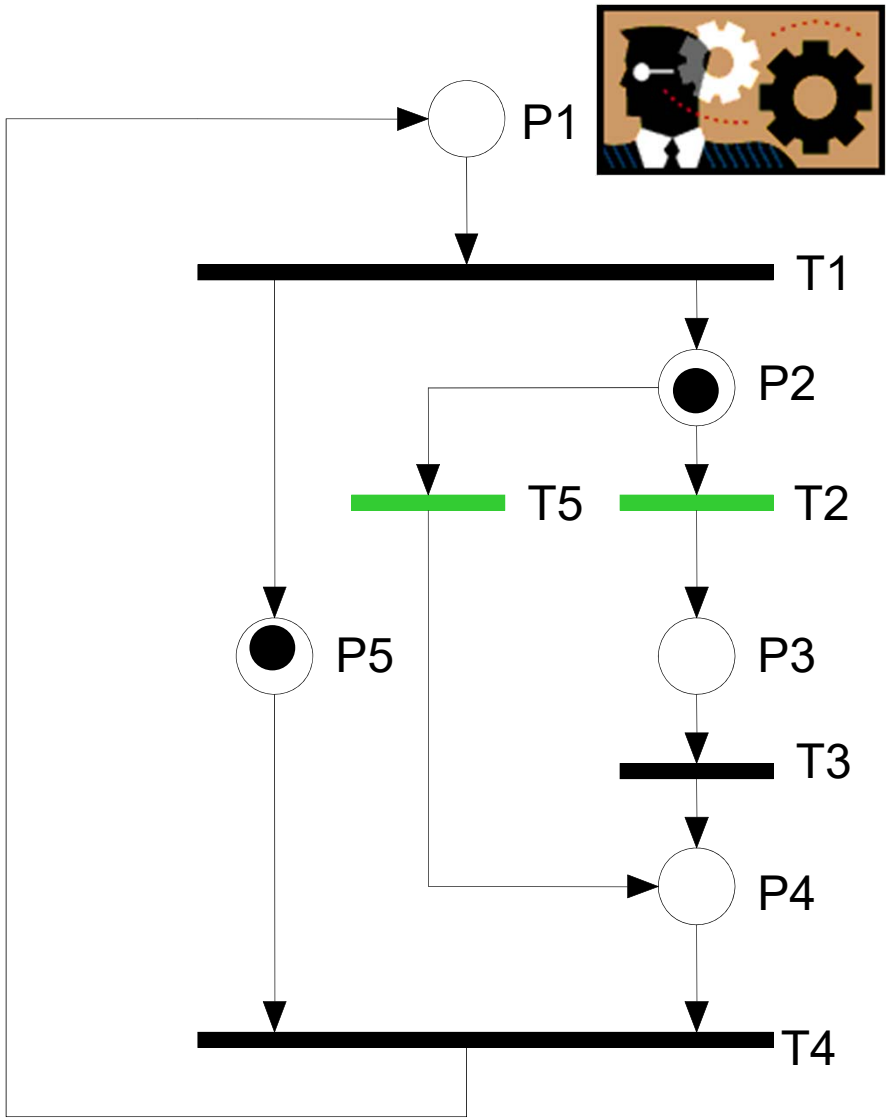
Erreichbarkeitsgraph und Erreichbarkeitsmenge zu Bsp. 1



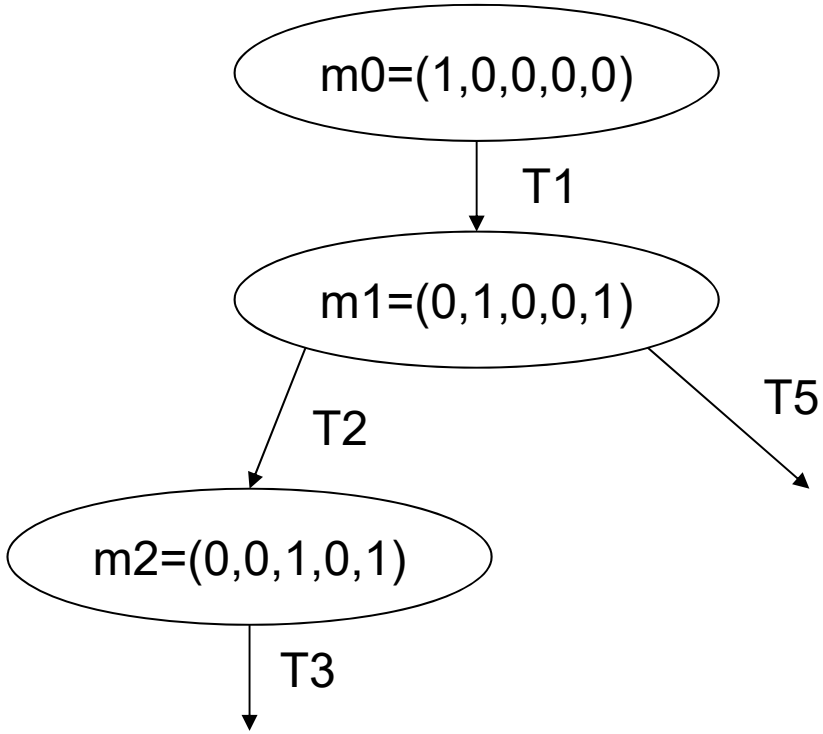


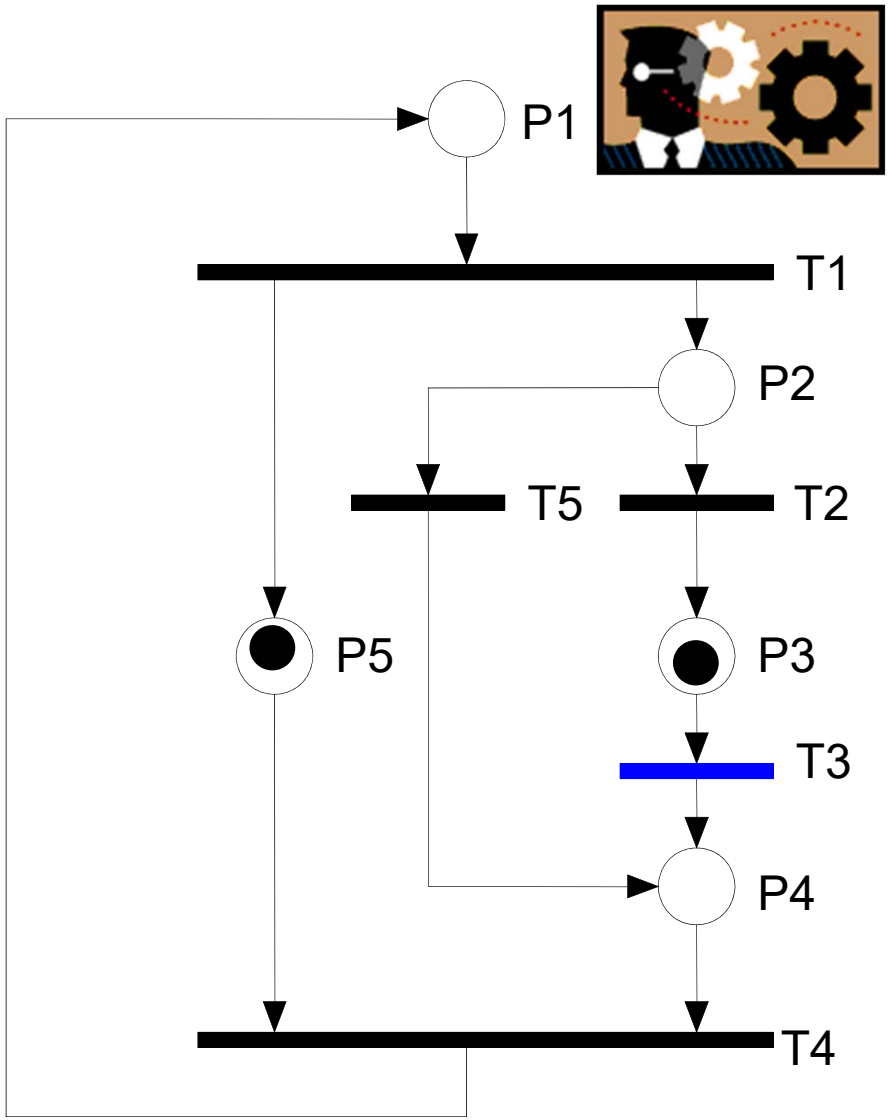
Erreichbarkeitsgraph und Erreichbarkeitsmenge zu Bsp. 1



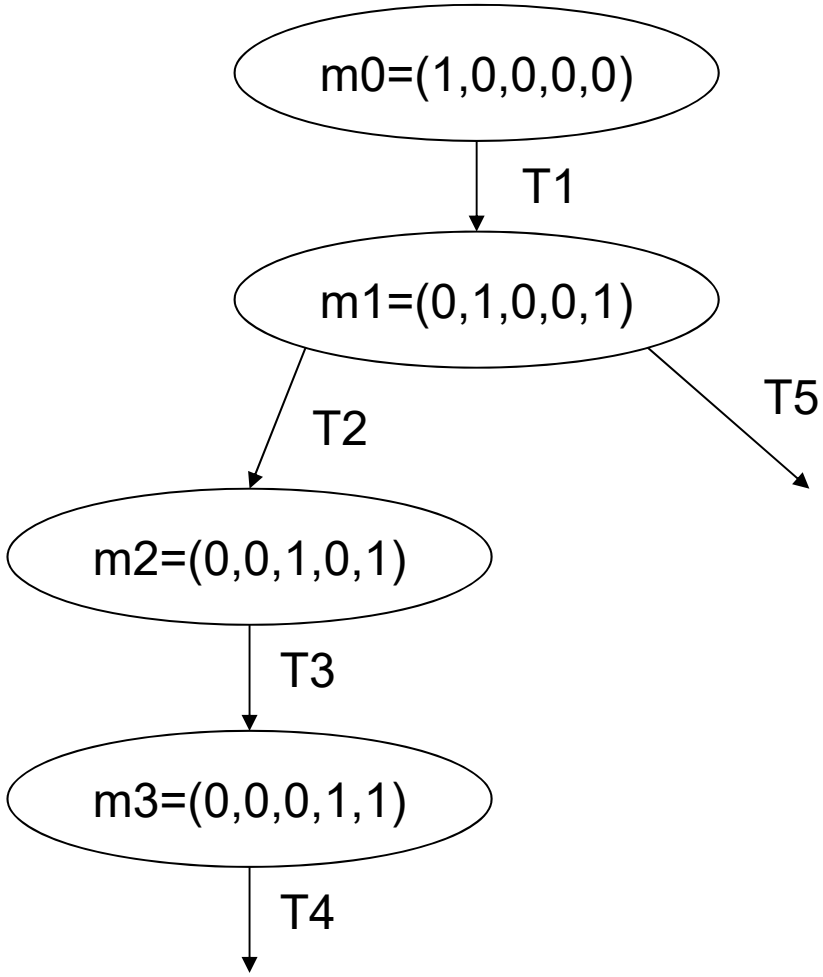


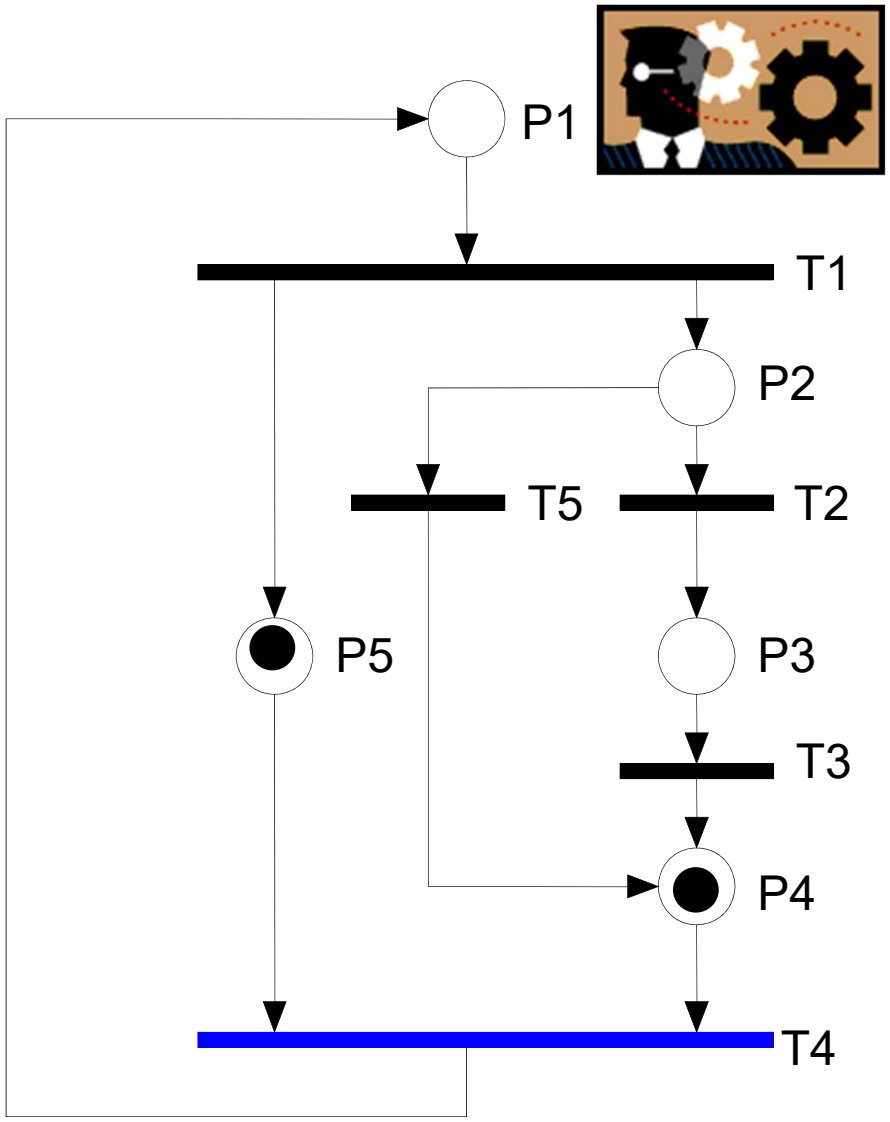
Erreichbarkeitsgraph und Erreichbarkeitsmenge zu Bsp. 1



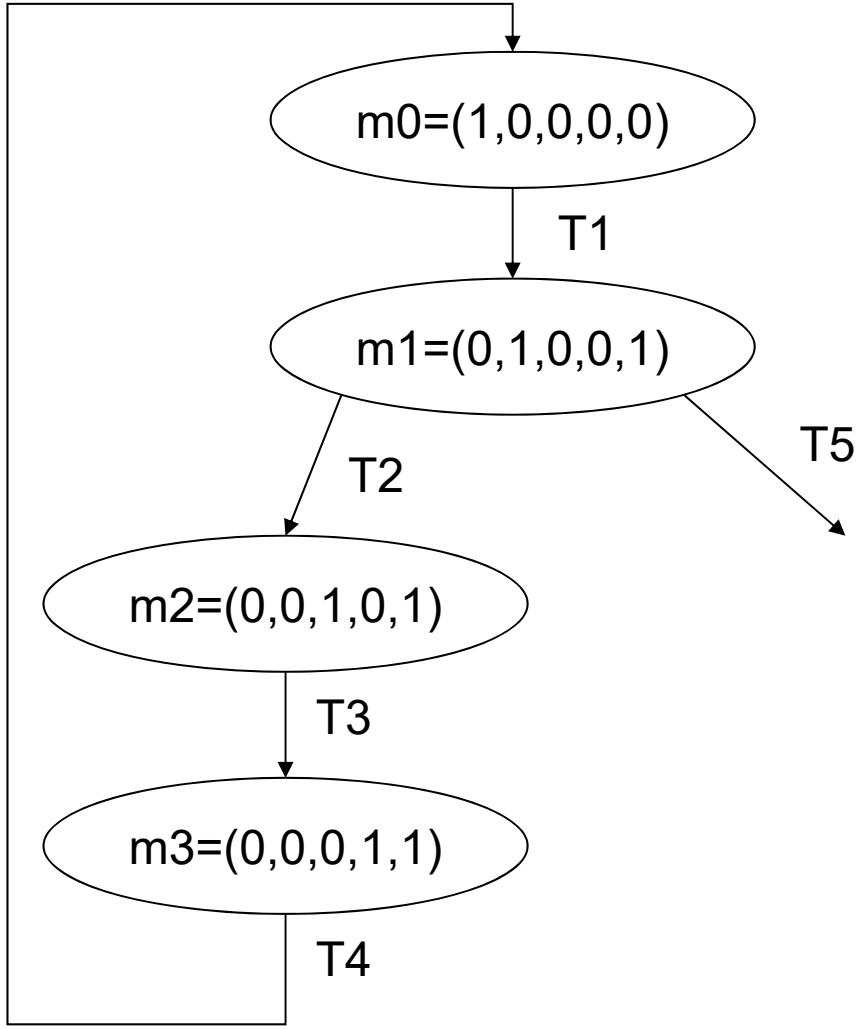


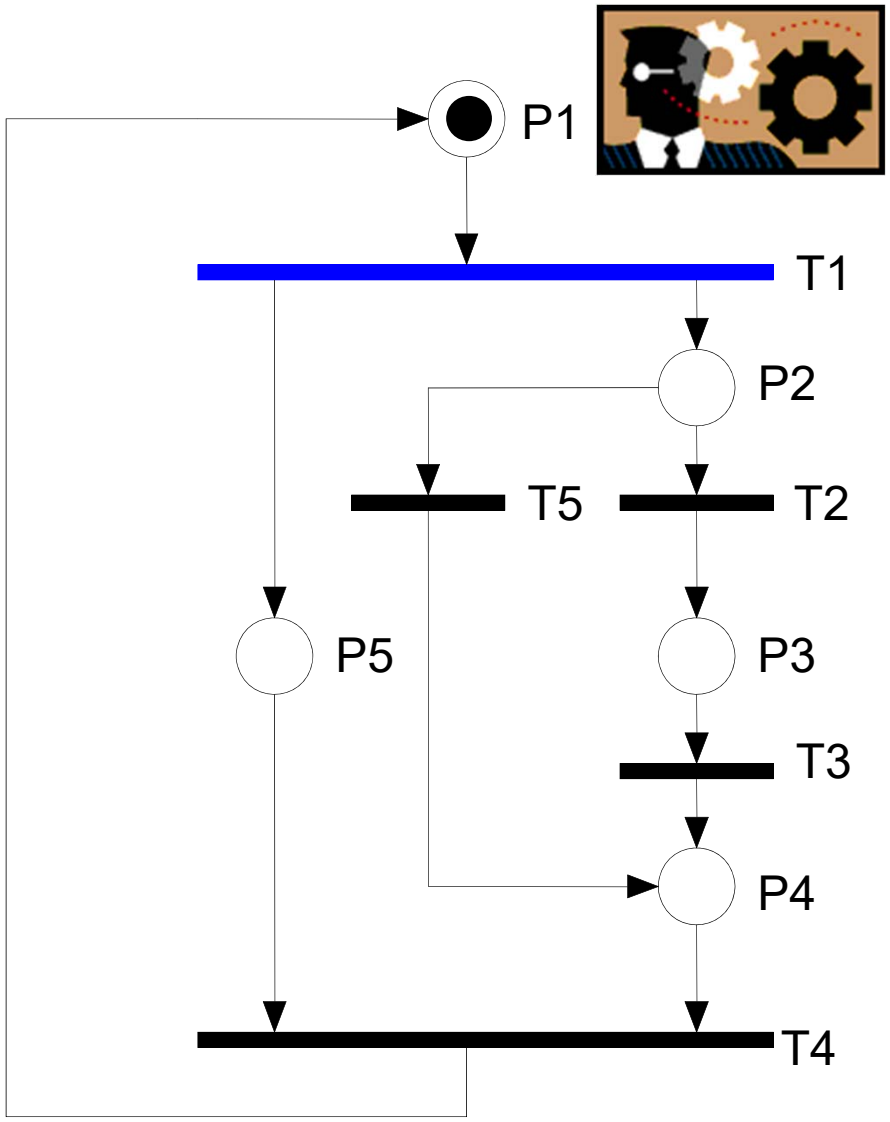
Erreichbarkeitsgraph und Erreichbarkeitsmenge zu Bsp. 1



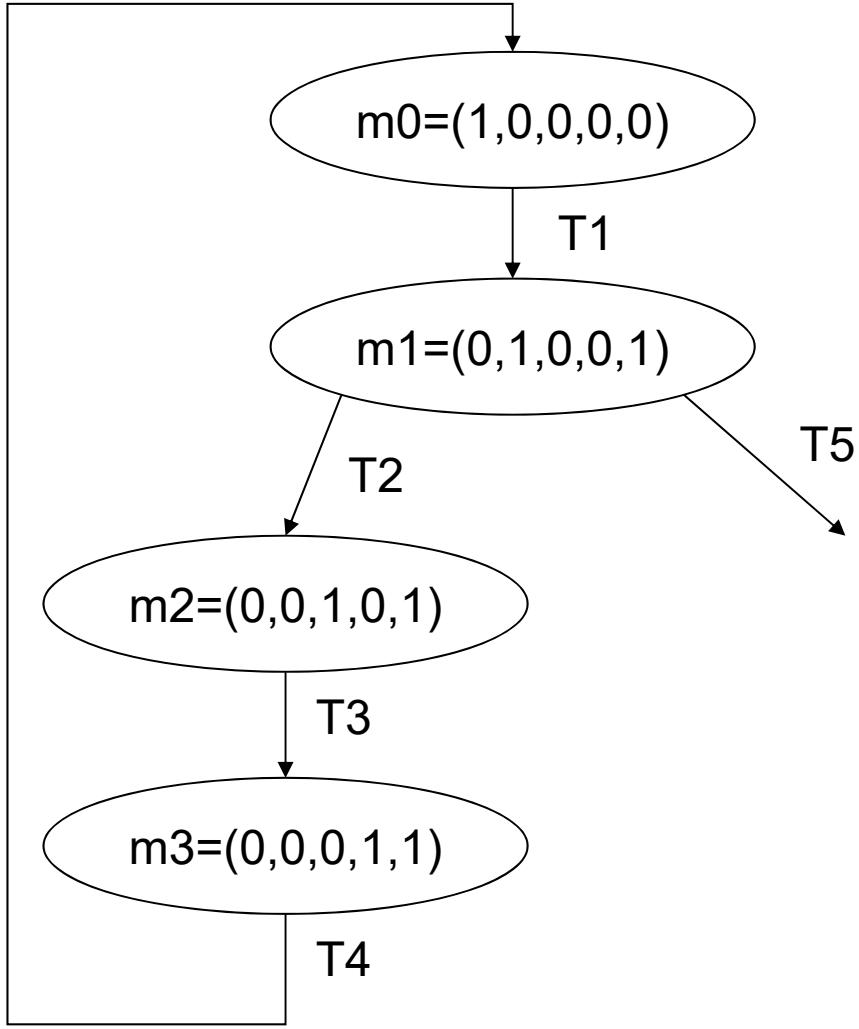


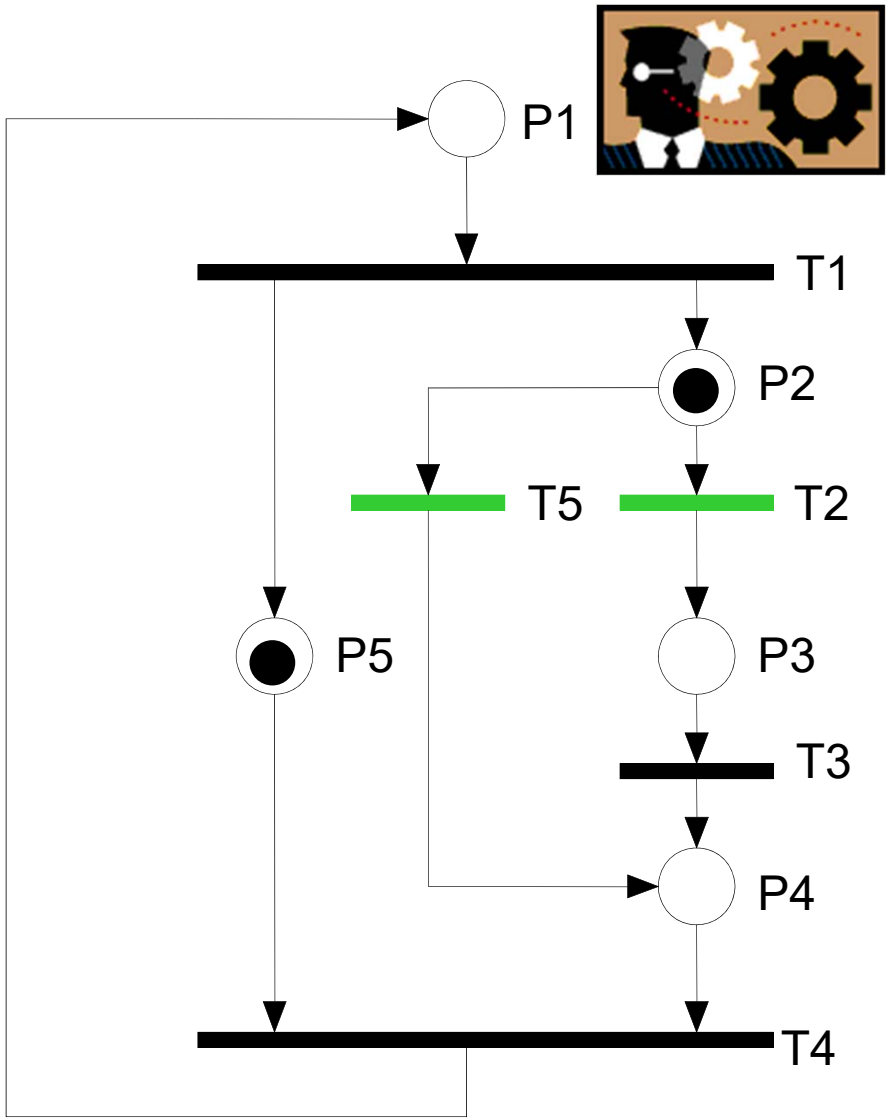
Erreichbarkeitsgraph und Erreichbarkeitsmenge zu Bsp. 1



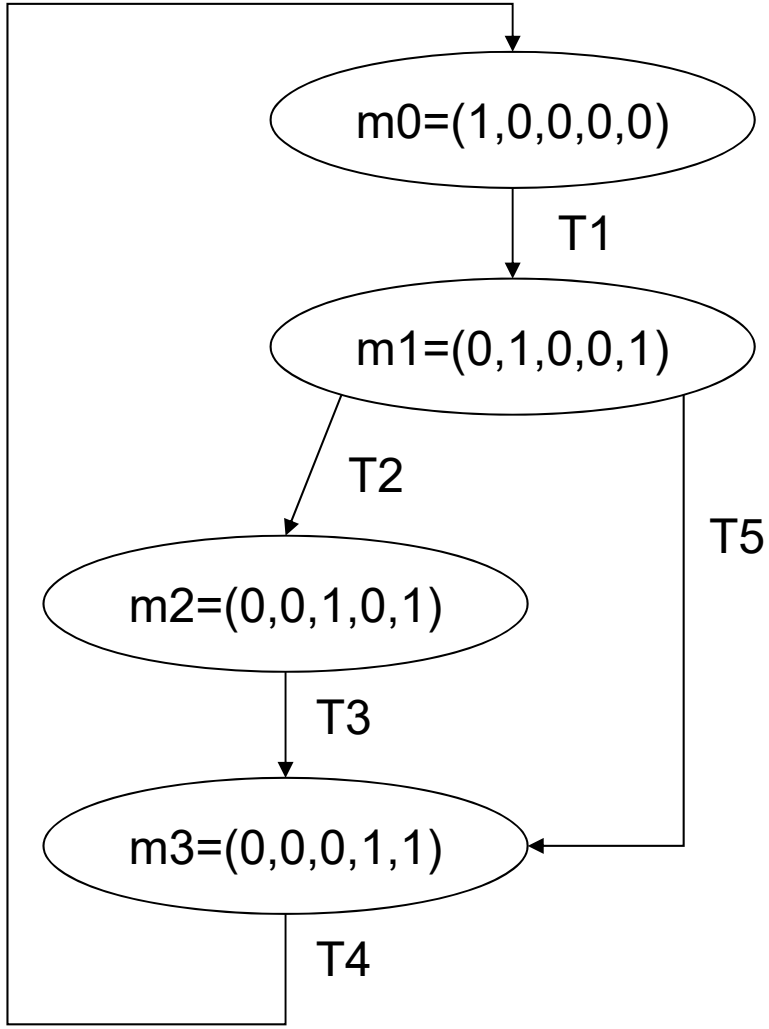


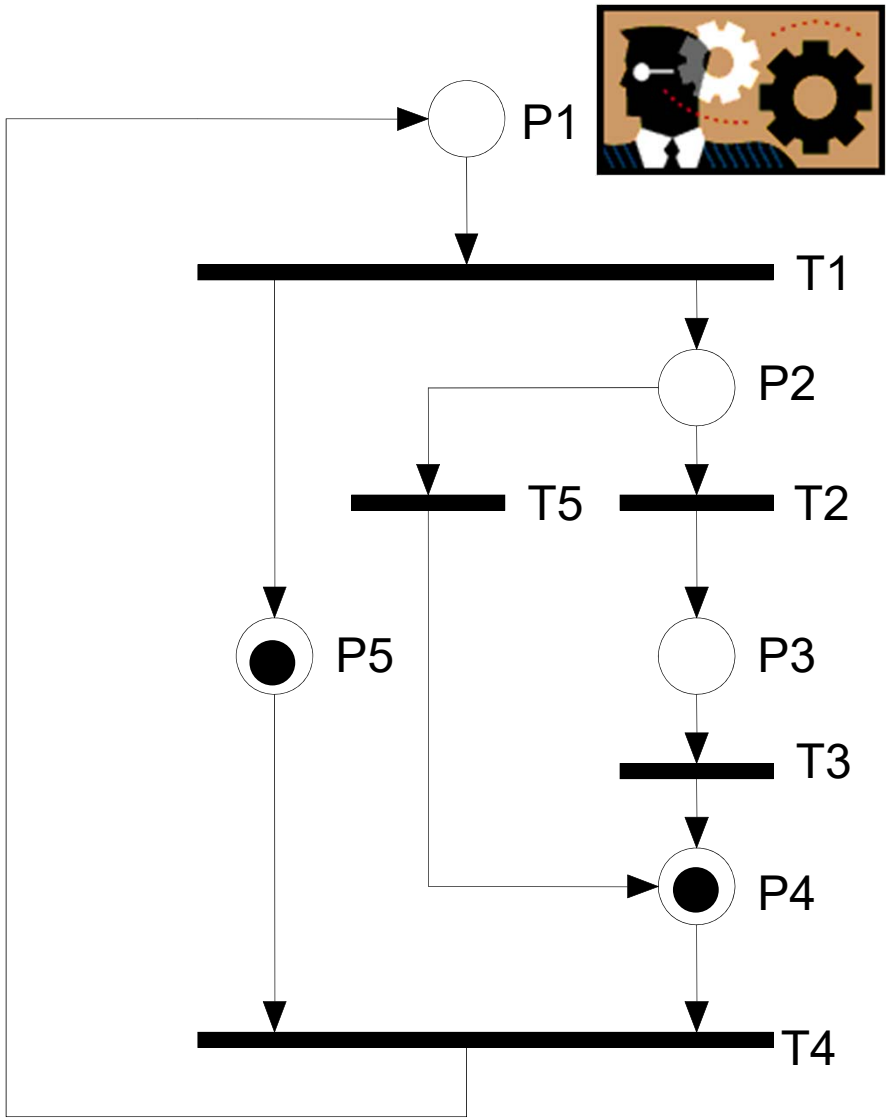
Erreichbarkeitsgraph und Erreichbarkeitsmenge zu Bsp. 1



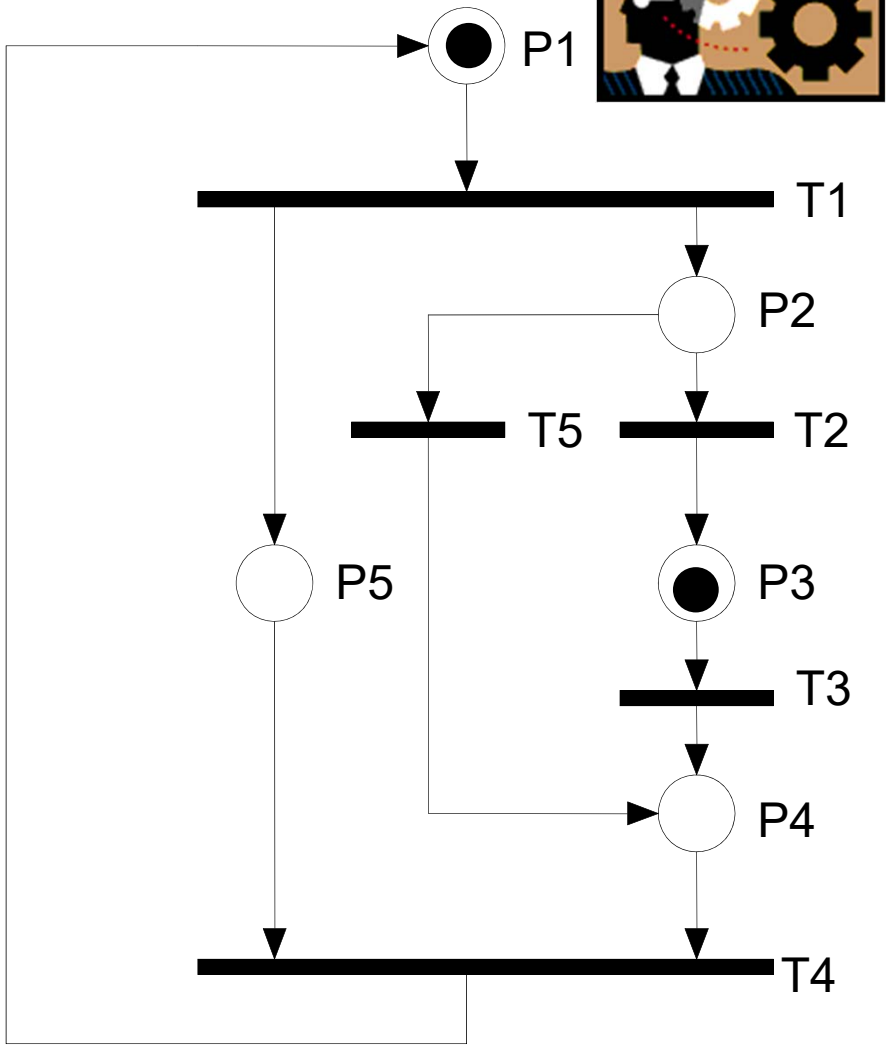


Erreichbarkeitsgraph und Erreichbarkeitsmenge zu Bsp. 1

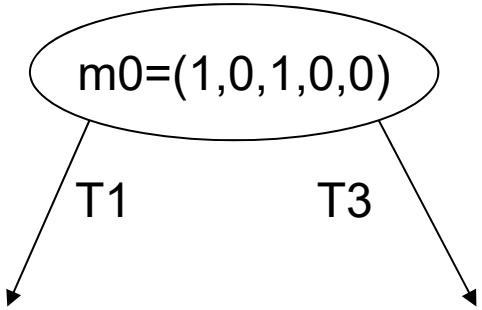


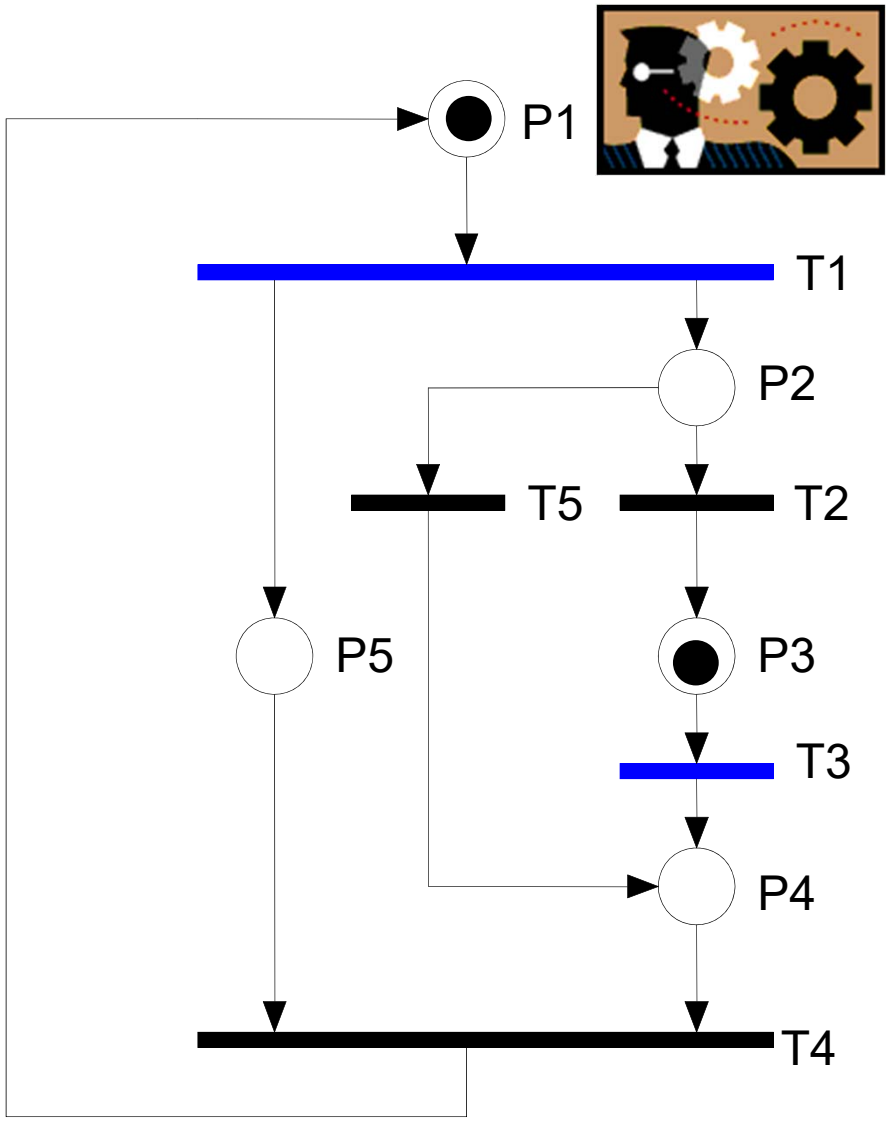


Erreichbarkeitsgraph und Erreichbarkeitsmenge zu Bsp. 2

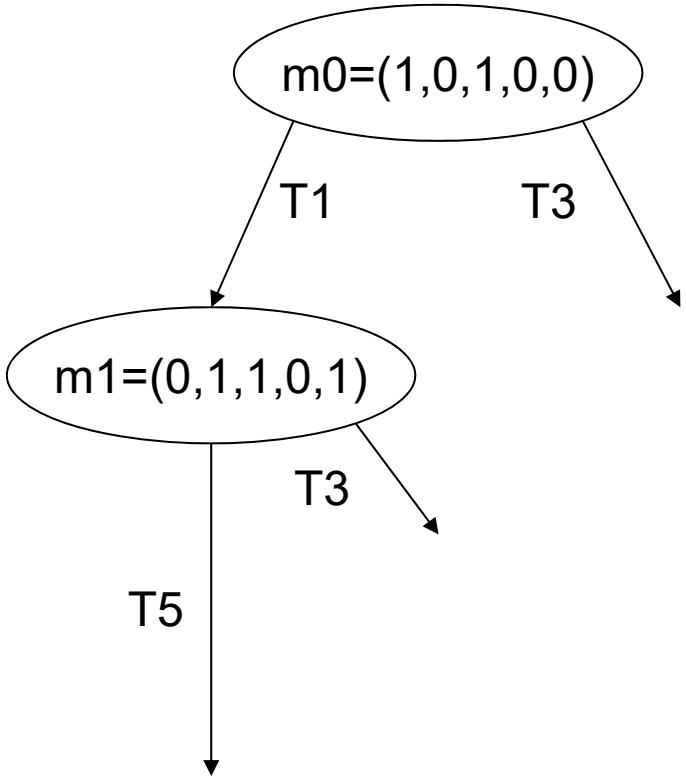


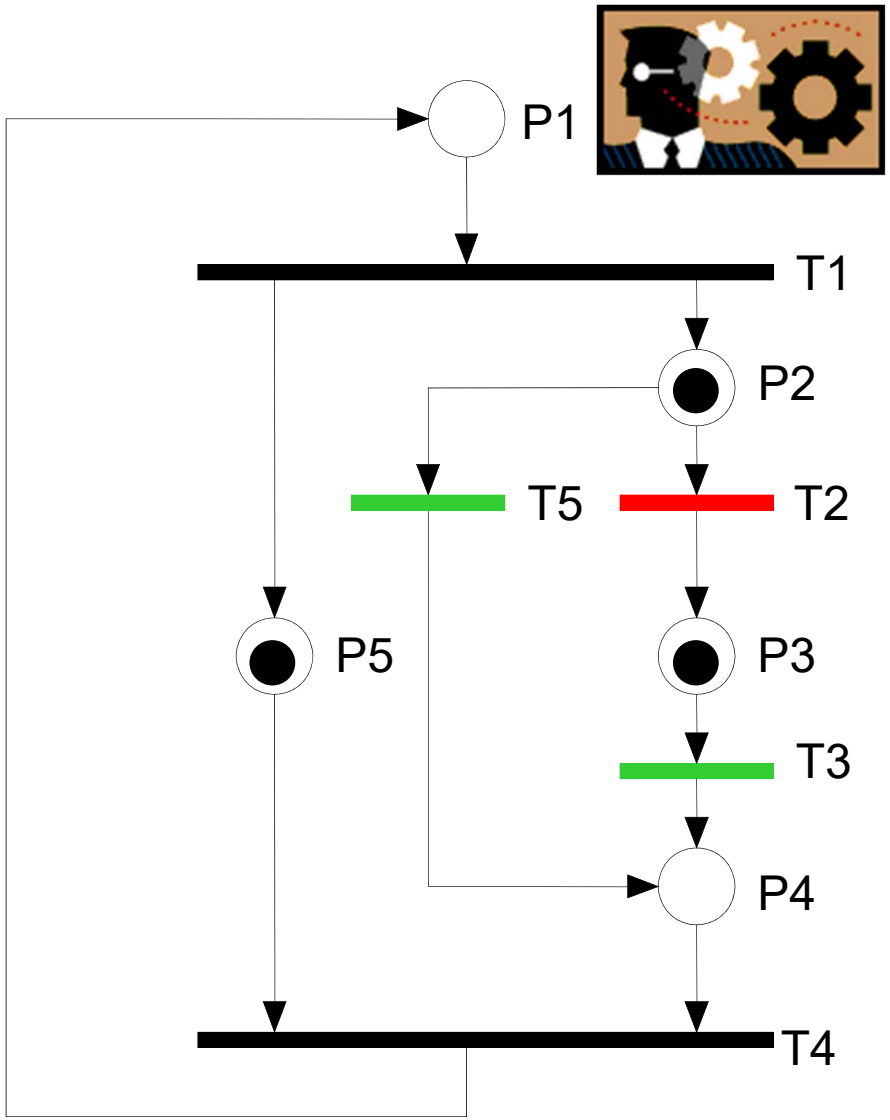
Erreichbarkeitsgraph und Erreichbarkeitsmenge zu Bsp. 2



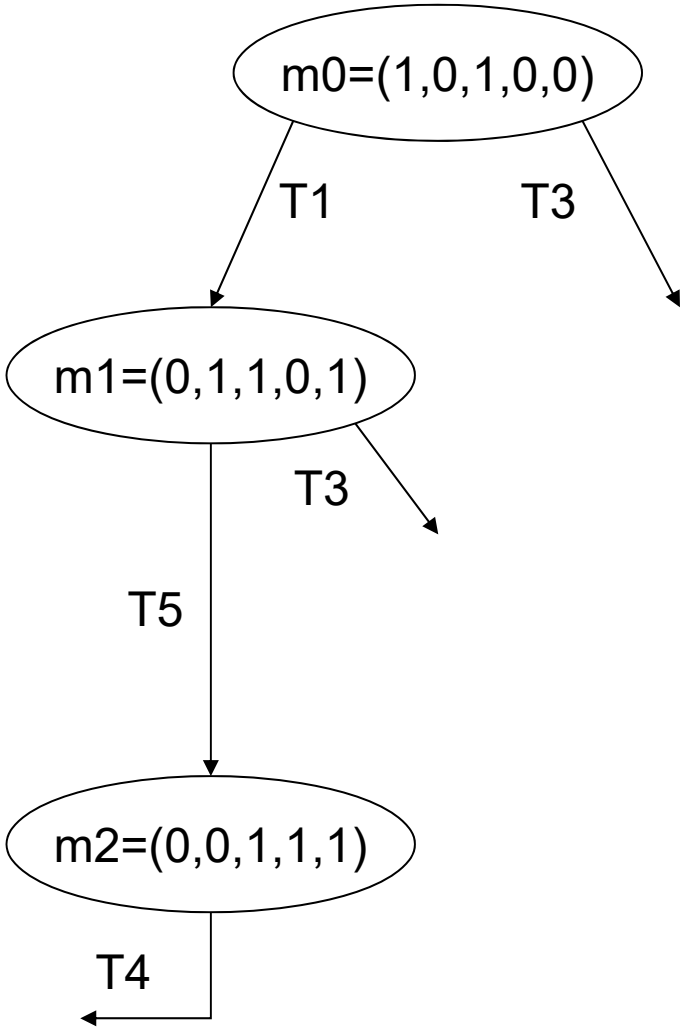


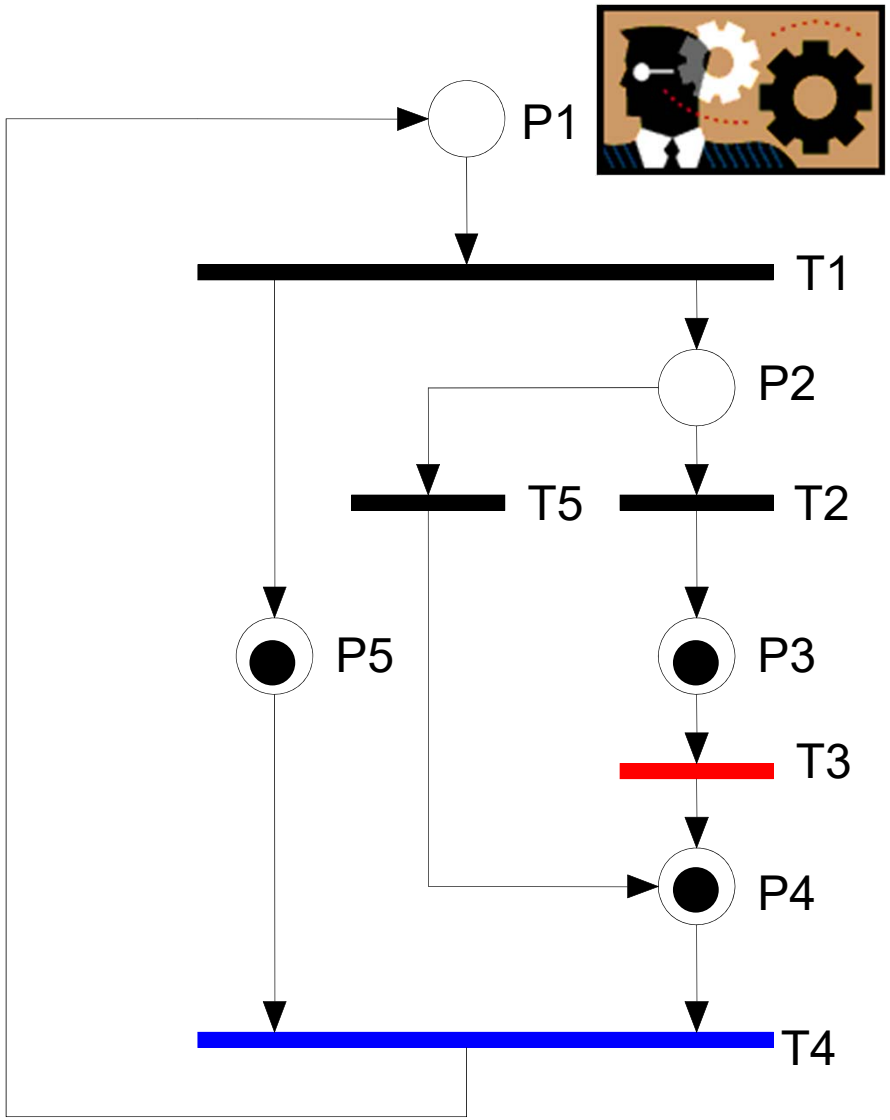
Erreichbarkeitsgraph und Erreichbarkeitsmenge zu Bsp. 2



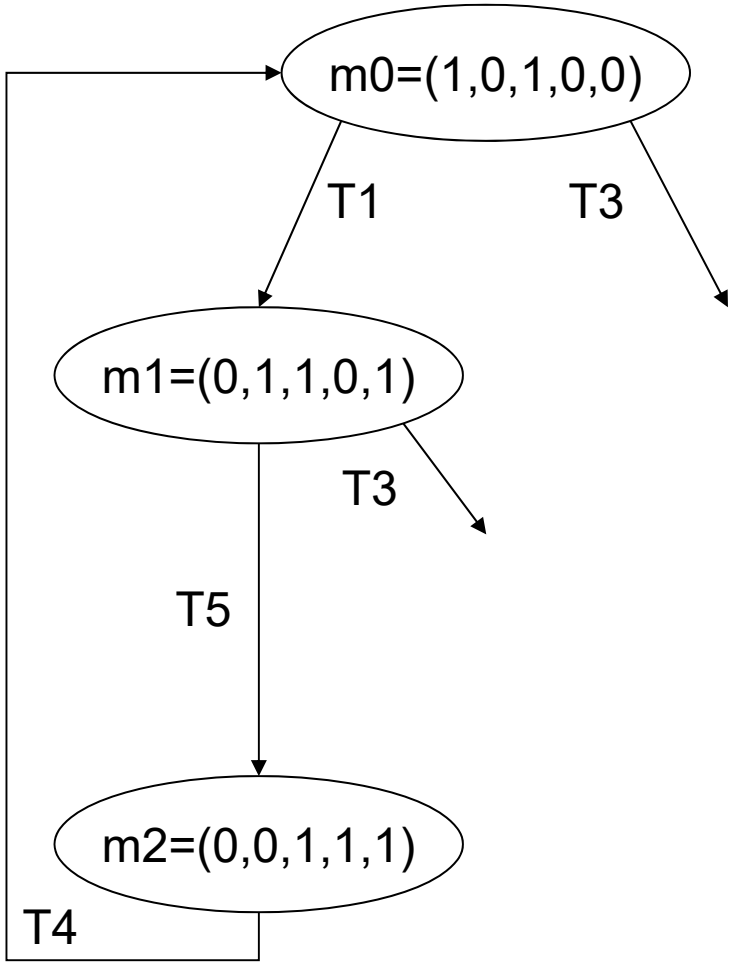


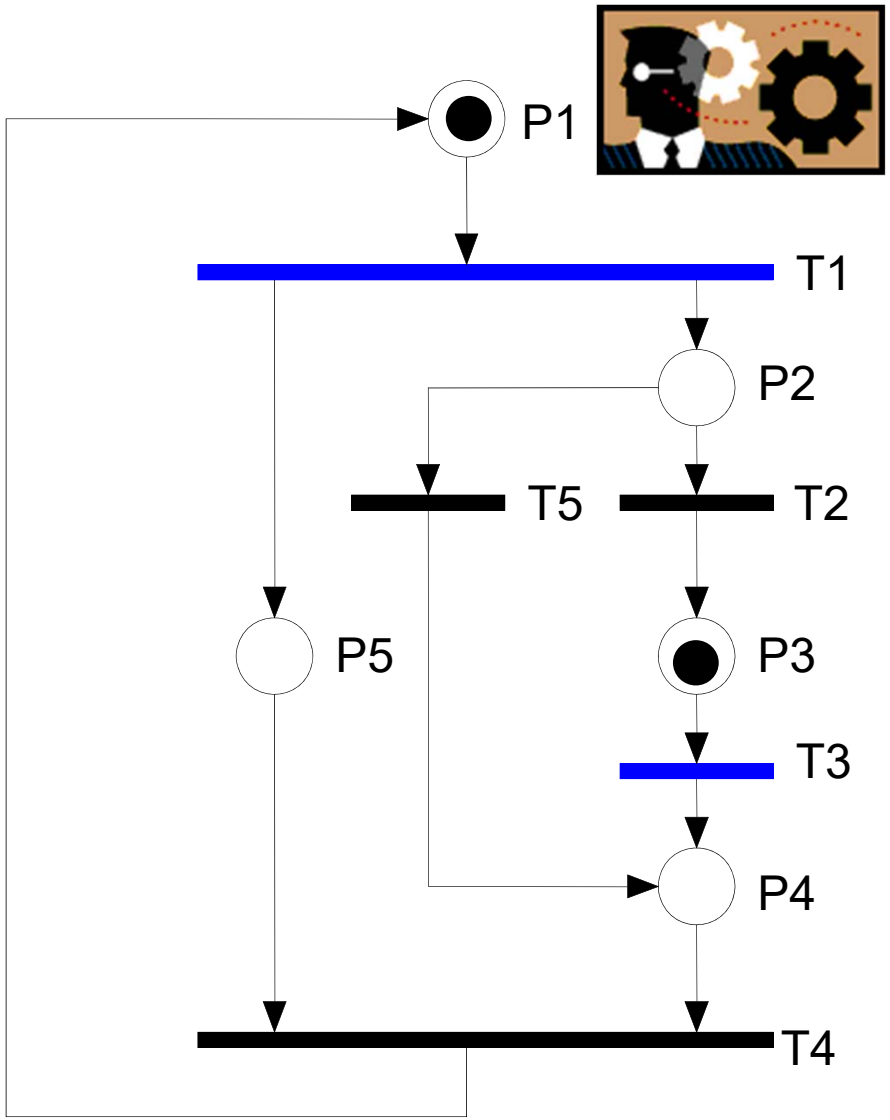
Erreichbarkeitsgraph und Erreichbarkeitsmenge zu Bsp. 2



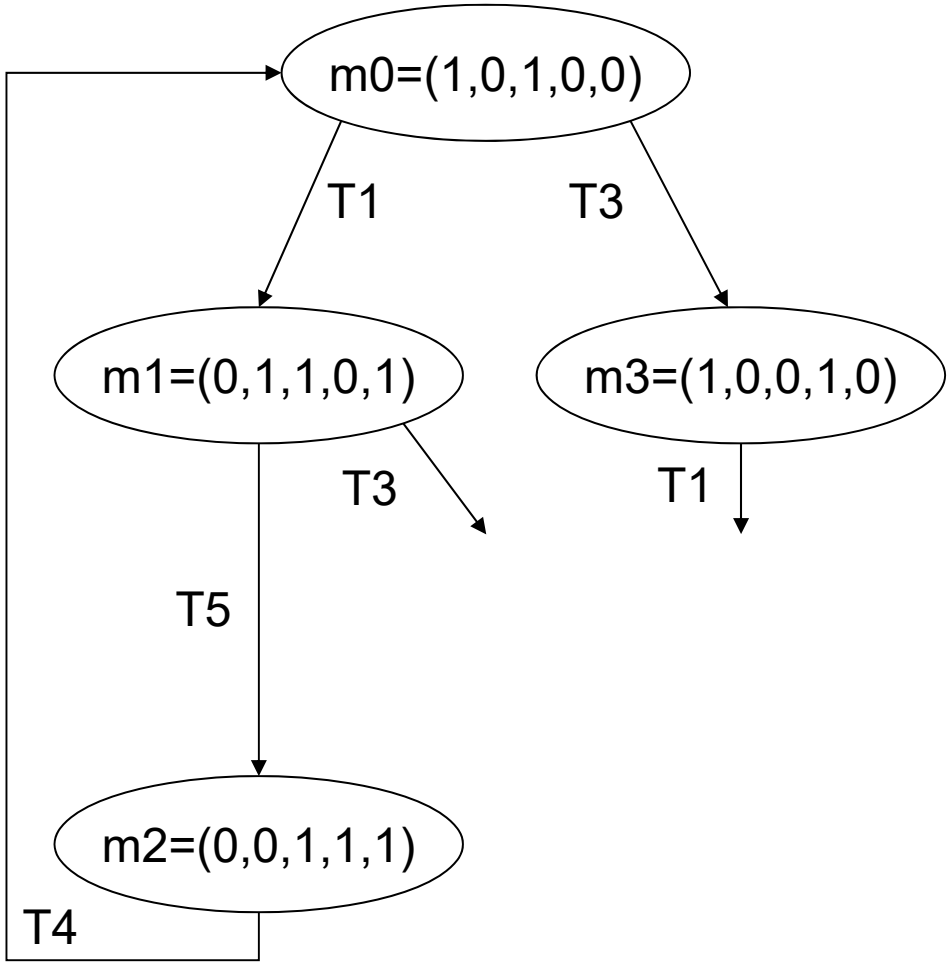


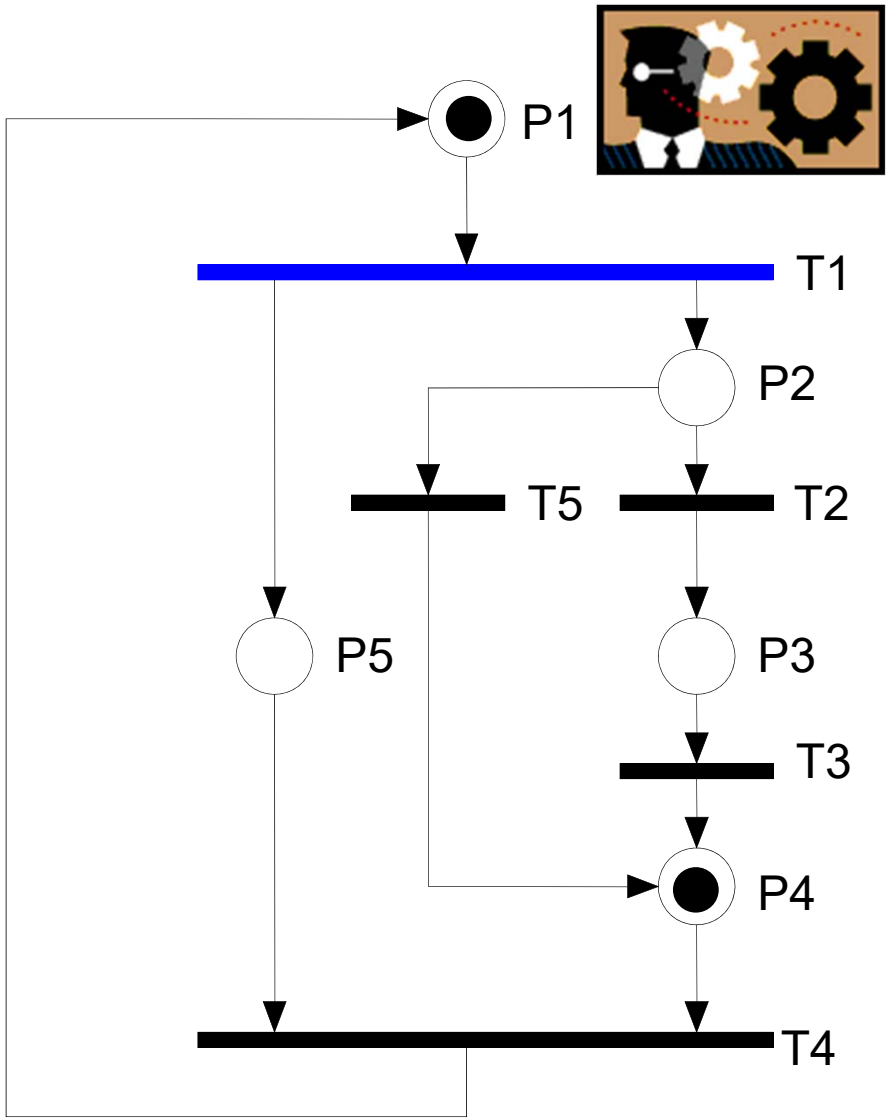
Erreichbarkeitsgraph und Erreichbarkeitsmenge zu Bsp. 2



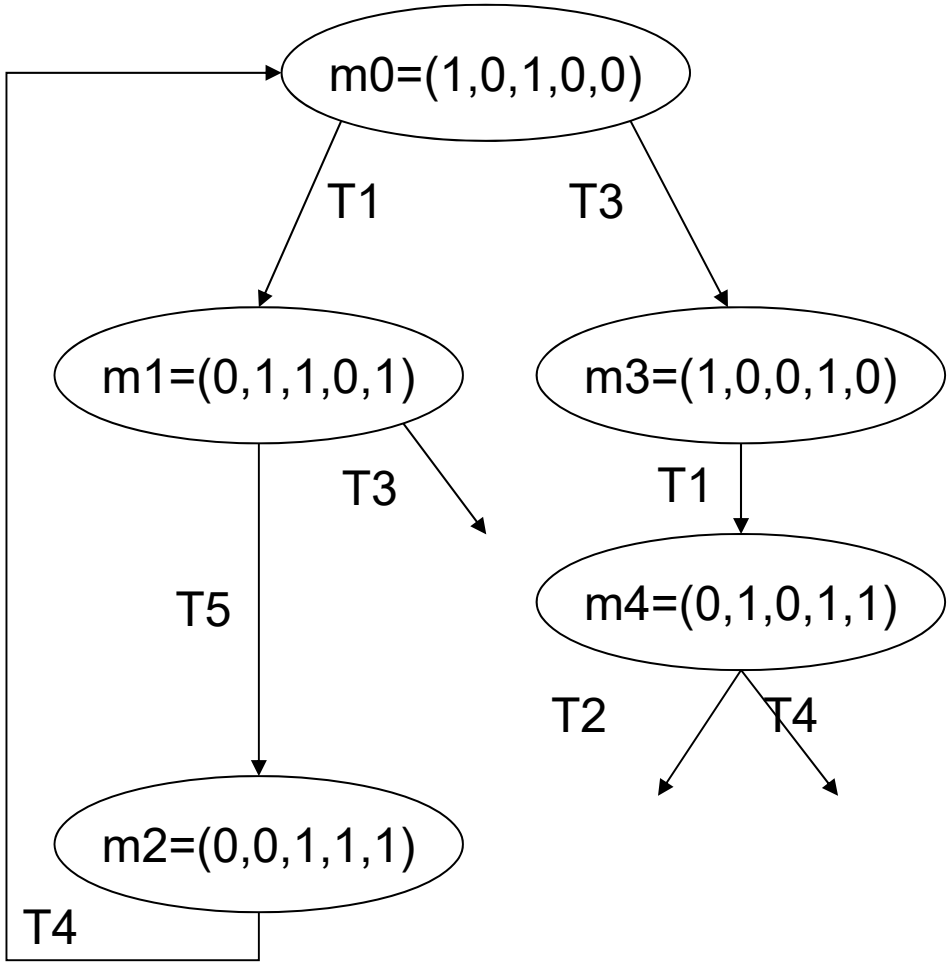


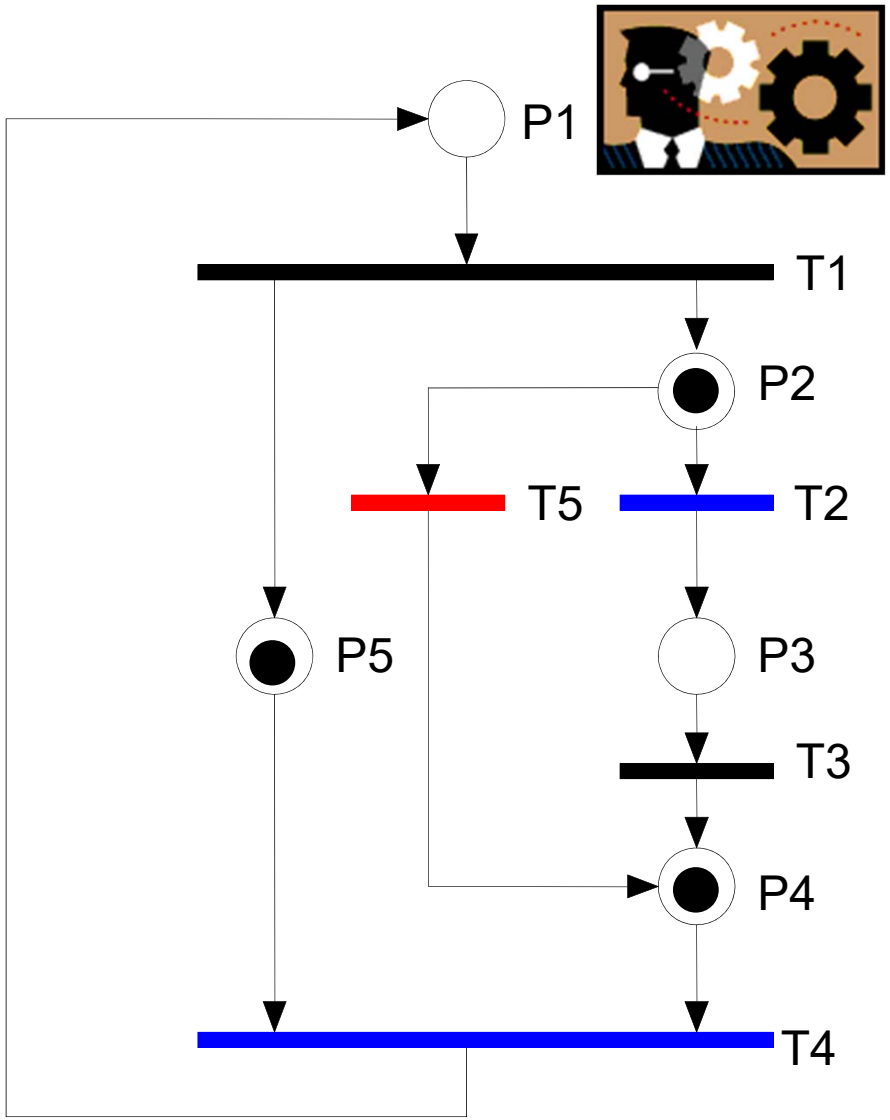
Erreichbarkeitsgraph und Erreichbarkeitsmenge zu Bsp. 2



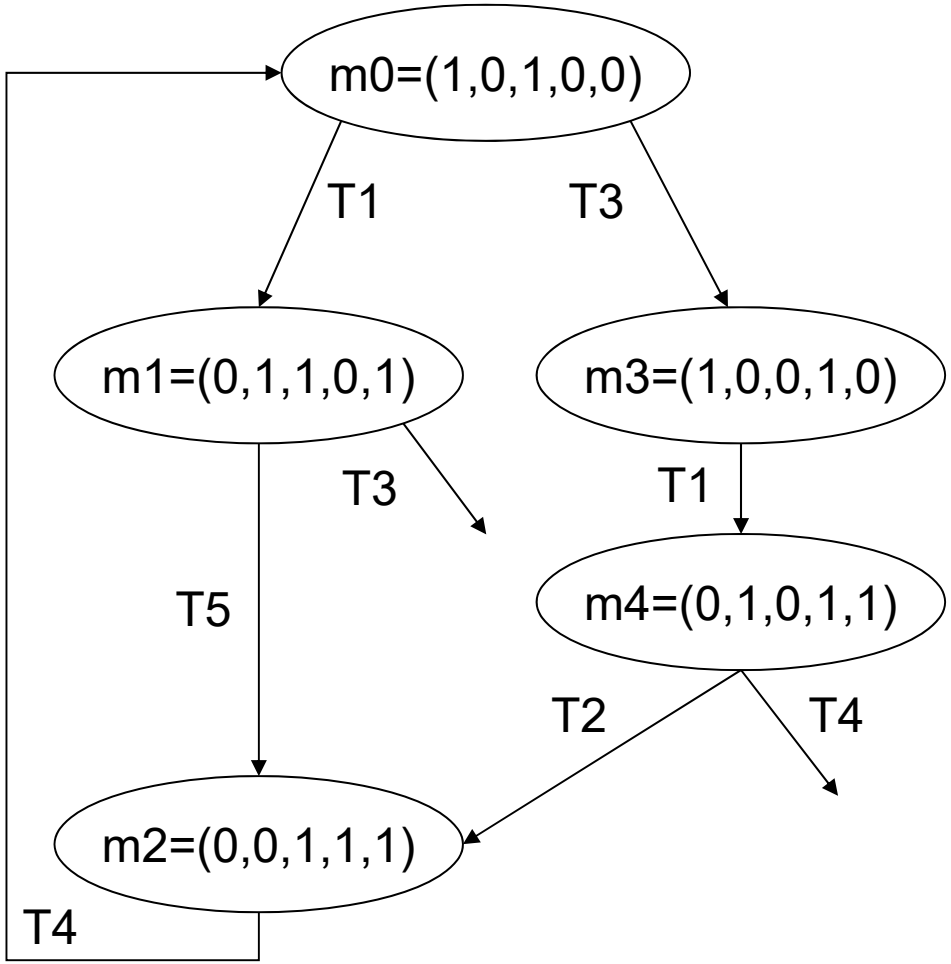


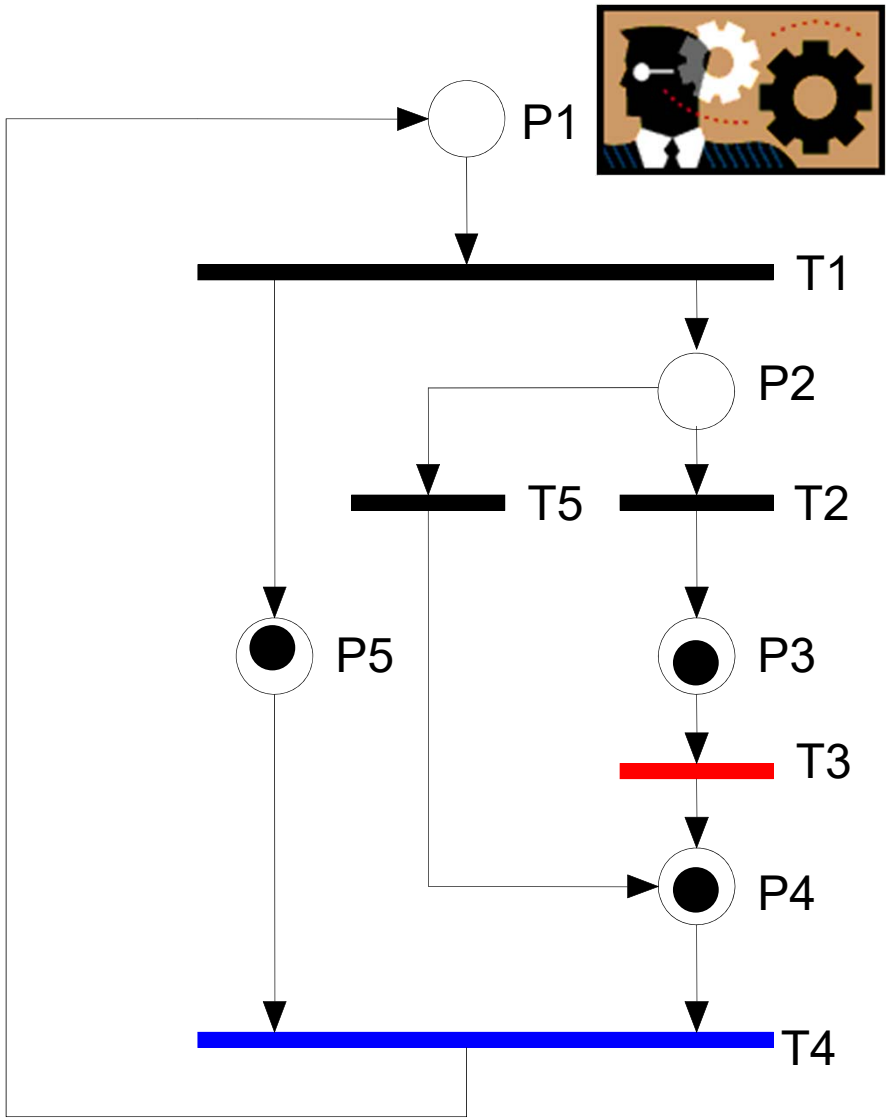
Erreichbarkeitsgraph und Erreichbarkeitsmenge zu Bsp. 2



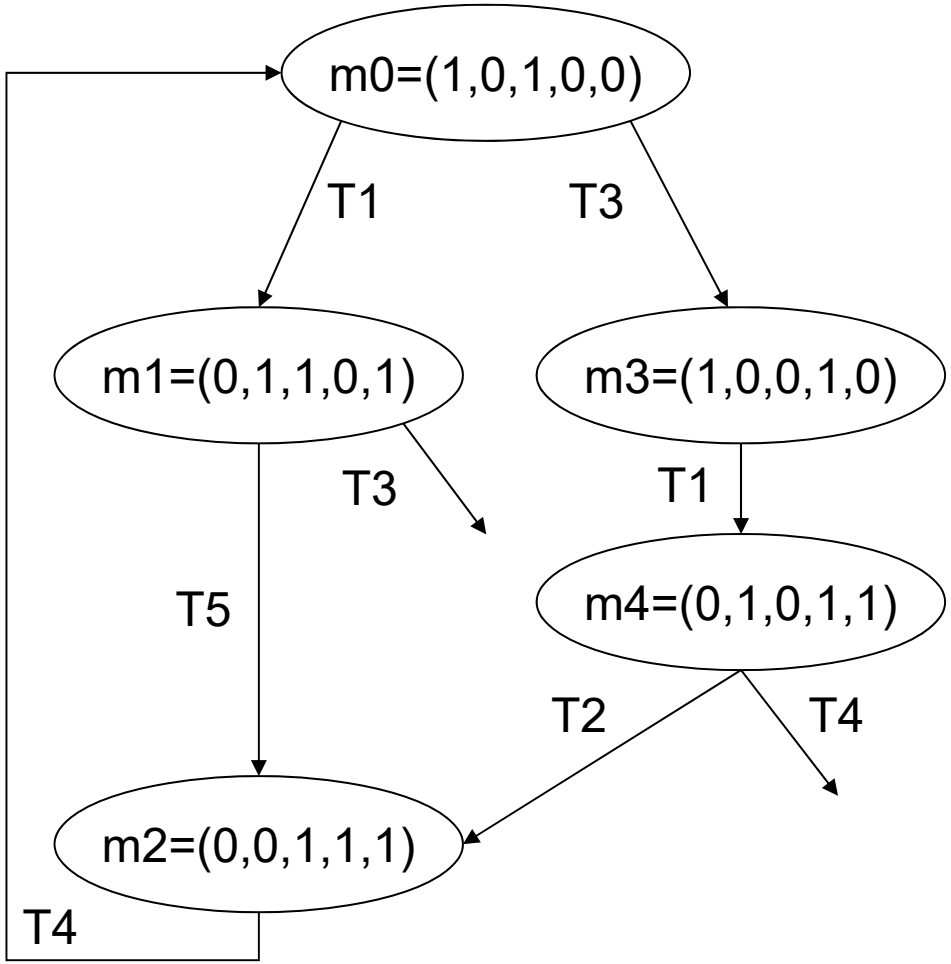


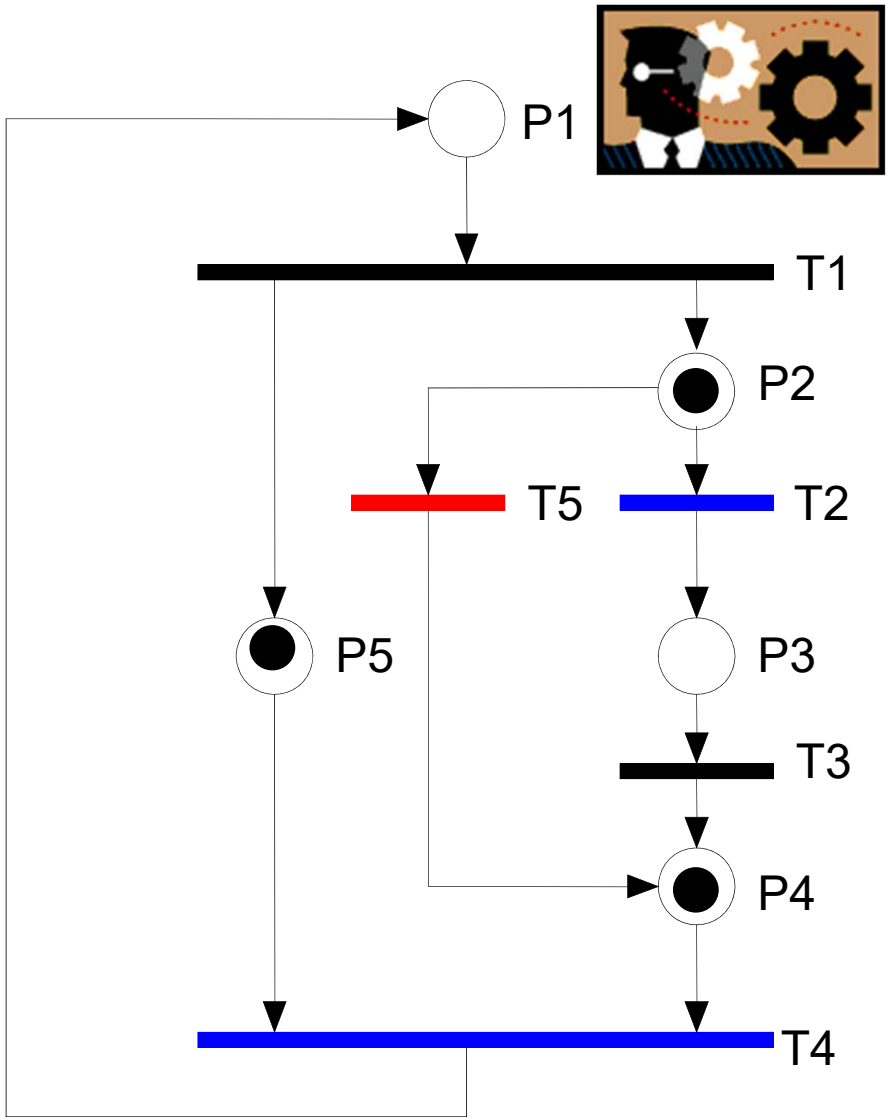
Erreichbarkeitsgraph und Erreichbarkeitsmenge zu Bsp. 2



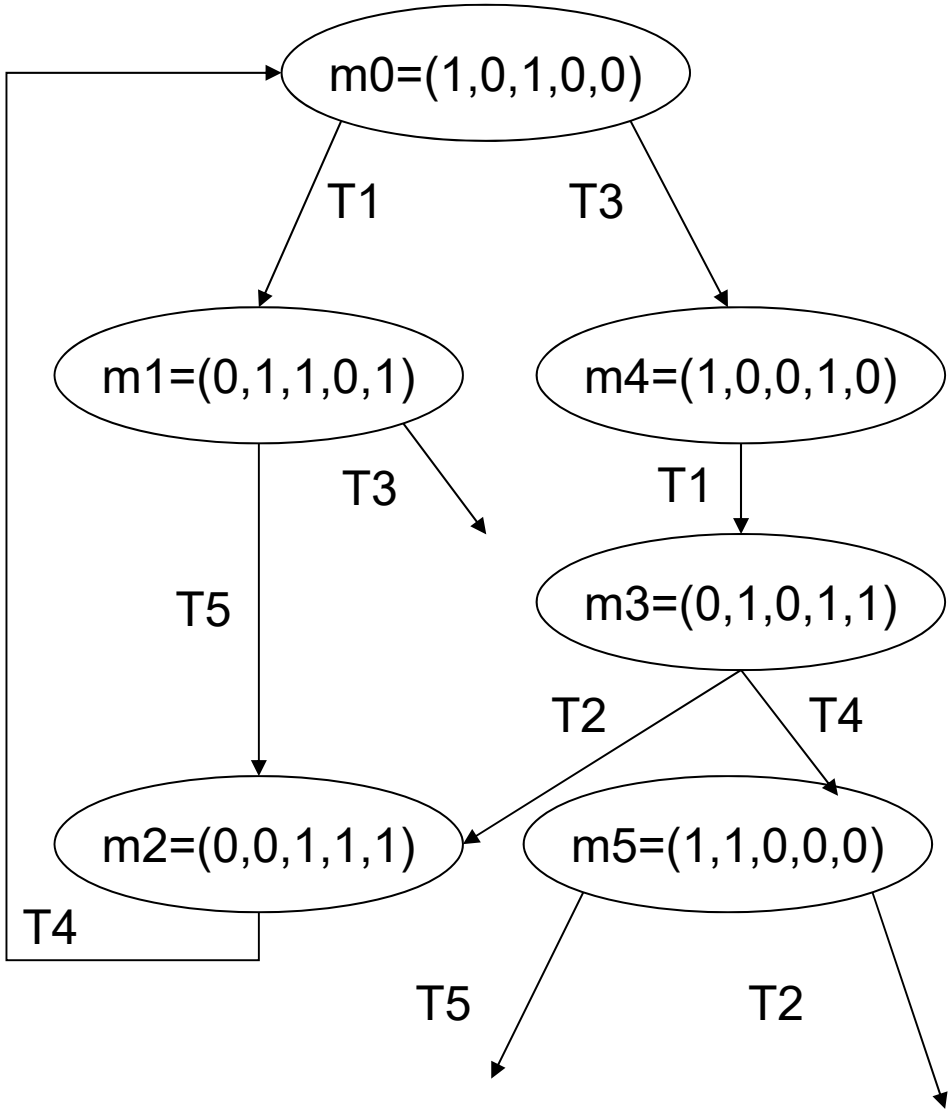


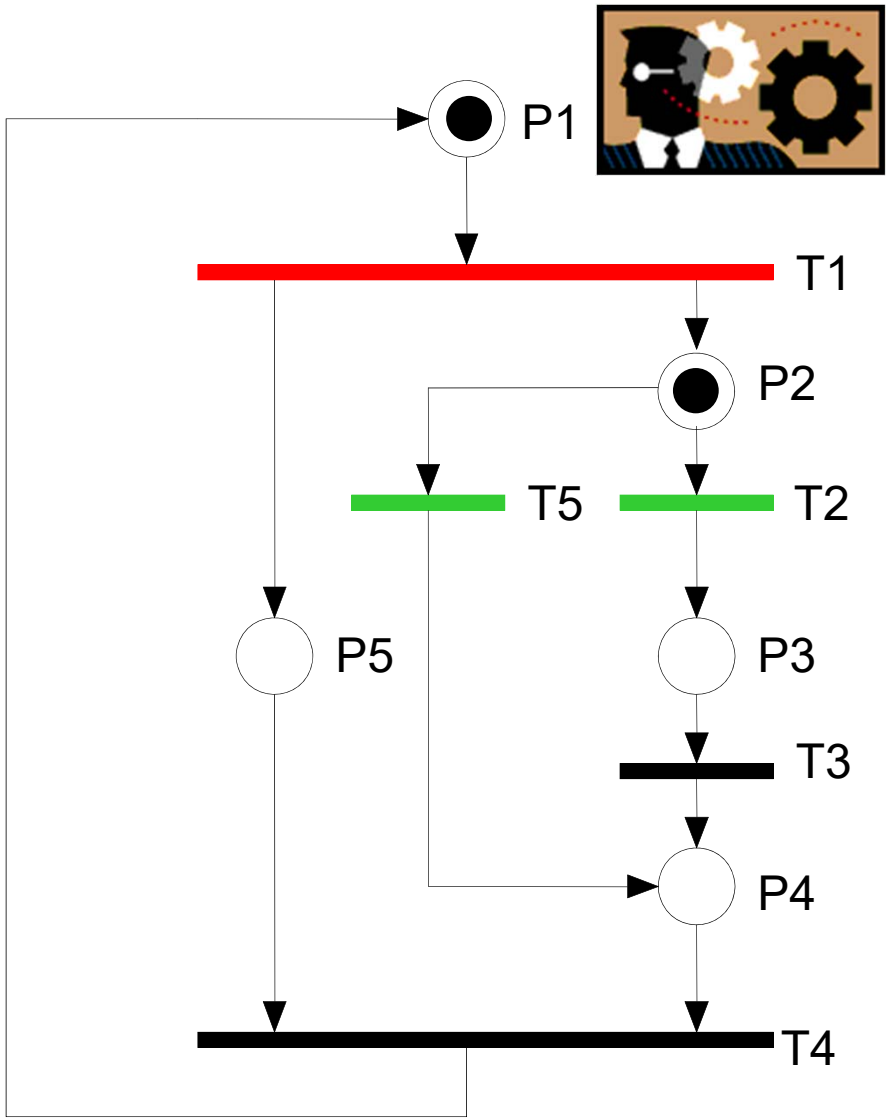
Erreichbarkeitsgraph und Erreichbarkeitsmenge zu Bsp. 2



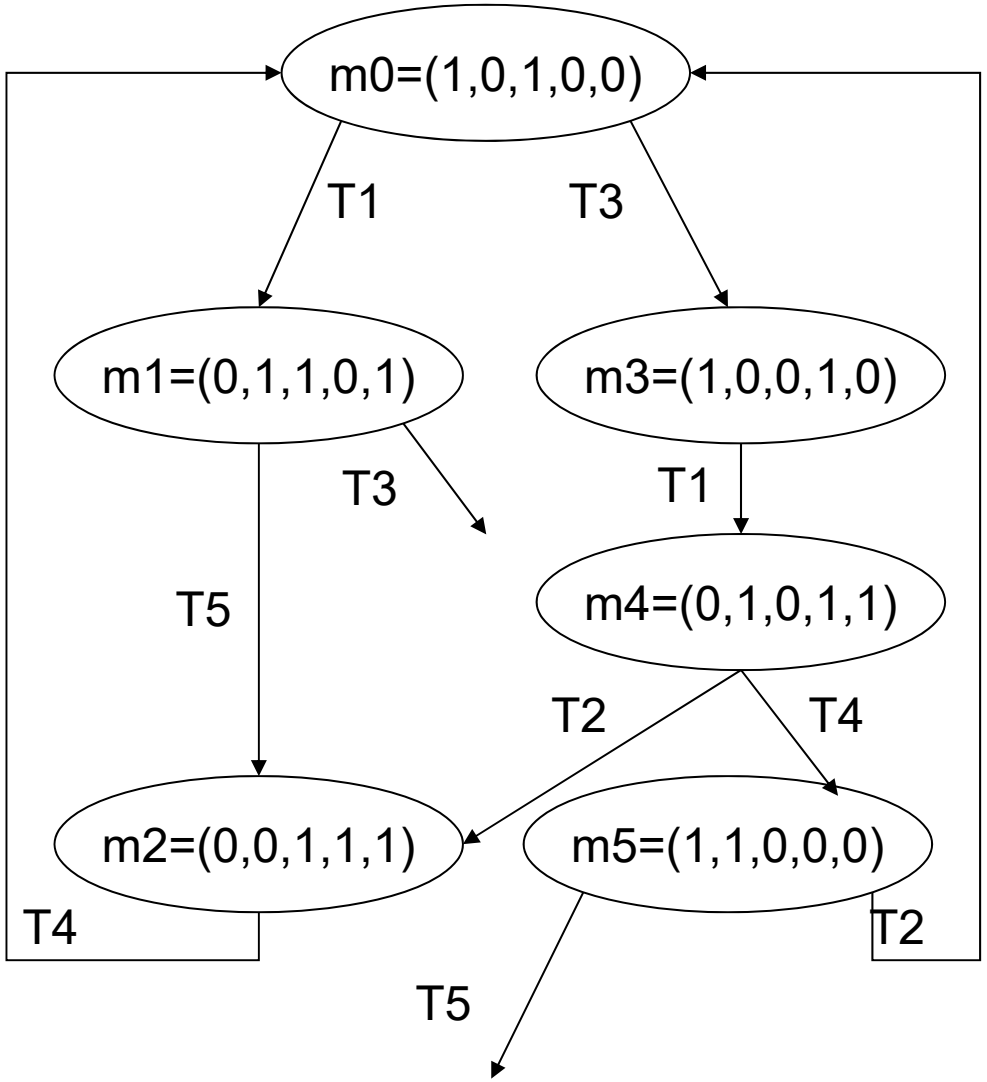


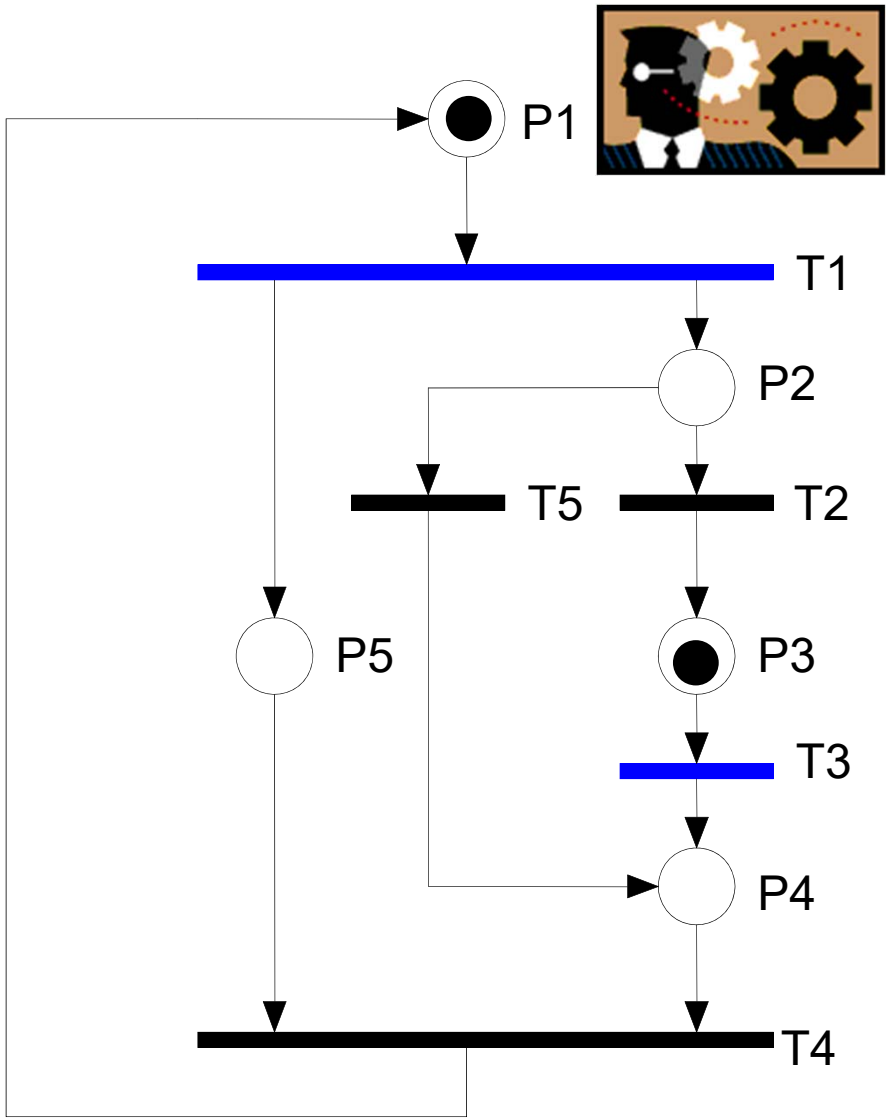
Erreichbarkeitsgraph und Erreichbarkeitsmenge zu Bsp. 2



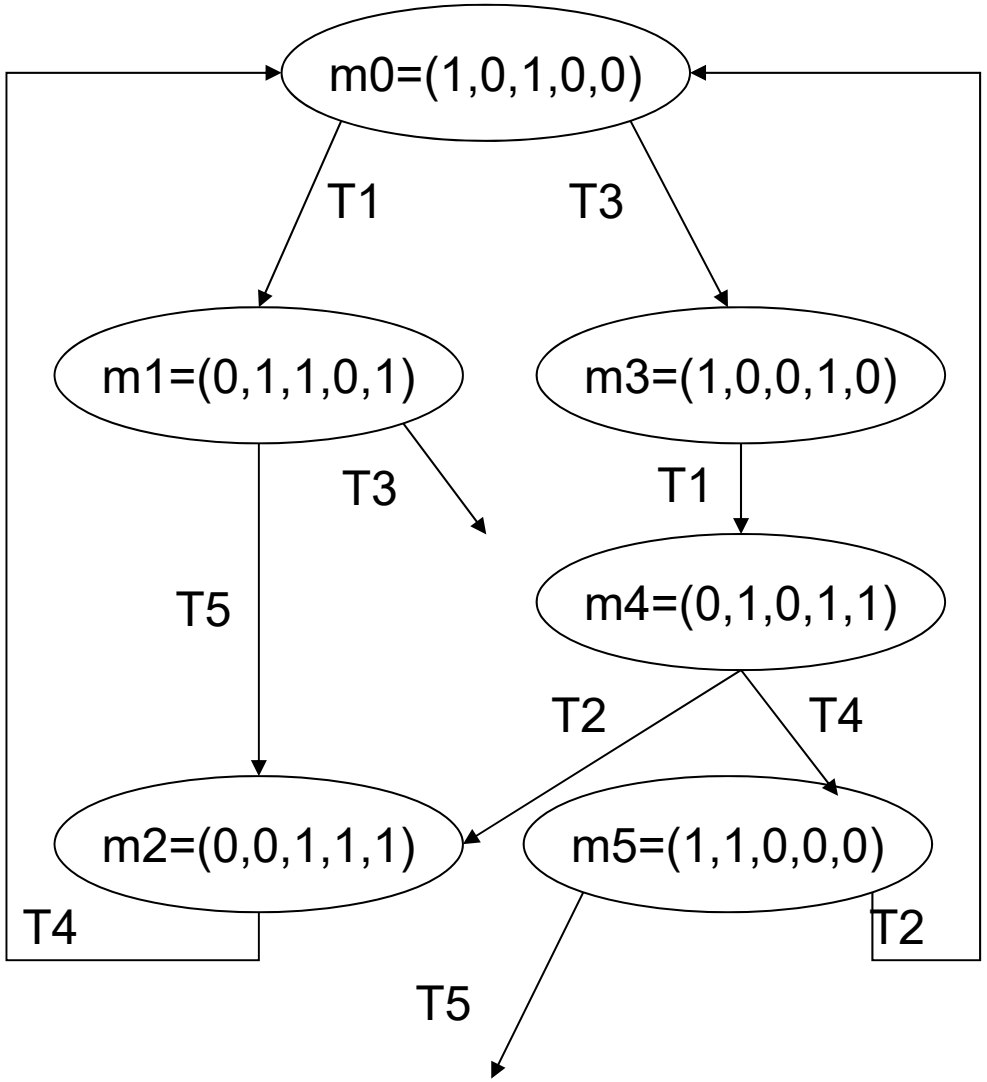


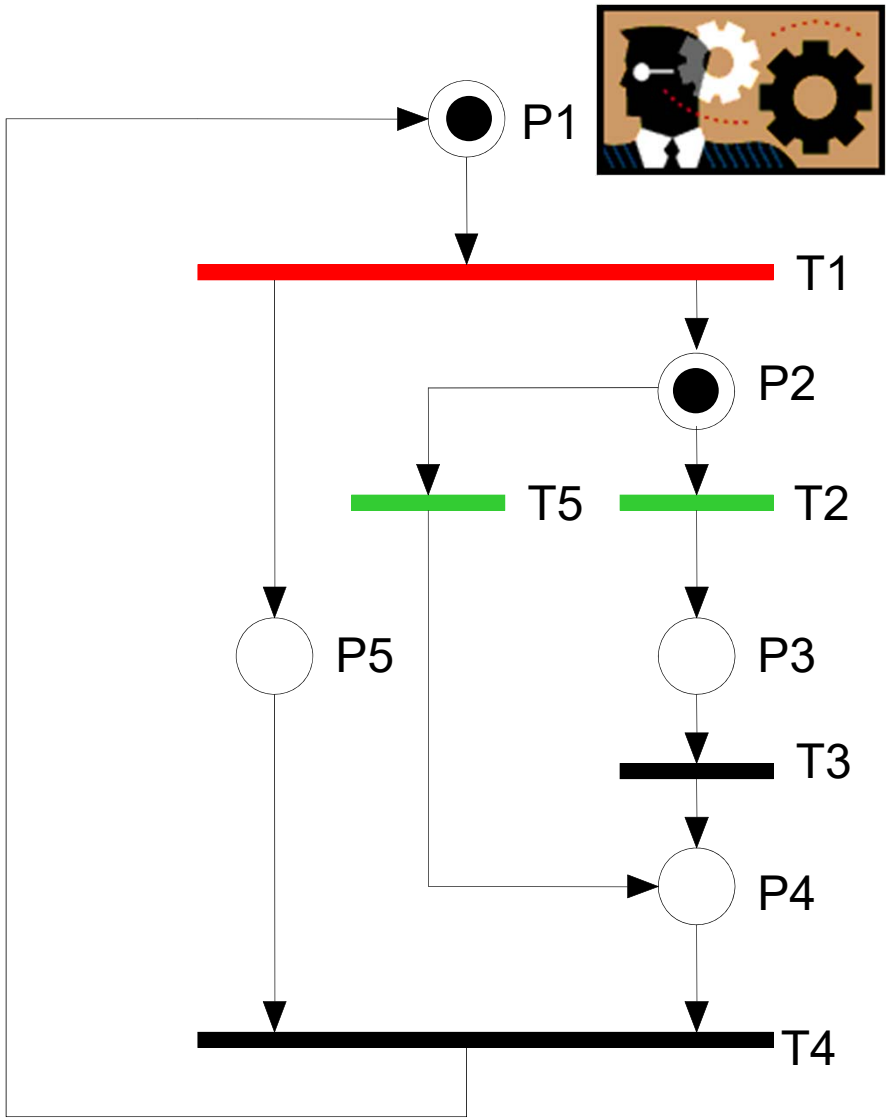
Erreichbarkeitsgraph und Erreichbarkeitsmenge zu Bsp. 2



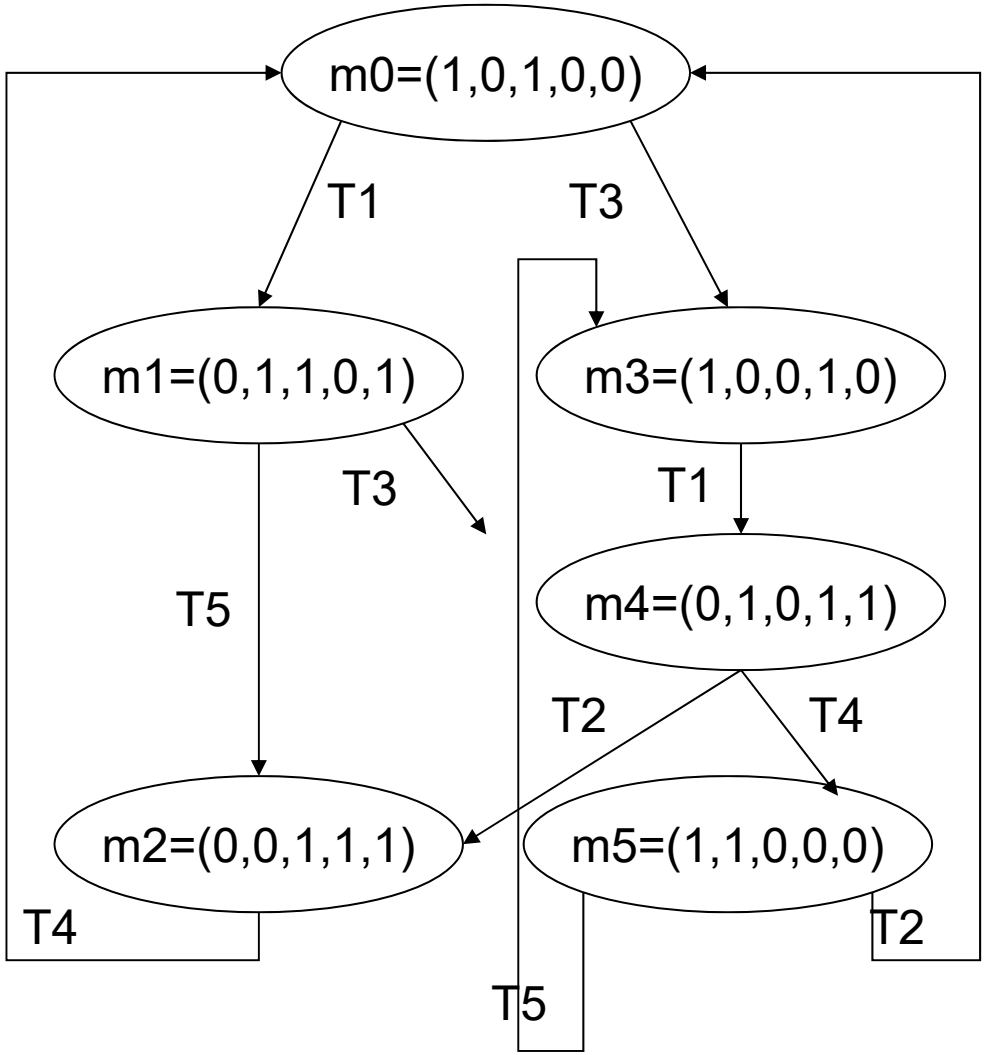


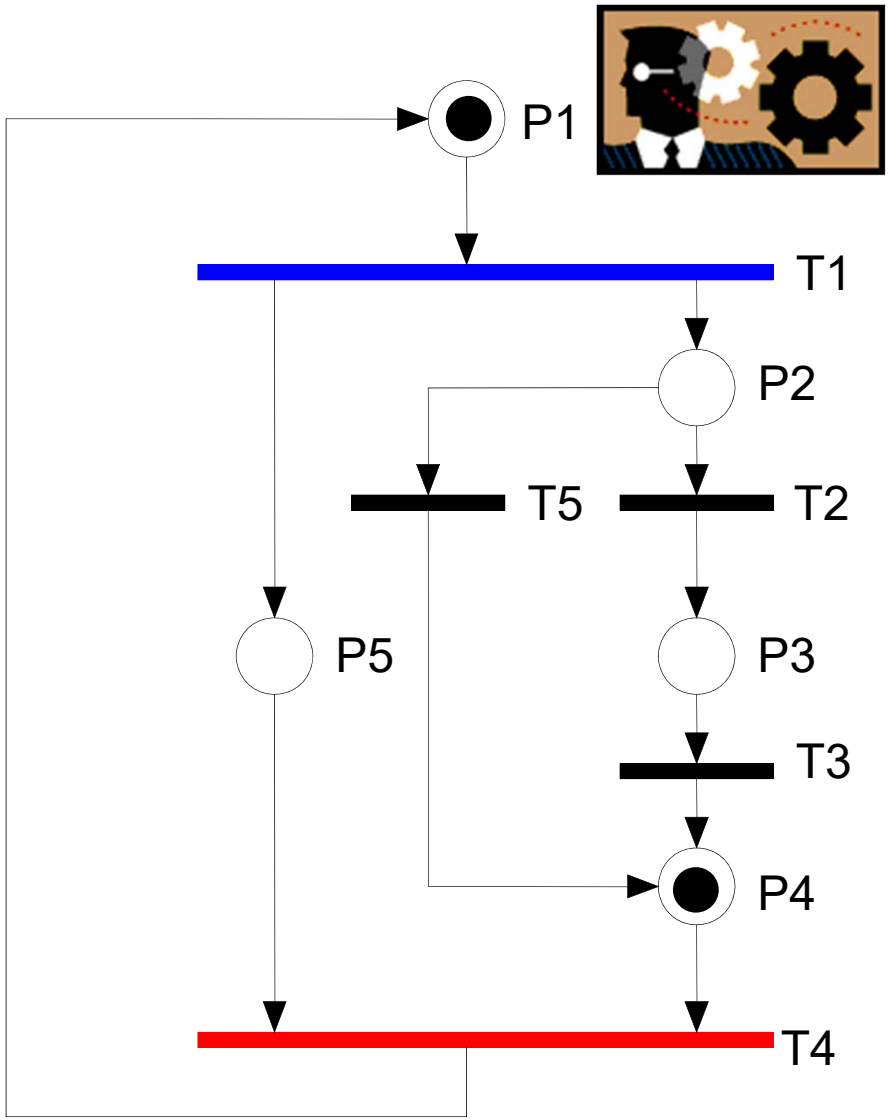
Erreichbarkeitsgraph und Erreichbarkeitsmenge zu Bsp. 2



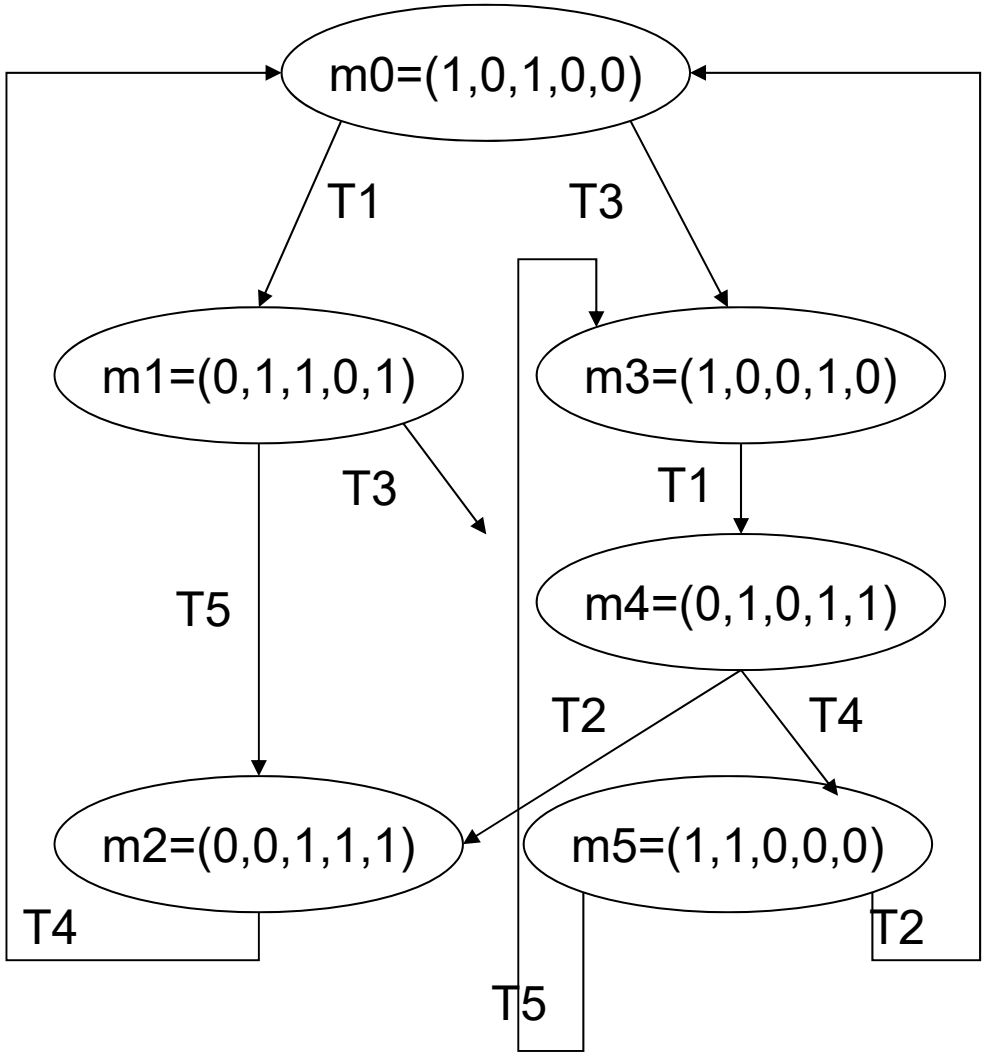


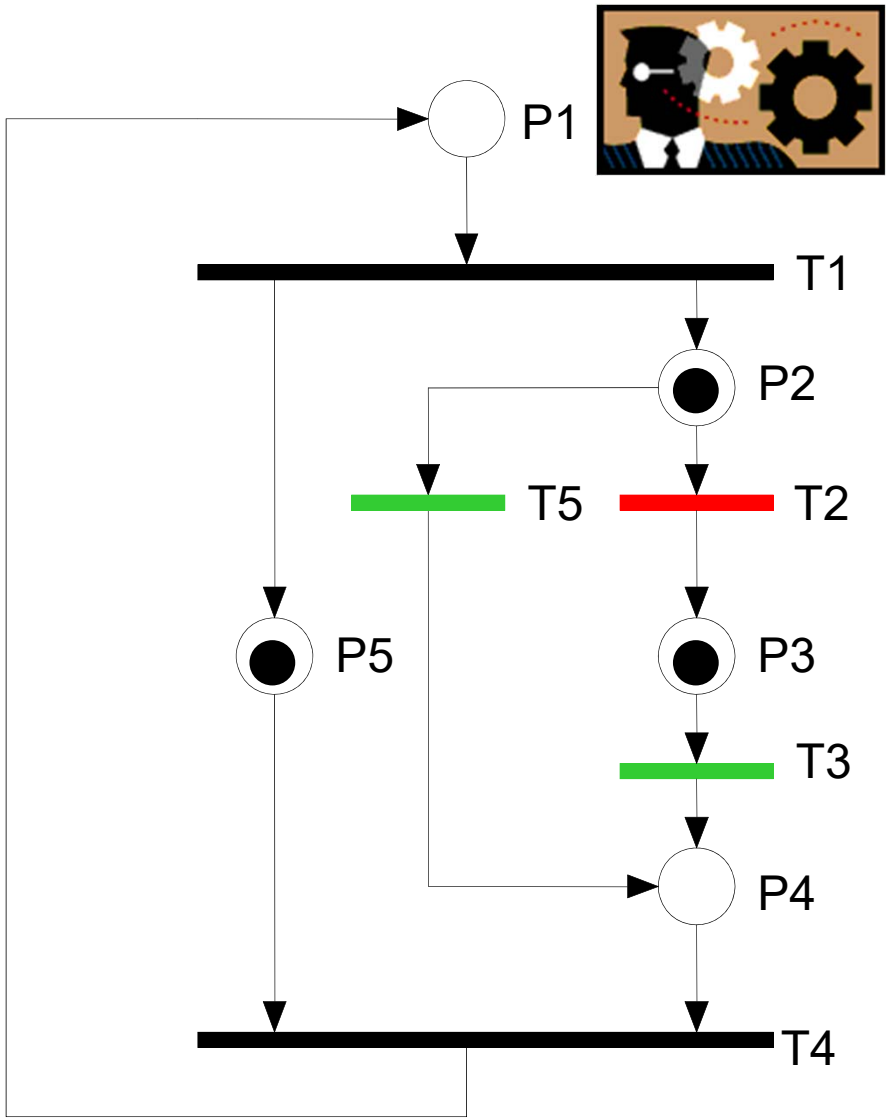
Erreichbarkeitsgraph und Erreichbarkeitsmenge zu Bsp. 2



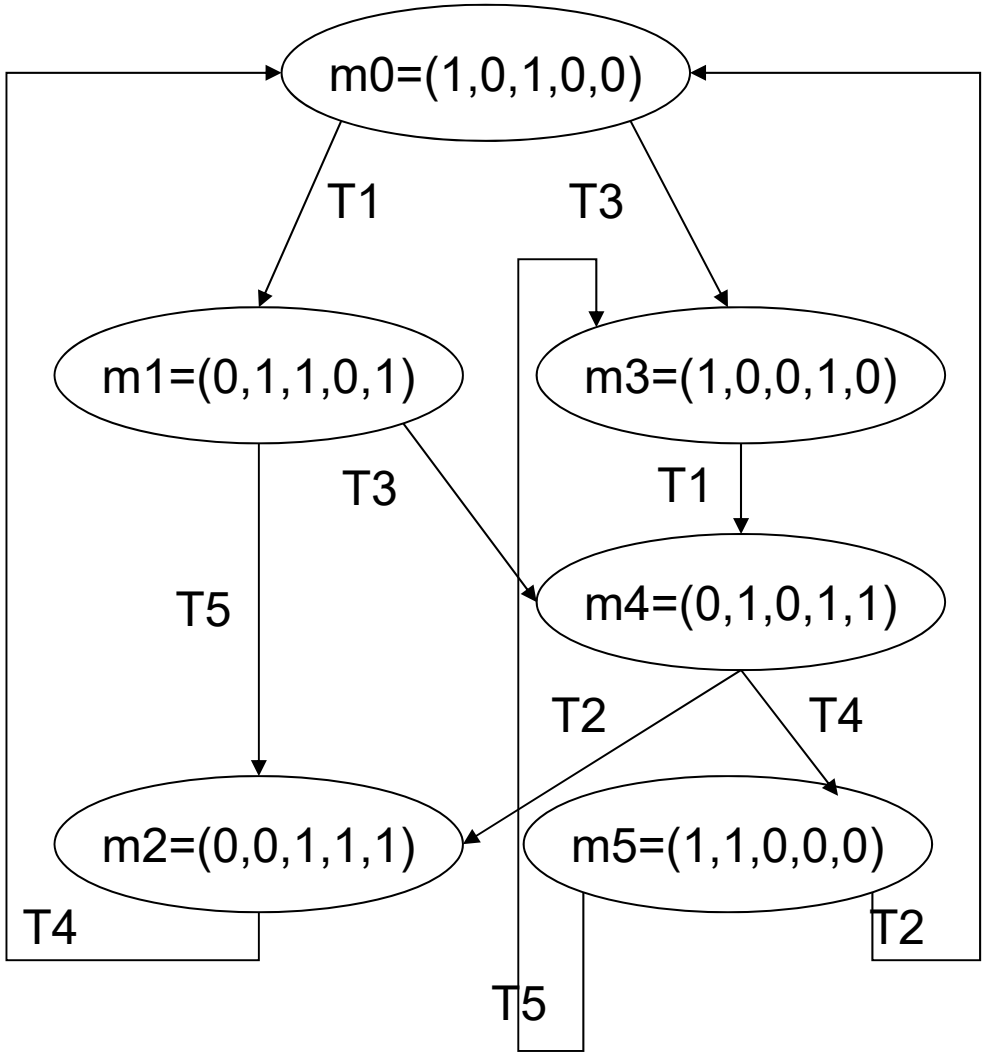


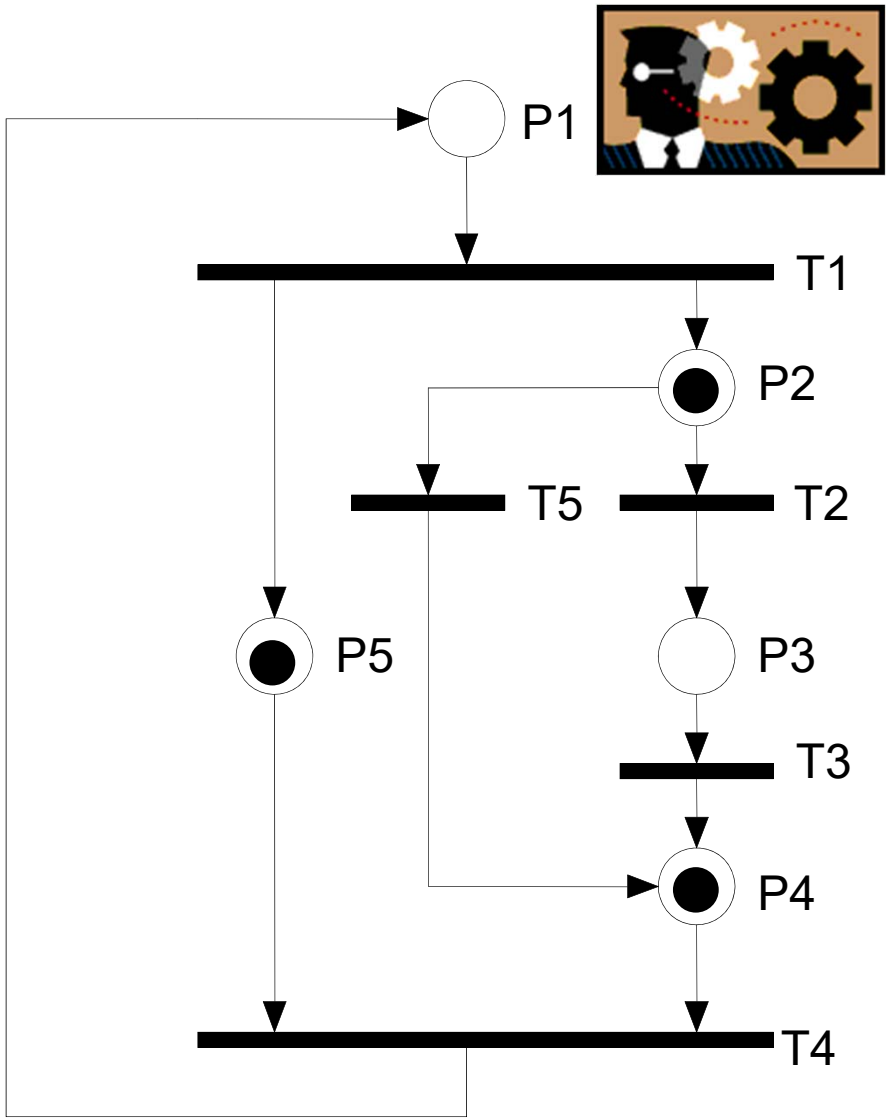
Erreichbarkeitsgraph und Erreichbarkeitsmenge zu Bsp. 2



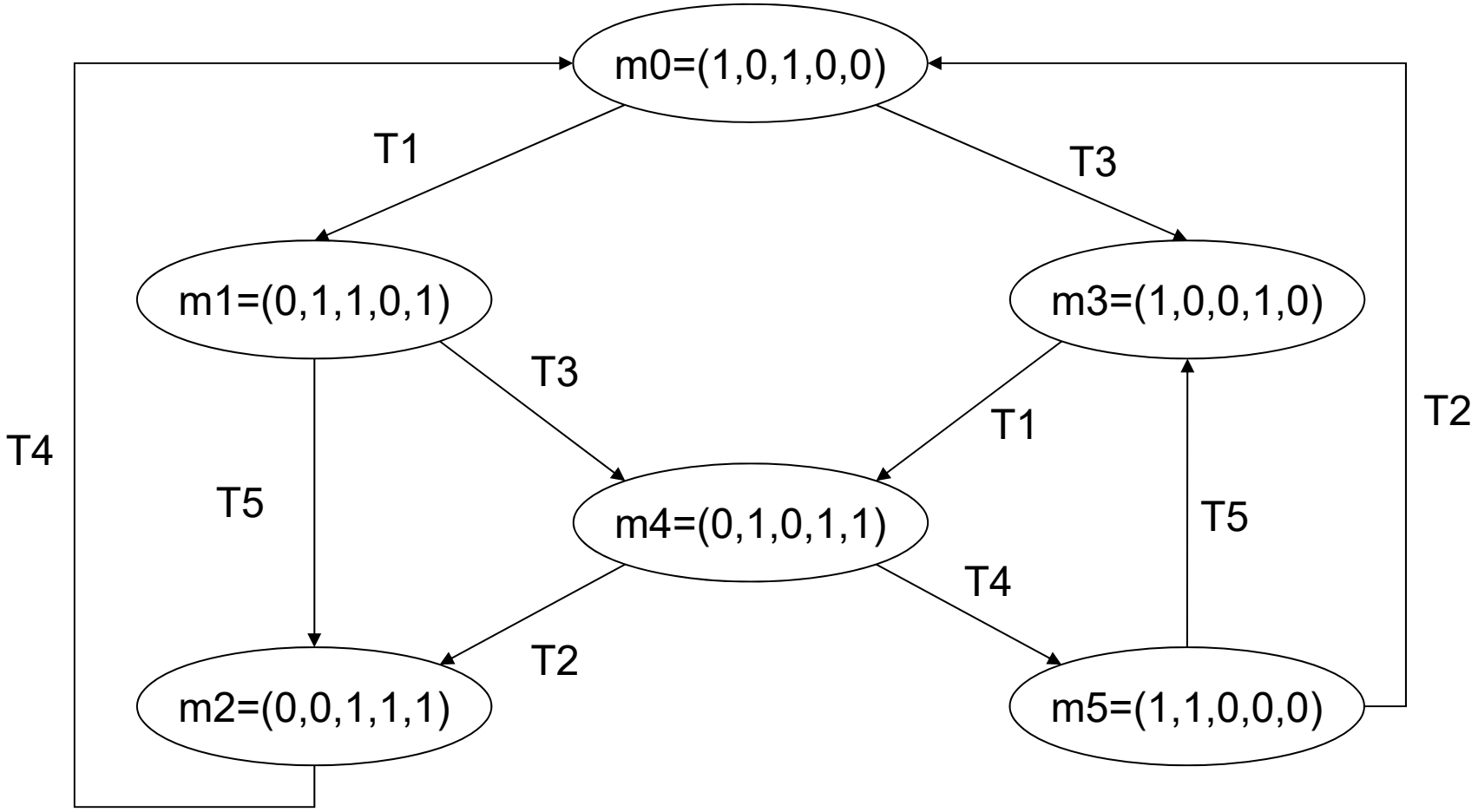


Erreichbarkeitsgraph und Erreichbarkeitsmenge zu Bsp. 2





Erreichbarkeitsgraph und Erreichbarkeitsmenge zu Bsp. 2



Ergebnis der Simulation der beiden Beispiele

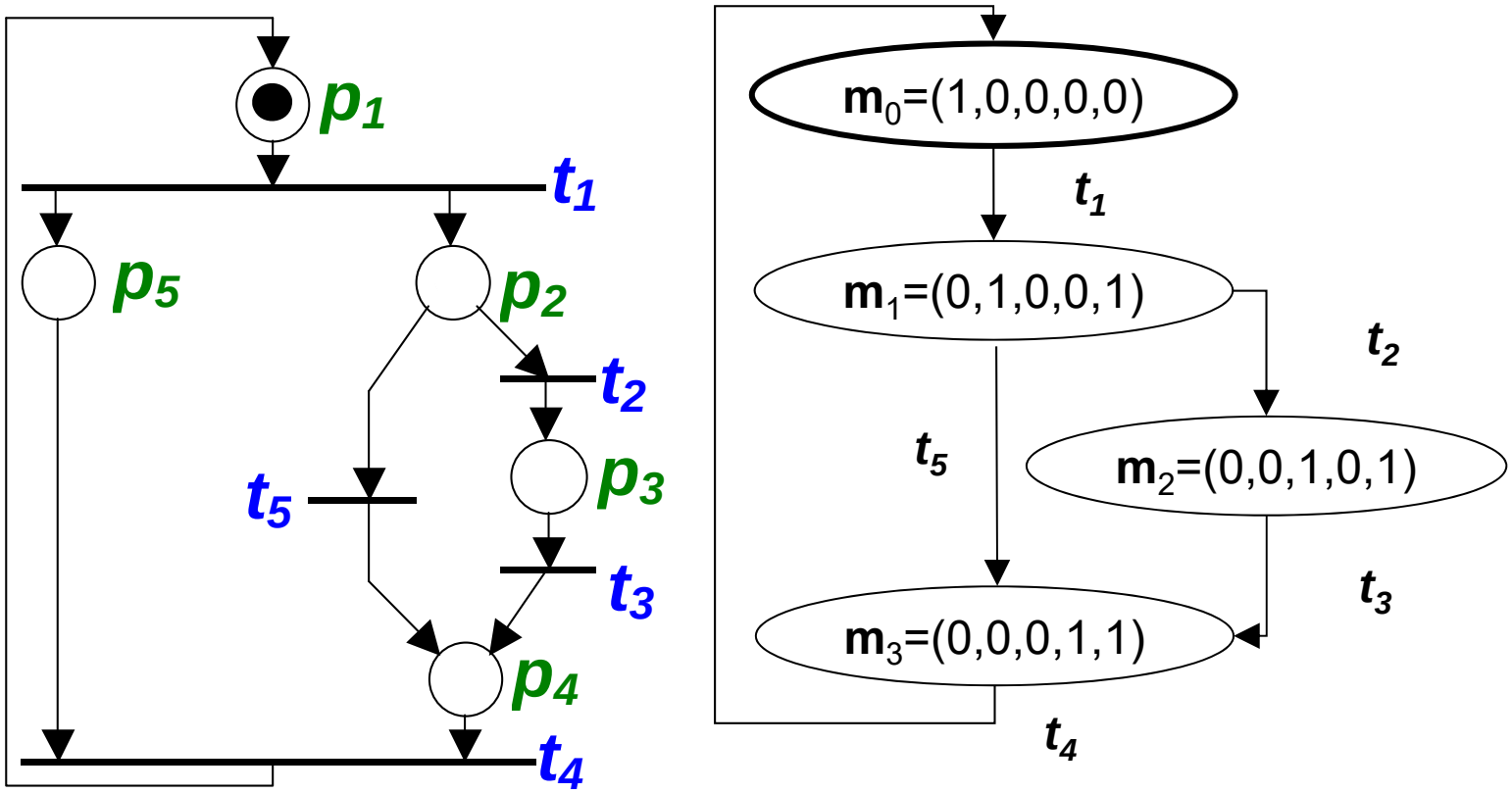
- Strukturell gleiche Netze mit unterschiedlicher Anfangsmarkierung ergeben unterschiedliches Verhalten:
  ➔ Anfangsmarkierung ist wichtiger Teil der Netzdefinition.
- Durch das Schalten von Transitionen werden neue Markierungen erzeugt.
- Der Zustand eines Petrinetzes ist durch seine Markierung gegeben (verteilt).
- Gängige verbale Interpretation:
  - Die Markierung in einem Platz beschreibt eine Situation.
  - Durch das Schalten einer Transition können mehrere Situationen in mehrere neue Situationen überführt werden.
  - Der Zustand ist die Menge der aktiven Situationen.
- Menge der möglichen Zustände eines Petrinetzes ist nicht direkt erkennbar.

➔ Frage der Erreichbarkeit

- Das Schalten einer konzessionierten Transition *t* unter einer Markierung **m** in einem PN erzeugt eine neue Markierung **m'**.
- Man sagt: **m**' ist *direkt erreichbar* vom **m** (oder auch Folgemarkierung).
- Wenn unter **m'** eine Transition *t'* schalten kann und die Markierung **m''** erzeugt, dann heißt **m''** von **m** erreichbar über die Schaltsequenz <*t*, *t'*>.
- Allgemein heißt eine Markierung $\mathbf{m}_2$ *erreichbar* von einer Markierung $\mathbf{m}_1$ wenn eine Schaltsequenz existiert, die ausgehend von $\mathbf{m}_1$ zu $\mathbf{m}_2$ führt.
- Die Erreichbarkeitsmenge (reachability set) ***EM*** eines Petrinetzes PN = (***P***, ***T***, ***F***, $\mathbf{m}_0$) besteht aus $\mathbf{m}_0$ und allen Markierungen die von $\mathbf{m}_0$ aus erreichbar sind.
- Der Erreichbarkeitsgraph (reachability graph) EG ist ein Graph EG = (***EM***, $\boldsymbol{E} \subseteq \boldsymbol{EM} \times \boldsymbol{EM} \rightarrow \boldsymbol{T}$) mit den erreichbaren Markierungen als Knoten. Eine Kante $e \in \boldsymbol{E}$ mit $e = (\mathbf{m}_i, \mathbf{m}_k) \rightarrow t_S$ gibt an, dass die Markierung $\mathbf{m}_k$ durch Schalten von t_S von $\mathbf{m}_i$ direkt erreichbar ist. Die Kanten des Graphen werden mit der entsprechenden Transition beschriftet, die Knoten mit den Markierungen.

→ EG eines PN ist ein Halbautomat

Beispiel: Erreichbarkeitsgraph



$\boldsymbol{ES}_{\mathrm{PN}} = \{\mathbf{m}_0, \mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3\}$

$\mathrm{EG}_{\mathrm{PN}} = (\boldsymbol{RS}_{\mathrm{PN}}, \{(\mathbf{m}_0,\mathbf{m}_1), (\mathbf{m}_1,\mathbf{m}_2), (\mathbf{m}_1,\mathbf{m}_3), (\mathbf{m}_2,\mathbf{m}_3), (\mathbf{m}_3,\mathbf{m}_0)\})$

Zusammenfassung

- Petrinetze erlauben die Abbildung von nebenläufigem Verhalten
- Problem der Erreichbarkeit

Systemmodellierung
Teil 1: Ereignisdiskrete Systeme

Kapitel 4
Ereignisdiskrete Systeme (V)

Petrinetze ctnd.

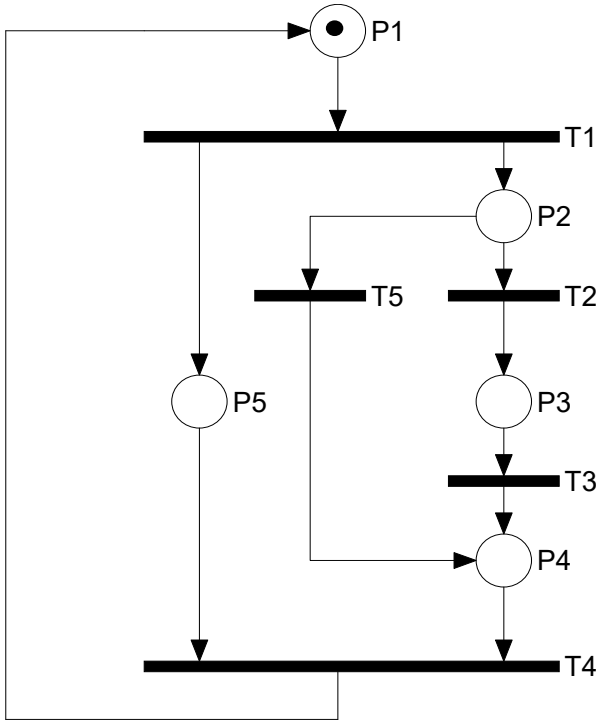
UNIVERSITÄT DES SAARLANDES

AES
Automatisierungs- und Energiesysteme

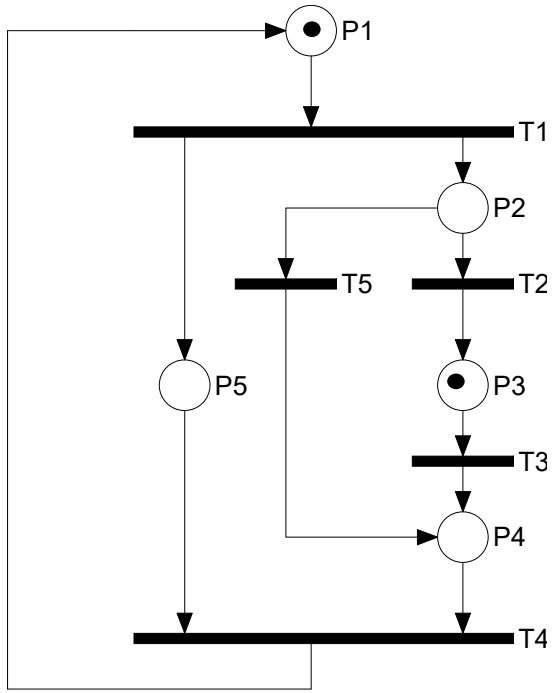
Eigenschaften von Petrinetzen (BE-Netze)

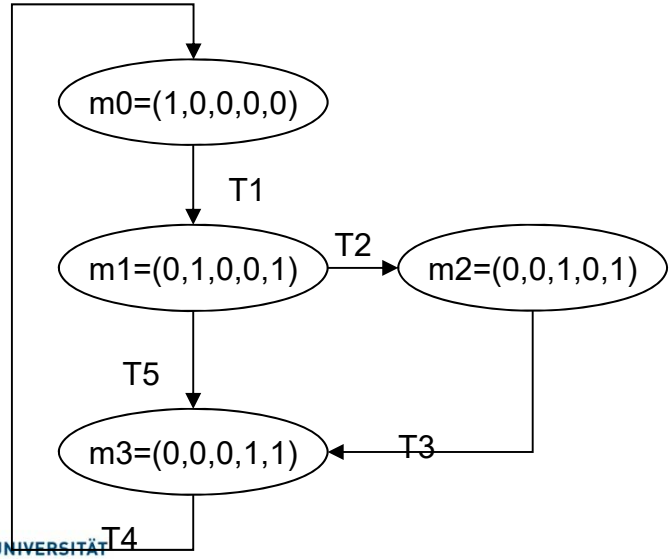
- Konflikt
- Kontakt
- Livelock, Deadlock
- Lebendigkeit
- Reversibilität

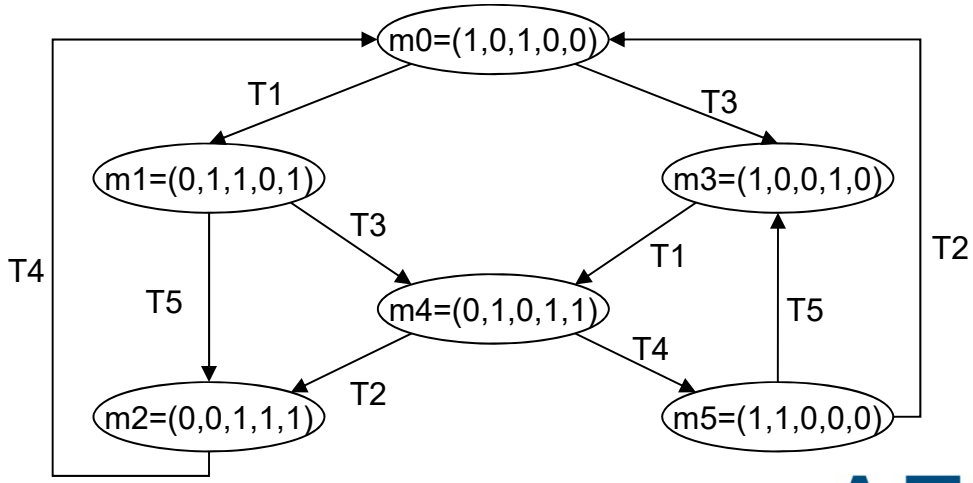
1


P1
T1
P2
T5
T2
P5
P3
T3
P4
T4

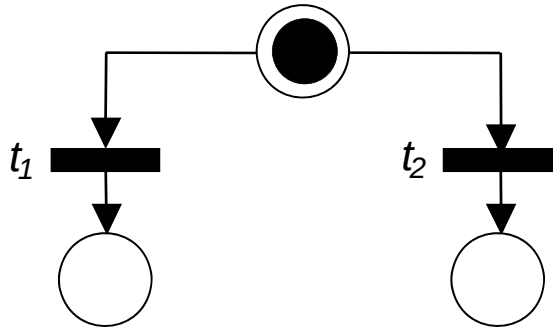
2


P1
T1
P2
T5
T2
P5
P3
T3
P4
T4

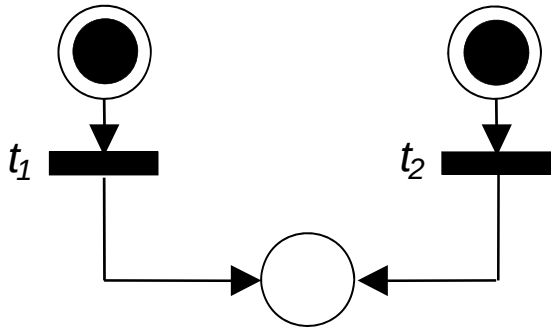

m0=(1,0,0,0,0)
T1
m1=(0,1,0,0,1)
T2
m2=(0,0,1,0,1)
T5
m3=(0,0,0,1,1)
T3
T4


m0=(1,0,1,0,0)
T1
T3
m1=(0,1,1,0,1)
m3=(1,0,0,1,0)
T3
T1
T2
T4
T5
m4=(0,1,0,1,1)
T5
T4
T2
m2=(0,0,1,1,1)
m5=(1,1,0,0,0)

- Zwei konzessionierte Transitionen befinden sich in einem **Konflikt**, wenn durch das Schalten der einen, der anderen die Konzession entzogen wird.
- Notwendige Bedingung: Die beiden Transitionen teilen sich zumindest einen Platz im Vorbereich oder im Nachbereich:



Vorwärtskonflikt



Rückwärtskonflikt

- Ein Konflikt ist eine dynamische Eigenschaft.
- Ein PN heißt **konfliktfrei** wenn unter keiner seiner erreichbaren Markierungen ein Konflikt entsteht.
- Konflikte werden benutzt um Entscheidungen zu modellieren.
- Das Verhalten eines PN mit Konflikten ist nicht deterministisch.

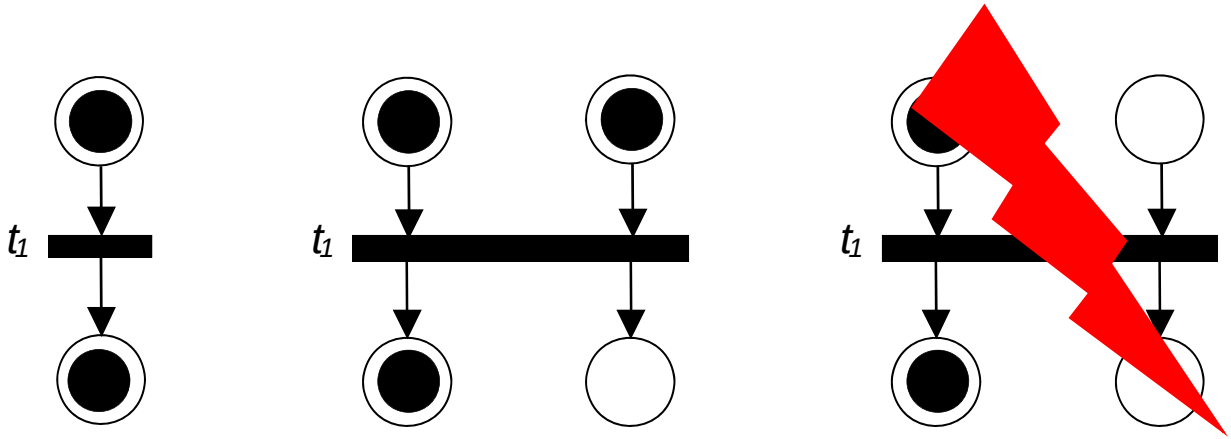
Konflikte in den beiden Beispielnetzen

SM 1

- Netz 1

- Netz 2

- Ein **Kontakt** liegt vor, wenn eine Transition nur deshalb nicht schalten kann, weil Plätze in ihrem Nachbereich belegt sind.



- Ein Kontakt ist eine dynamische Eigenschaft.
- Ein PN heißt **kontaktfrei** wenn unter keiner seiner erreichbaren Markierungen ein Kontakt entsteht.
- Kontakte sind in der Interpretation mit Bedingung und Ereignis unschön, da ein Ereignis nicht länger nur von der Erfüllung seiner Vorbedingungen sondern auch von der Nichterfüllung seiner Nachbedingungen abhängt.

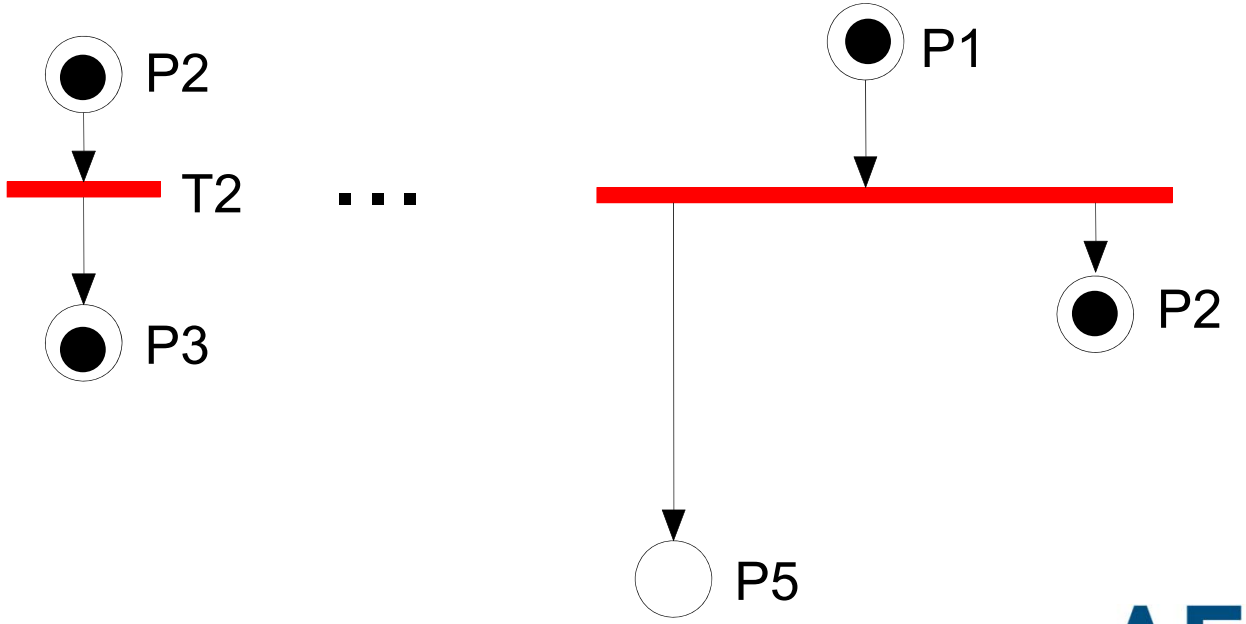
Kontakte in den beiden Beispielnetzen

- Netz 1

  KEINE KONTAKTE

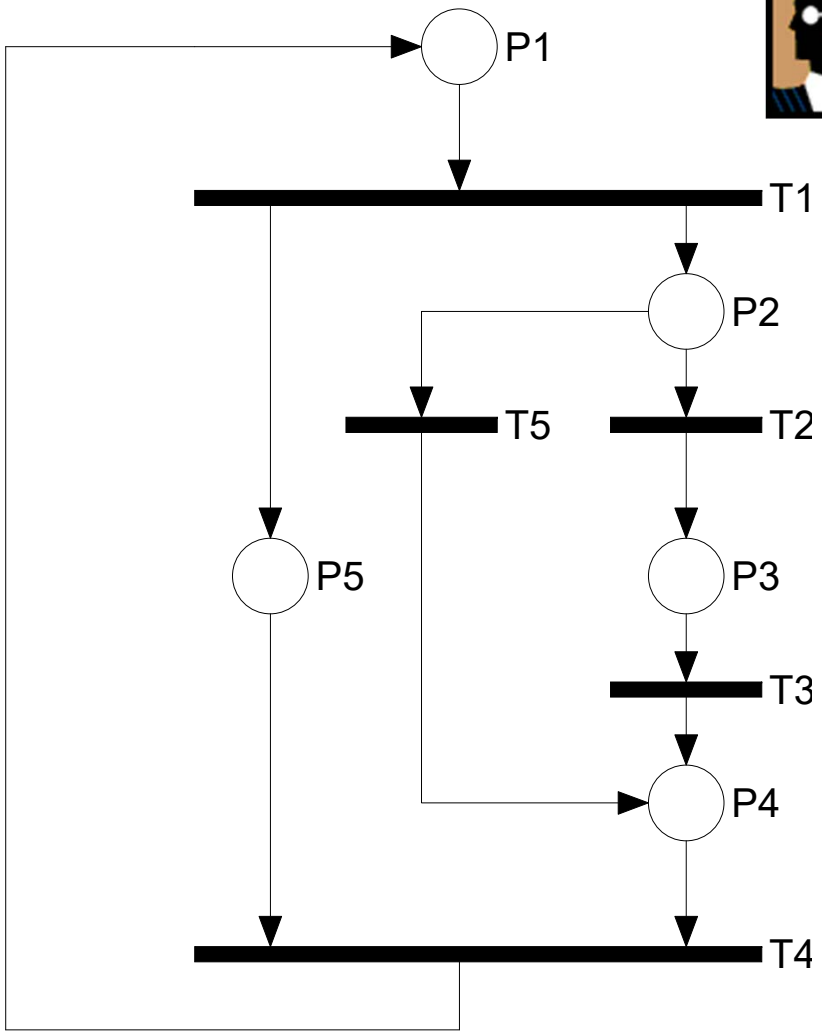
- Netz 2

  T2 unter m1
  T3 unter m2
  T5 unter m4
  T1 unter m5



Beseitigung von Kontakten durch Komplementär-Plätze

- Ein Platz p‘ ist ein Komplementär-Platz zu einem Platz p wenn folgendes gilt:
  - der Nachbereich von p‘ ist gleich dem Vorbereich von p
  - der Vorbereich von p‘ ist gleich dem Nachbereich von p
  - p‘ ist unter $\mathbf{m_0}$ genau dann markiert wenn p unter $\mathbf{m_0}$ nicht markiert ist
- Fügt man für jeden Platz, dessen Markierung einen Kontakt hervorruft einen Komplementär-Platz ein, dann wird das PN kontaktfrei.
- **Anwendung auf Beispiel 2**



- Eine Transition t in einem Petrinetz heißt **lebendig**, wenn sie immer wieder konzessioniert werden kann.
- Eine Transition heißt **tot** wenn sie nie konzessioniert wird.
- Achtung aus (t ist nicht tot) folgt nicht (t ist lebendig).
- Eine Markierung heißt **Deadlock** bzw. **totale Verklemmung** wenn unter dieser Markierung keine Transition schalten kann.
- Ein PN heißt **Deadlock-frei** bzw. **verklemmungsfrei** wenn keine der erreichbaren Markierungen ein Deadlock ist.
- Ein Petrinetz heißt **lebendig** wenn alle seine Transitionen lebendig sind.
- Ein lebendiges Netz ist Deadlock-frei, ABER die Umkehrung gilt nicht.
- Ein Petrinetz ist **reversibel**, wenn von jeder erreichbaren Markierung die Anfangsmarkierung wieder erreicht werden kann.
- Ein reversibles Netz ist Deadlock-frei, ABER die Umkehrung gilt nicht.
- Eine Menge von Markierungen heißt **Livelock** wenn nach Erreichen eines der Zustände, die Menge nicht mehr verlassen werden kann.

Livelock

- stark zusammenhängende Komponente im Erreichbarkeitsgraph ohne abgehende Kanten.

Deadlock

- Einzelner Knoten im Erreichbarkeitsgraph ohne abgehende Kanten.

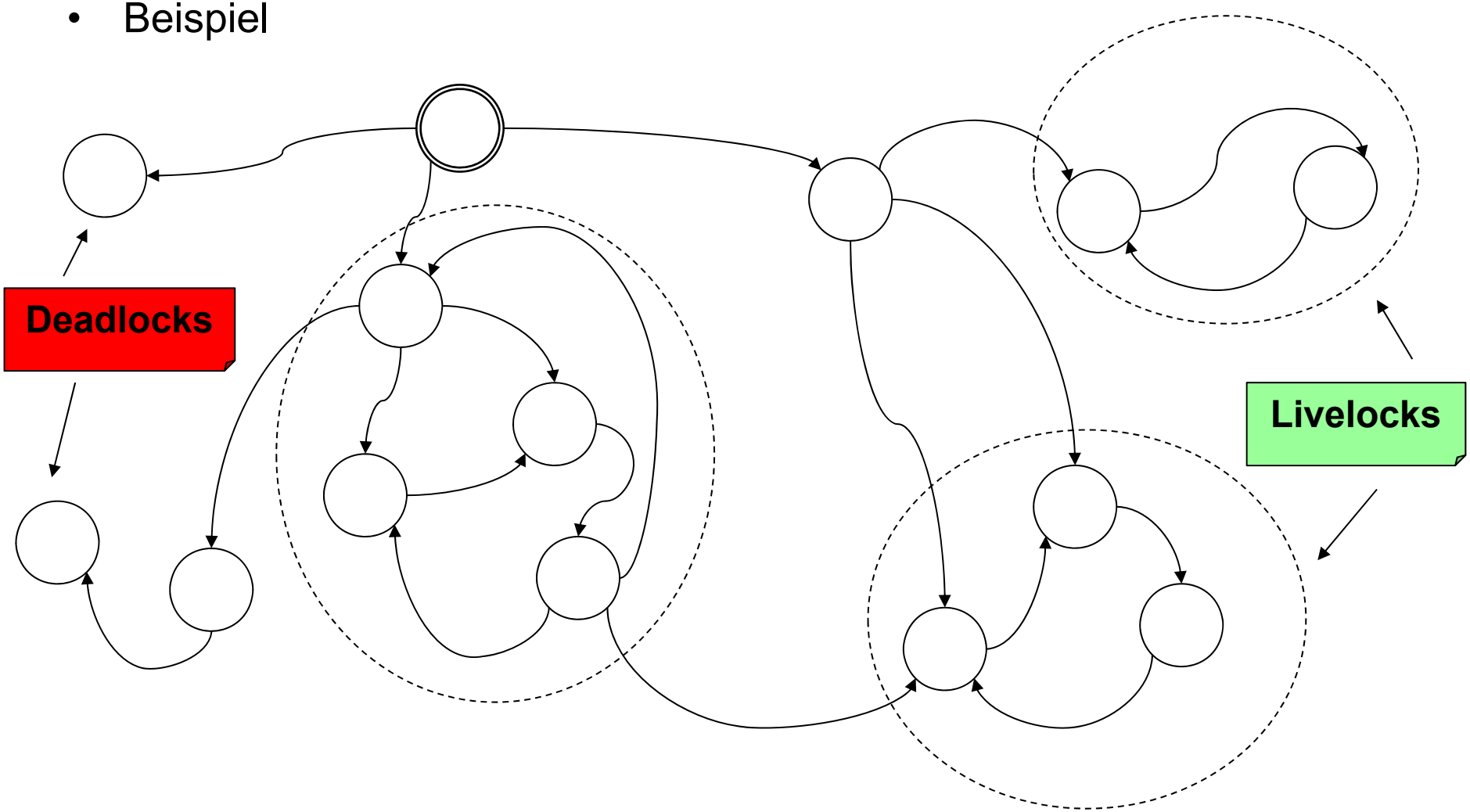
Lebendigkeit

- EG enthält keinen Deadlock,
- Alle Livelocks im EG enthalten alle Transitionen.

Reversibilität

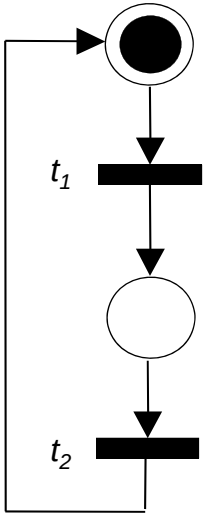
- EG enthält keinen Deadlock,
- EG enthält genau einen Livelock,
- $\mathbf{m_0}$ ist in diesem einen Livelock enthalten.

- Man fasst im Graphen alle die Knoten, die stark zusammenhängend sind zu einem Meta-Knoten zusammen
- Beispiel



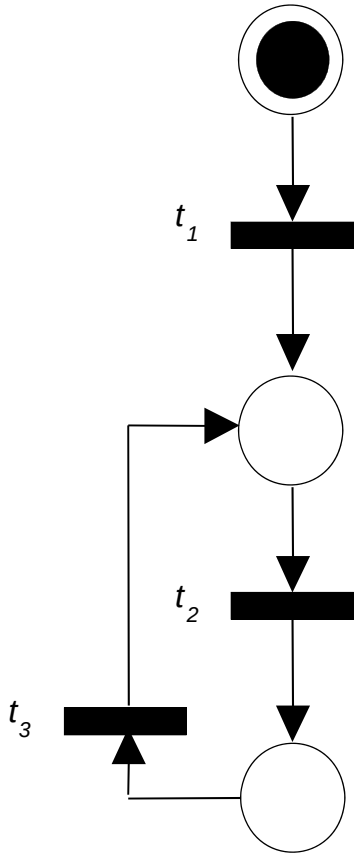
Unabhängigkeit von Lebendigkeit und Reversibilität

- Beispiel: lebendig und reversibel

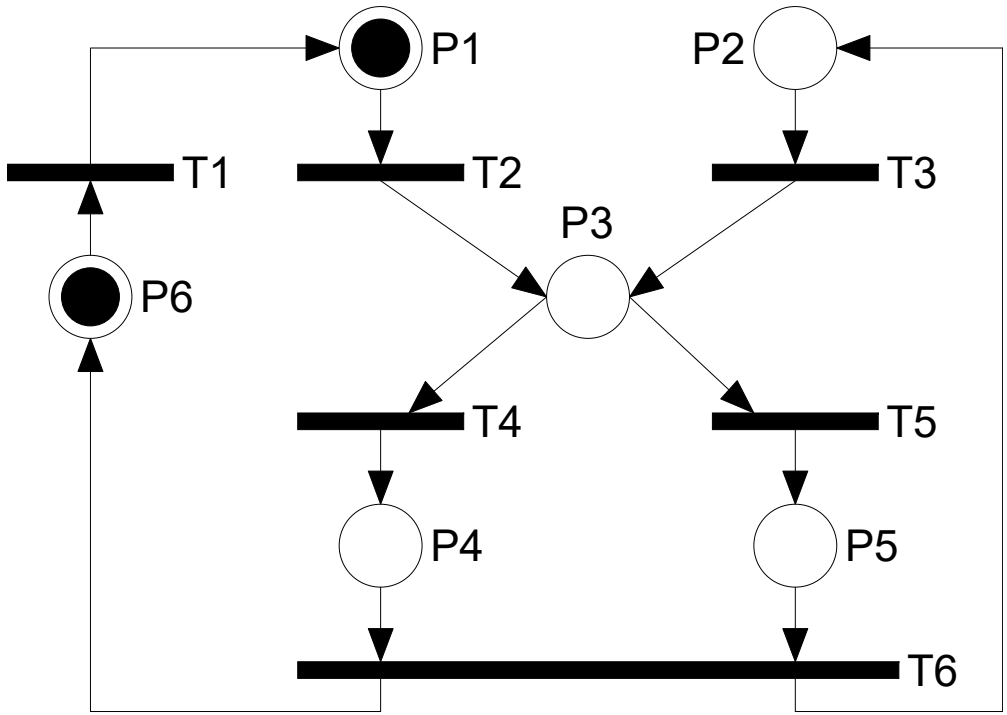


Unabhängigkeit von Lebendigkeit und Reversibilität

- Beispiel: nicht lebendig und nicht reversibel

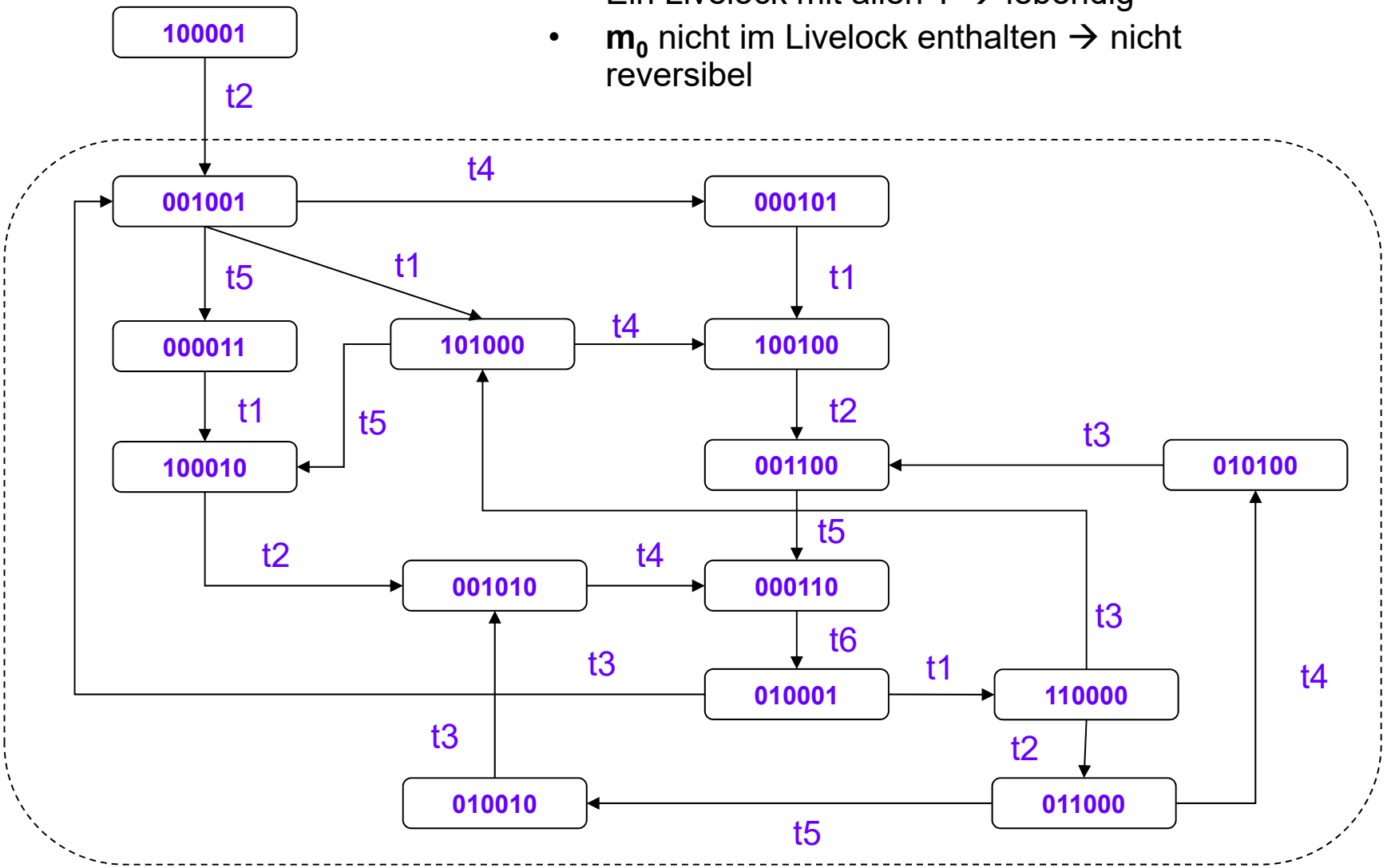


- Beispiel: lebendig aber nicht reversibel



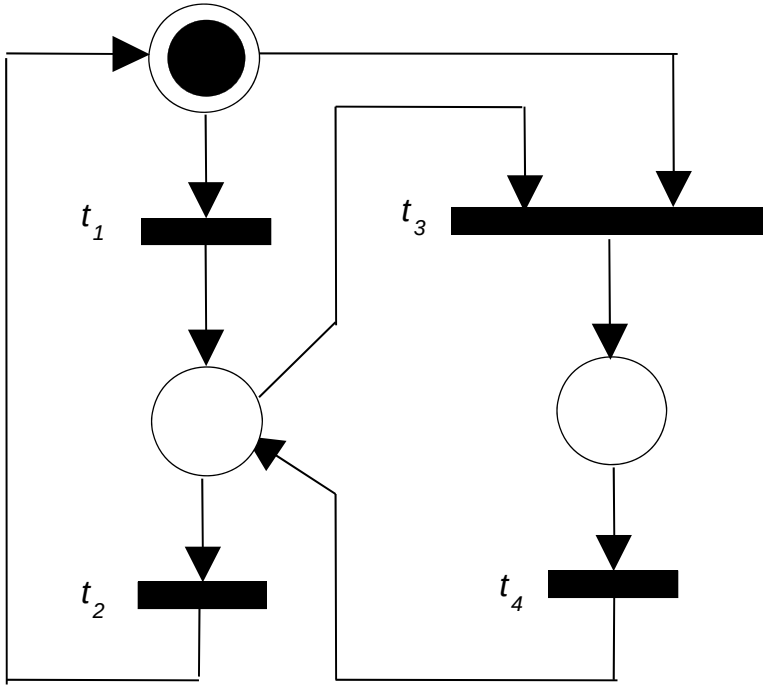
Erreichbarkeitsgraph zum Beispiel

- kein Deadlock
- Ein Livelock mit allen T → lebendig
- $\mathbf{m_0}$ nicht im Livelock enthalten → nicht reversibel



Unabhängigkeit von Lebendigkeit und Reversibilität

- Beispiel: nicht lebendig aber reversibel



Synonym: Stellen-Transitions-Netz (S/T-Netz)

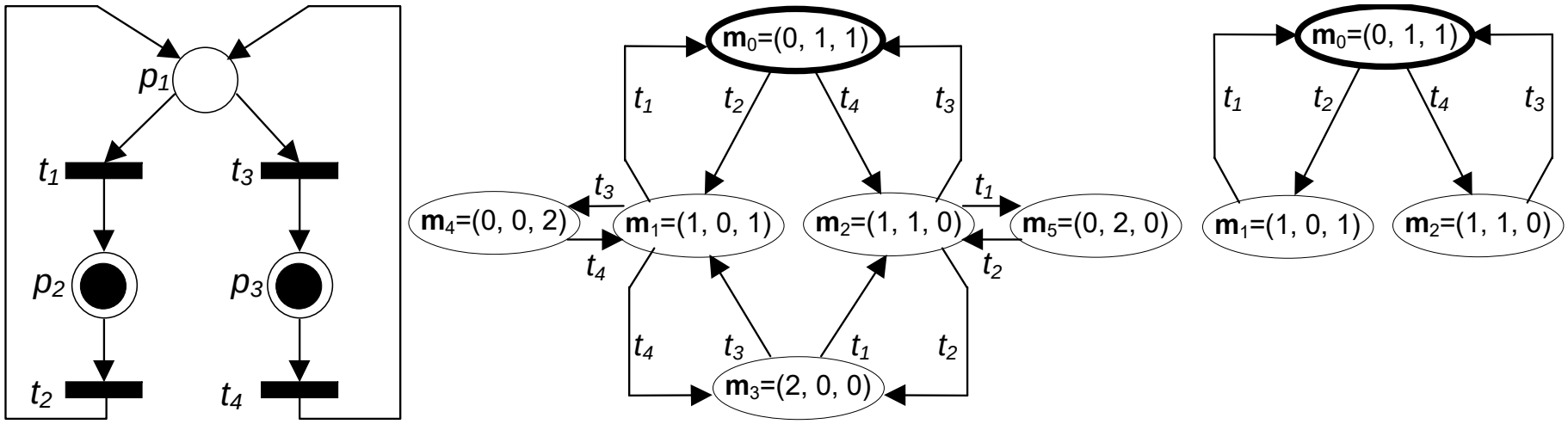
Struktur

- Ein Platz-Transitions-Netz (PT-Netz) ist ein Quadrupel PN = ($\boldsymbol{P}$, $\boldsymbol{T}$, $\boldsymbol{F}$, $\mathbf{m}_0$) bestehend aus einem gewöhnlichen Petrinetz ($\boldsymbol{P}$, $\boldsymbol{T}$, $\boldsymbol{F}$) und einer nicht-negativen, ganzzahligen Anfangsmarkierung $\mathbf{m}_0$: $\boldsymbol{P} \rightarrow \{0, 1, 2, 3, ...\}^{|\boldsymbol{P}|}$.

Dynamik

- Eine Transition t in einem PT-Netz ist konzessioniert (enabled) unter einer Markierung $\mathbf{m}$, En(t, $\mathbf{m}$) = True, wenn Alle Plätze in ihrem Vorbereich markiert sind:
  En(t, $\mathbf{m}$) = True $\Leftrightarrow \forall\, p \in \bullet t$: m($p$) > 0
- Eine konzessionierte Transition **KANN** schalten (feuern).
- Mehrere gleichzeitig konzessionierte Transitionen schalten **EINZELN** nacheinander, niemals gleichzeitig.
- Beim Schalten wird allen Plätzen im Vorbereich eine Marke entzogen, und allen Plätzen im Nachbereich eine Marke hinzugefügt.

NEIN



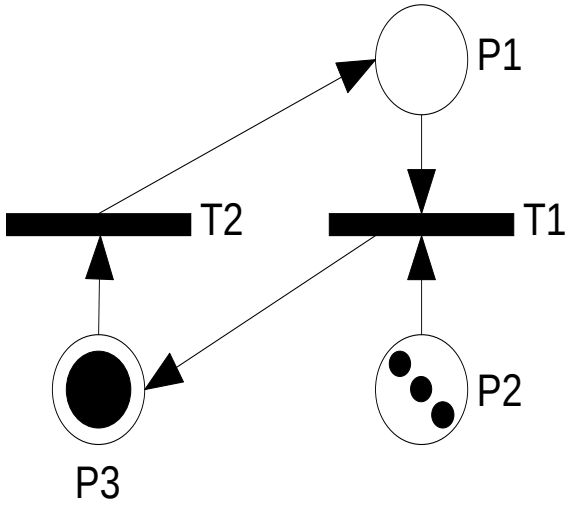
Deshalb zur weiteren Unterscheidung: Namen für die Konzessionsregeln:

- BE-Netz: starke Konzessionsregel
- ST-Netz: schwache Konzessionsregel

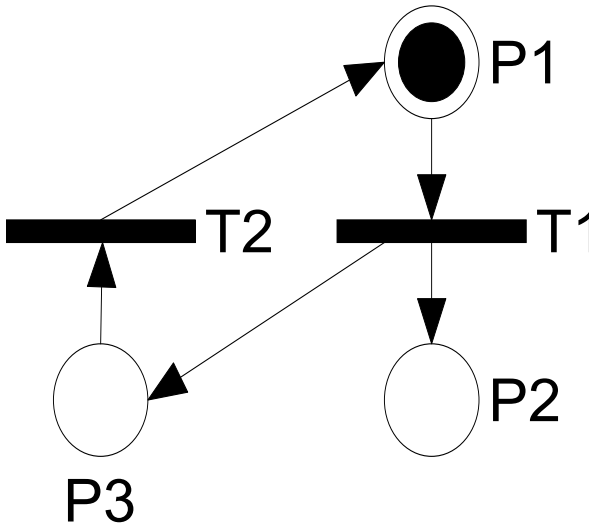
Eigenschaften beim PT-Netz

- Konflikt, Konfliktfreiheit
  - Nur Konflikte aufgrund gemeinsamer Vorbereichsplätze möglich
- Kontakt, Kontaktfreiheit
  - im PT-Netz nicht definiert
- Lebendigkeit
  - analog zu BE-Netz
- Deadlockfreiheit
  - analog zu BE-Netz
- Reversibilität
  - analog zu BE-Netz
- Zusätzlich: Beschränktheit und Sicherheit
  - Beim PT-Netz können mehrere Marken auf einem Platz liegen.
  - Ein Petrinetz heißt **k-beschränkt** (oder einfach beschränkt), wenn es eine obere Grenze k für die Anzahl der Marken auf einem Platz gibt, d.h. wenn keine Markierung erreichbar ist bei der eine Komponente von **m** größer k ist.
  - Für k=1 heißt das Netz **sicher**.
  - Für ein unbeschränktes Netz ist die Erreichbarkeitsmenge, d.h. die Zahl der Zustände nicht endlich.
  - Nachweis von Beschränktheit und Sicherheit kann durch Berechnung der Erreichbarkeitsmenge erfolgen.

Beispiel zu Beschränktheit

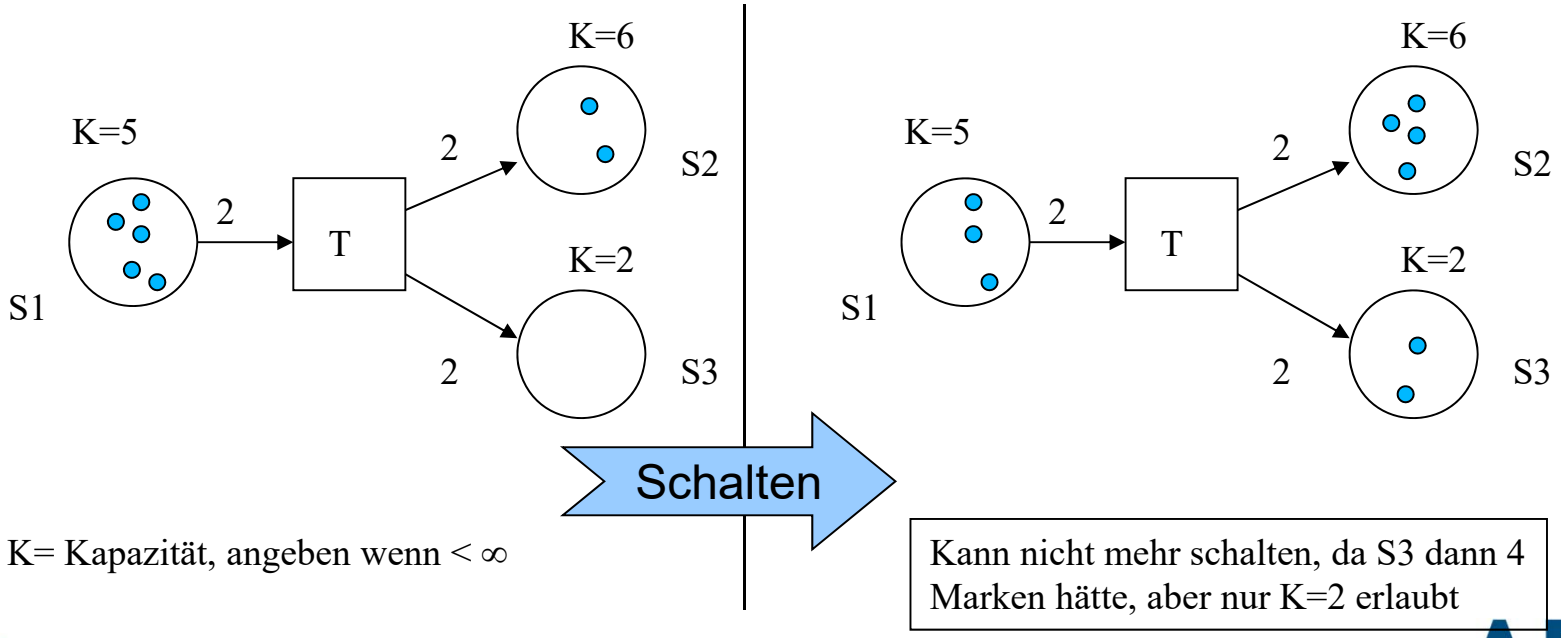


beschränkt (k=3)



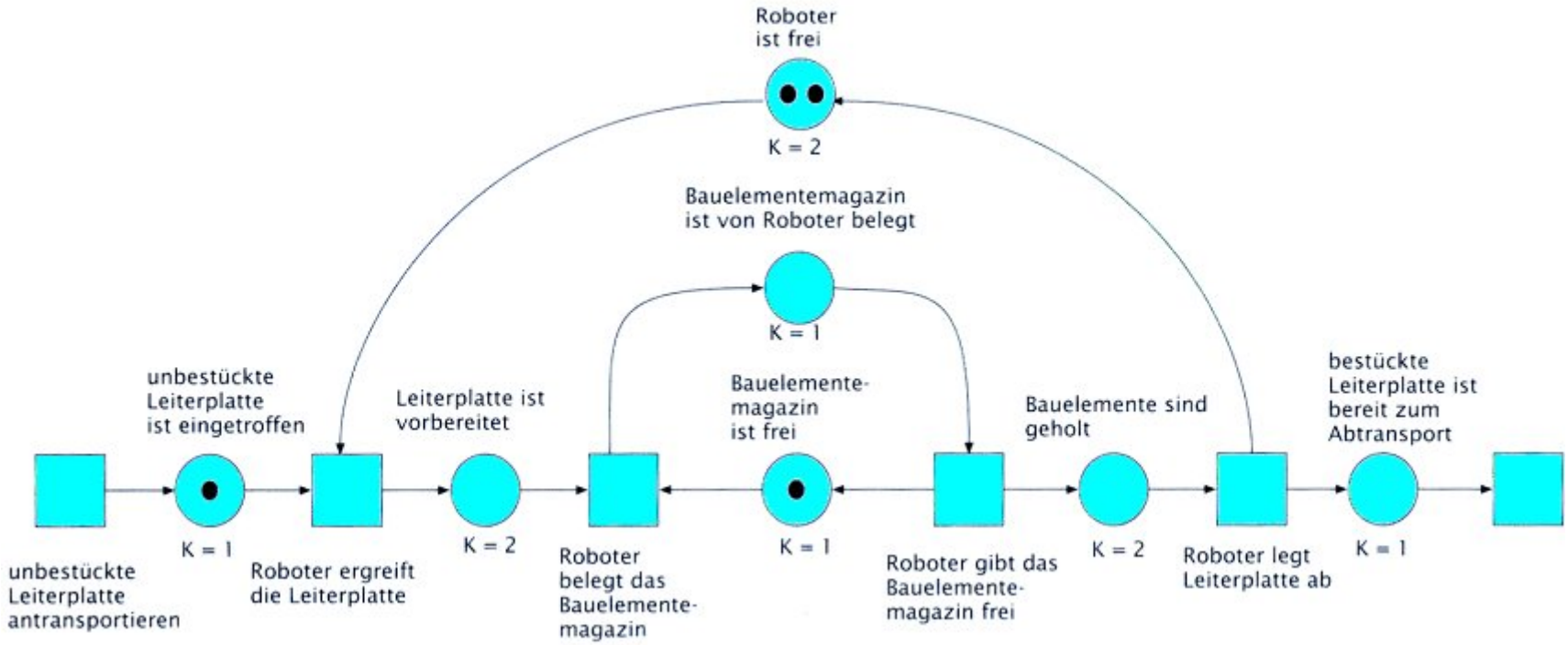
nicht beschränkt

- Kapazitäten
  - Es wird für jeden Platz festgelegt, wie viele Marken er maximal aufnehmen kann.
  - Nachteil: Dadurch können wie im BE-Netz Kontakte und Konflikte aufgrund der Ausgangsplätze auftreten.
  - Vorteil: Zustandsmenge ist auf jeden Fall beschränkt.
- Kantengewichte
  - An die Kanten werden Gewichte angeschrieben.
  - Beim Schalten werden nicht mehr nur einzelne Marken transportiert sondern jeweils so viele Marken, wie das Kantengewicht angibt.



K= Kapazität, angeben wenn < ∞

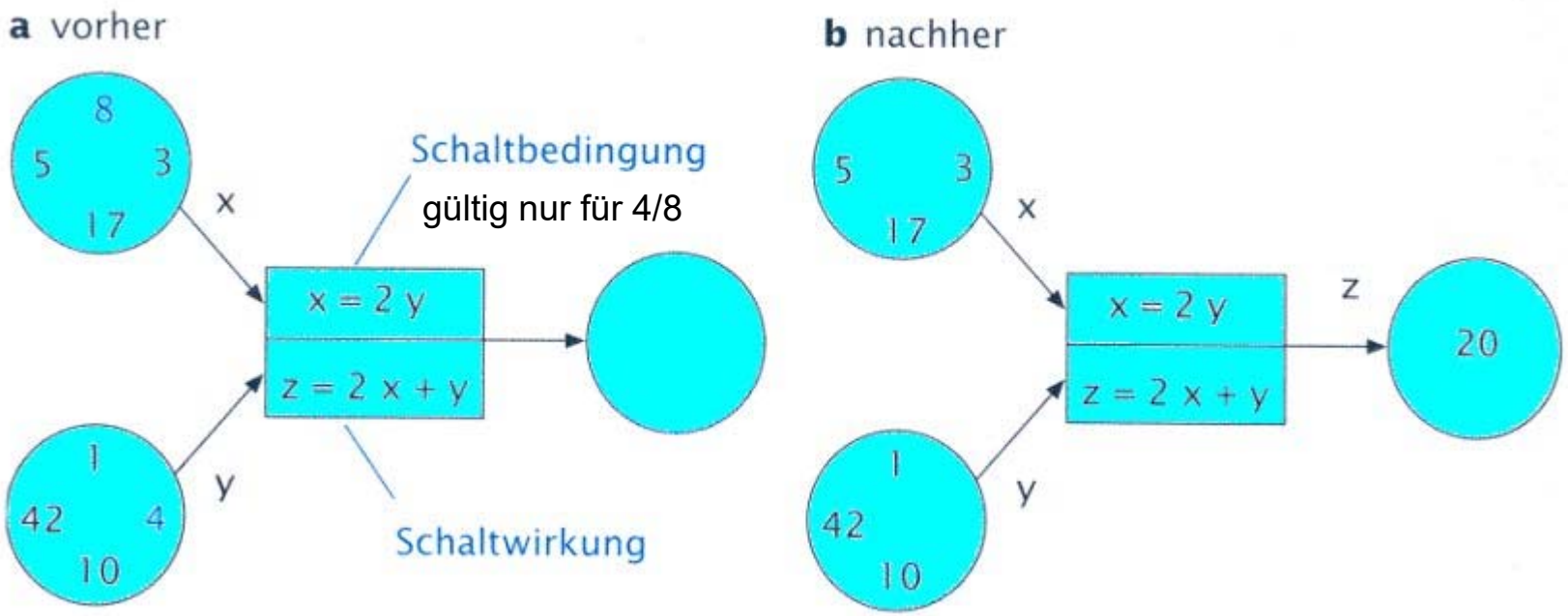
Kann nicht mehr schalten, da S3 dann 4 Marken hätte, aber nur K=2 erlaubt



Abstraktion von zwei Robotern auf Roboter mit K=2.

Quelle: Balzert

- Verwendung von individuellen / gefärbten Marken.
  - Individuelle Marken werden durch Wert, z.B. eine Zahl, repräsentiert.
  - Transitionen können nur beim Vorliegen bestimmter Marken schalten.
  - Den Transitionen werden Schaltbedingungen mit Variablen zugeordnet, die die von den Eingabeplätzen erhaltenen Marken repräsentieren.



Zeitbewertete Petrinetze

Zur Untersuchung der Leistungsfähigkeit oder zur Simulation eines Systems kann die Dauer einer Aktion festgelegt erden.

Die Marken müssen daher eine bestimmte Zeitdauer (t>0) auf einem Platz verharren, bevor sie von einer Transition verbraucht werden können.

Die Zeitintervalle können in den Plätzen, den Transitionen oder den Kanten festgelegt werden, verharren müssen die Marken aber jeweils auf den Plätzen.

Stochastische Petrinetze

Zur Untersuchung der Leistungsfähigkeit oder zur Simulation eines Systems kann auch die Wahrscheinlichkeit für das Eintreten von Ereignissen festgelegt werden.

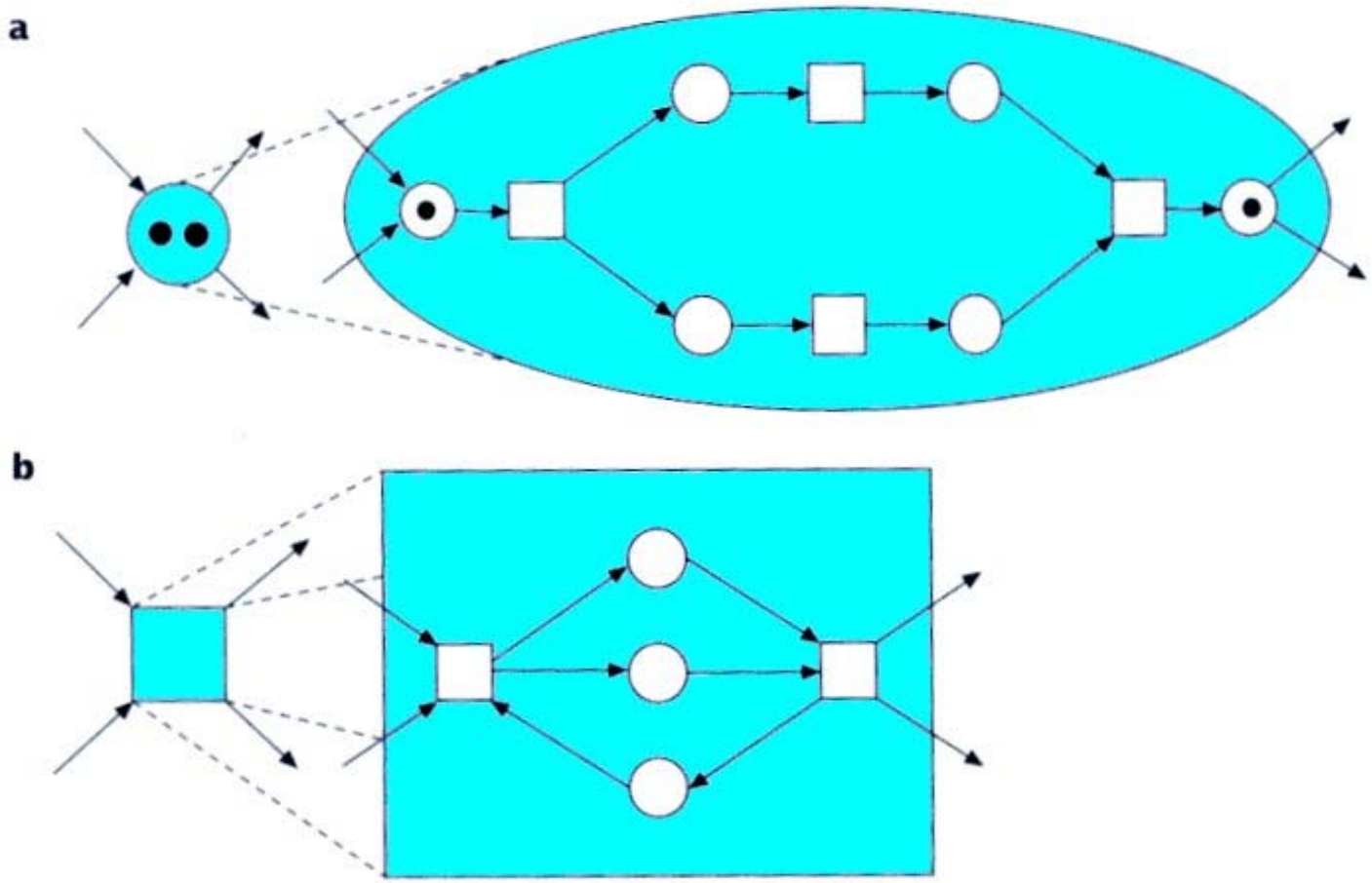
Kombinationen sind möglich: Beispielsweise Angabe von Zeitdauern über Mittelwert und Varianz

- Zur besseren Strukturierung von Petrinetzen können Netzknoten durch detaillierte Unternetze verfeinert werden

- Netzknoten können zu einer höheren Einheit zusammengefasst werden.

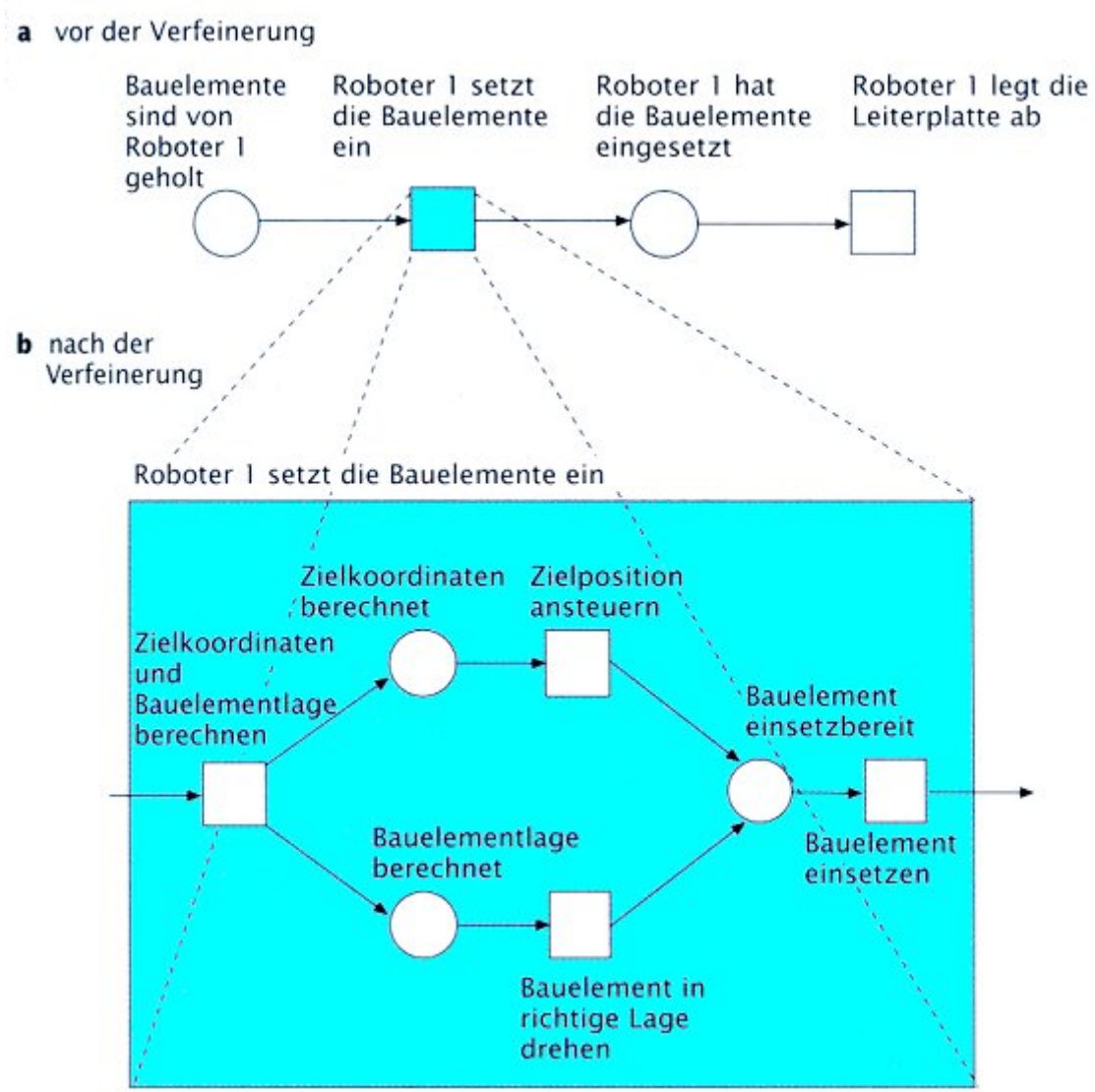
- Regeln zur Konsistenzerhaltung:
  - Zusätzliche Pfeile können nur innerhalb eines Baumknotens zwischen Plätzen und Transitionen eingefügt werden, d.h. es dürfen zwischen Baumknoten auf tieferer Ebene keine neuen Pfeile gezogen werden.
  - Pfeile, die von einem Unternetz weg- oder zu ihm hinführen, müssen bereits vor der Verfeinerung vorhanden sein und die vorgegebenen Richtungen müssen berücksichtigt werden.
  - Eine Verfeinerung muss markengetreu erfolgen, d.h. sie soll gleich viele Marken abgeben, wie sie verbraucht.

Markengetreue Verfeinerung eines Platzes a und einer Transition b



Quelle: Balzert

Beispiel einer Transitionsverfeinerung „Unternetz“



Quelle: Balzert

- Kanal-Agenten-Netz

Hinzufügen von einfacher Dynamik

- Bedingungs-Ereignis-Netz

Hinzufügen von mehreren Marken (Modellierung von Ressourcen)

- Platz-Transitions-Netz

Hinzufügen zusätzlicher Randbedingungen

- Platz-Transitions-Netz mit Kapazitäten und Gewichten

Hinzufügen komplexer Bedingungen bzw. komplexer Merkmale

- Farbiges Netz oder Prädikats-Transitions-Netz

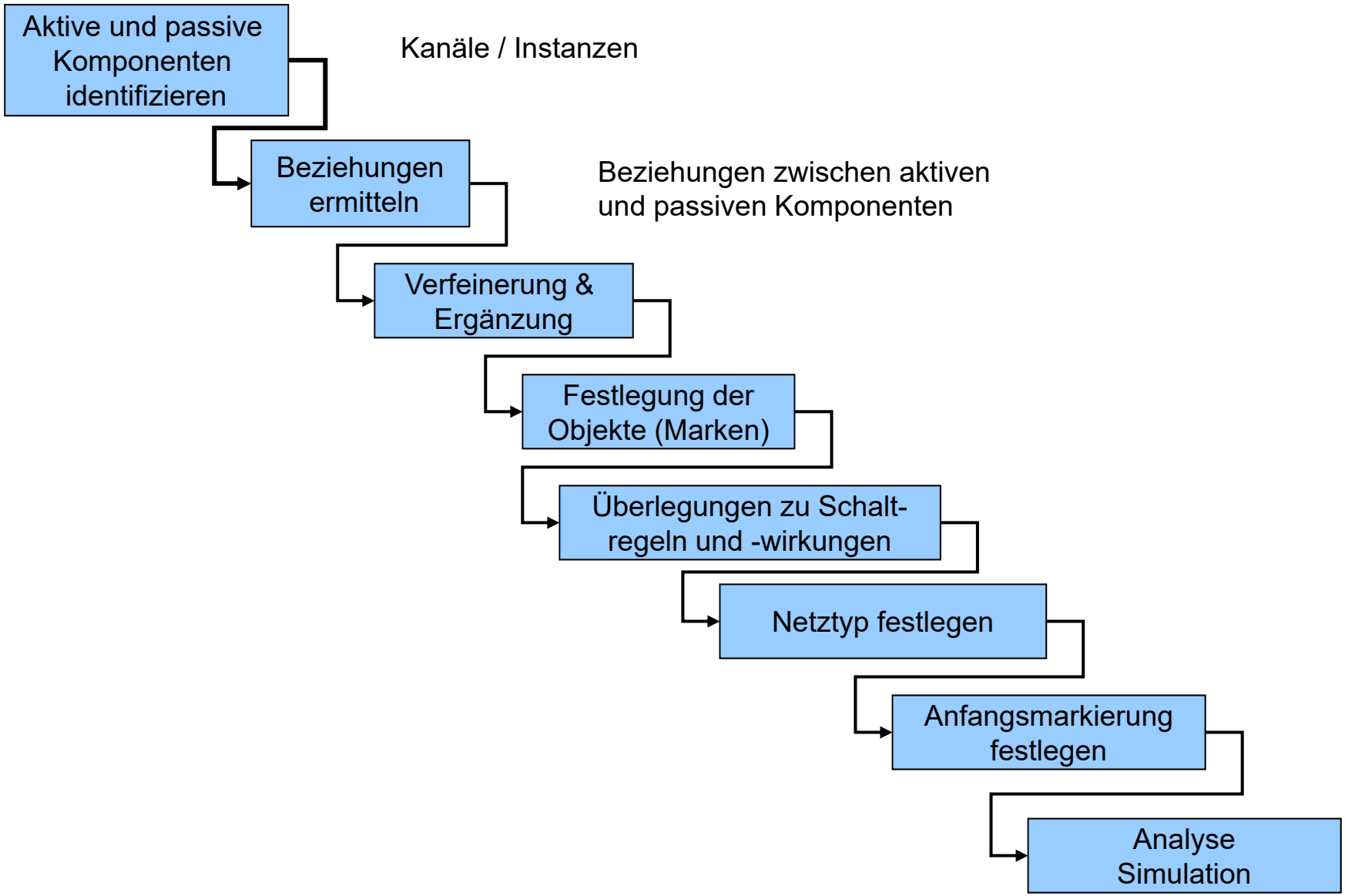
Bei allen Netzen zusätzlich möglich:

- *Hierarchische Verfeinerung*

Bei allen dynamischen Petrinetzen zusätzlich möglich:

- *Beschreibungselementen für Zeiten*
- *und Wahrscheinlichkeiten*

Vorgehensweise/Methodik



Zusammenfassung

- Eigenschaften von Petrinetzen
  - Konflikt
  - Kontakt
  - Lebendigkeit
  - Reversibilität
  - Livelock, Deadlock
- Erweiterungen