Angewandte Informatik I

Wintersemester 2006/2007

Universität Karlsruhe (TH)
Forschungsuniversität • gegründet 1825

Institut für Angewandte Informatik und
Formale Beschreibungsverfahren AIFB

Prof. Dr. Andreas Oberweis, Prof. Dr. Rudi Studer,
Dr. Pascal Hitzler, Dipl. Wirtsch. Inf. Victor Pankratius,
Sebastian Rudolph

Gliederung

1 Einführung und Modellierung (RS)

2 UML-Vertiefung (RS)

3 ER-Modell (RS)

4 Description Logics (RS)

5 Das relationale Datenmodell (AO)

6 DB-Entwurf (AO)

7 Petri-Netze (AO)

7 Petrinetze

7 Petrinetze

Carl-Adam Petri

Carl-Adam-Petri-Preis
Wissenschaftspreis der Fakultät für Wirtschafts-wissenschaften für das Fachgebiet Informatik

Grundlagen der Petrinetz-Theorie:

- Dissertation von **Carl-Adam Petri** (Kommunikation mit Automaten), **1962**, TH Darmstadt
- bis etwa **1985:**
  viele theoretische Arbeiten, insbesondere am Institut für formale Grundlagen der GMD unter der Leitung von Petri
- **danach:**
  zunehmende Verbreitung als praktisch anwendbarer Formalismus, durch Aufkommen von Werkzeugen und höheren Netzklassen.
- Mehrere Tausend Publikationen über Petrinetze
- Petri hat im Dezember **1997** den **Werner-von-Siemens-Ring** für sein Lebenswerk erhalten (höchste Ehrung in Deutschland im technischen Bereich), siehe Informatik Spektrum **1/1998**

7.1 Verhaltensmodellierung von Informationssystemen und Geschäftsprozessen

Verhaltensmodellierung

Beschreibung der Dynamik eines Informationssystems

- **mögliche Aktivitäten**
- **Vor- und Nachbedingungen einer Aktivität**
- **Zustände** (aller Bedingungen)
  - mögliche Ausprägung für jede einzelne Vor- bzw. Nachbedingung

7.1 Verhaltensmodellierung von Informationssystemen und Geschäftsprozessen (2|2)

- **Anfangszustand**
- **sequentielle Abläufe**
  (mögliche Folgen von Aktivitäten)
- **nichtsequentielle Abläufe**
  (Flüsse und Kausalbeziehungen)
- **erreichbare Zustände**
  (konkret auftretende Zustände in einem möglichen Ablauf)
- **dynamische Eigenschaften**
  (Invarianten, Erfüllung von Zielen,
  "es wird schließlich etwas Bestimmtes (Gewünschtes) eintreten")
- **Geschäftsprozesse**
  (mit Anfang, Ende, Varianten, ...)

7 Petrinetze

7.1 Verhaltensmodellierung von Informationssystemen und Geschäftsprozessen

7.2 Klassische Petrinetze (Stellen/Transitions-Netze)

7.2.1 Allgemeine Interpretation

7.2.2 Das Schalten einer Transition

7.3 Modellierung mit Petrinetzen

7.4 Netztransformationen

7.5 Netzmorphismen

7.6 Kanal/Instanzen-Netze

7.7 Stellen/Transitions-Netze

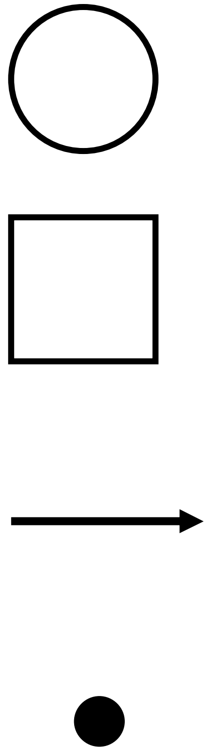
7.8 Modellierung von Prozessen mit Petrinetzen

7.9 Ausblick

7.2 Klassische Petrinetze (Stellen/Transitions-Netze) (1|1)

Ein Petrinetz besteht aus

- **Stellen**
- **Transitionen**
- **Verbindungen**
  zwischen Stellen und Transitionen
- **Marken in Stellen**



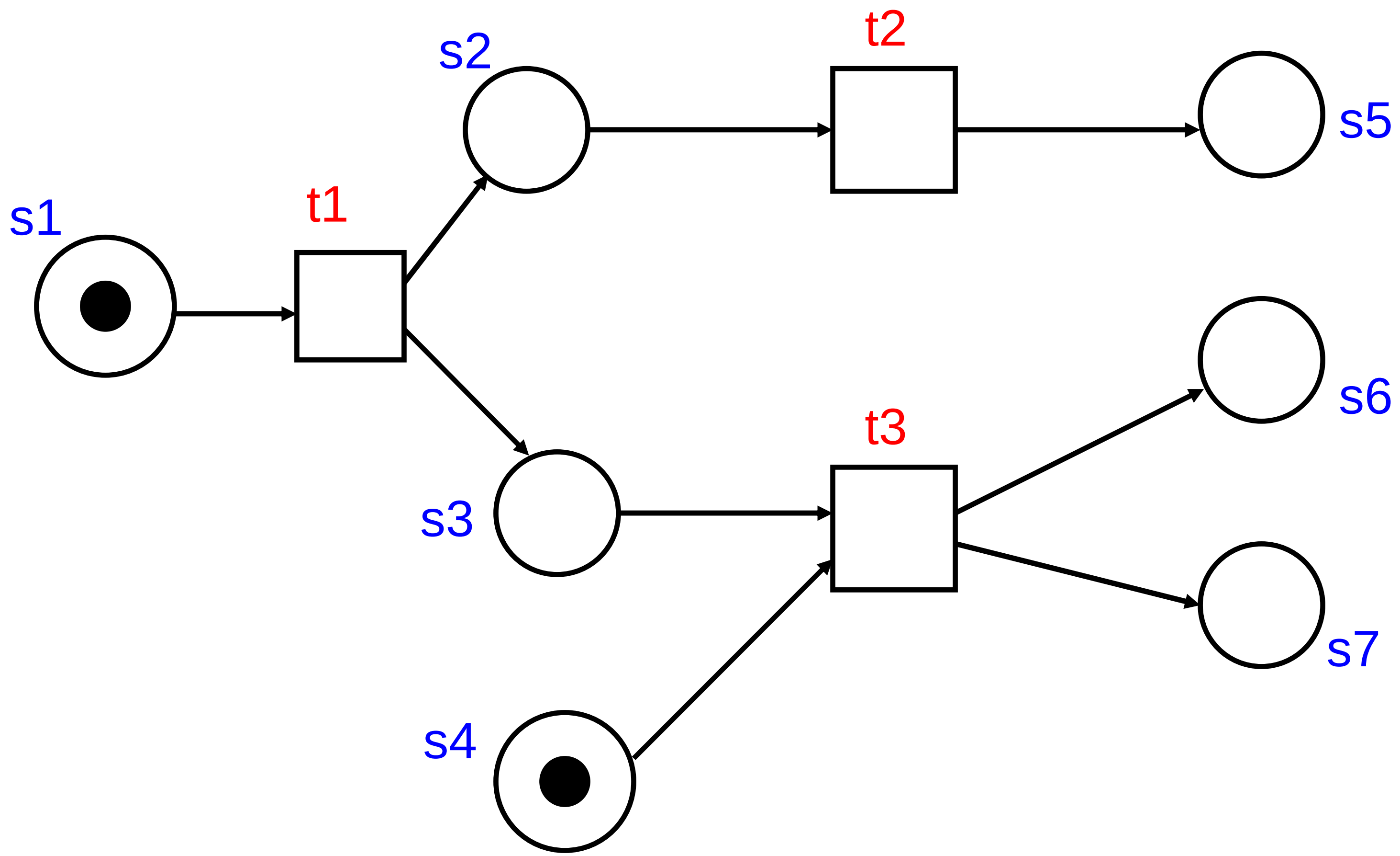
7.2.1 Allgemeine Interpretation

Transitionen	**—**	**Aktionen**
Stellen	**—**	**Bedingungen**
Verbindungen	**—**	**Beziehungen** (Vor- und Nachbedingungen einer Aktion)
Marken	**—**	**Zustände einer Bedingung**
Markierungen	**—**	**Gesamtzustände**

7.2.1 Allgemeine Interpretation

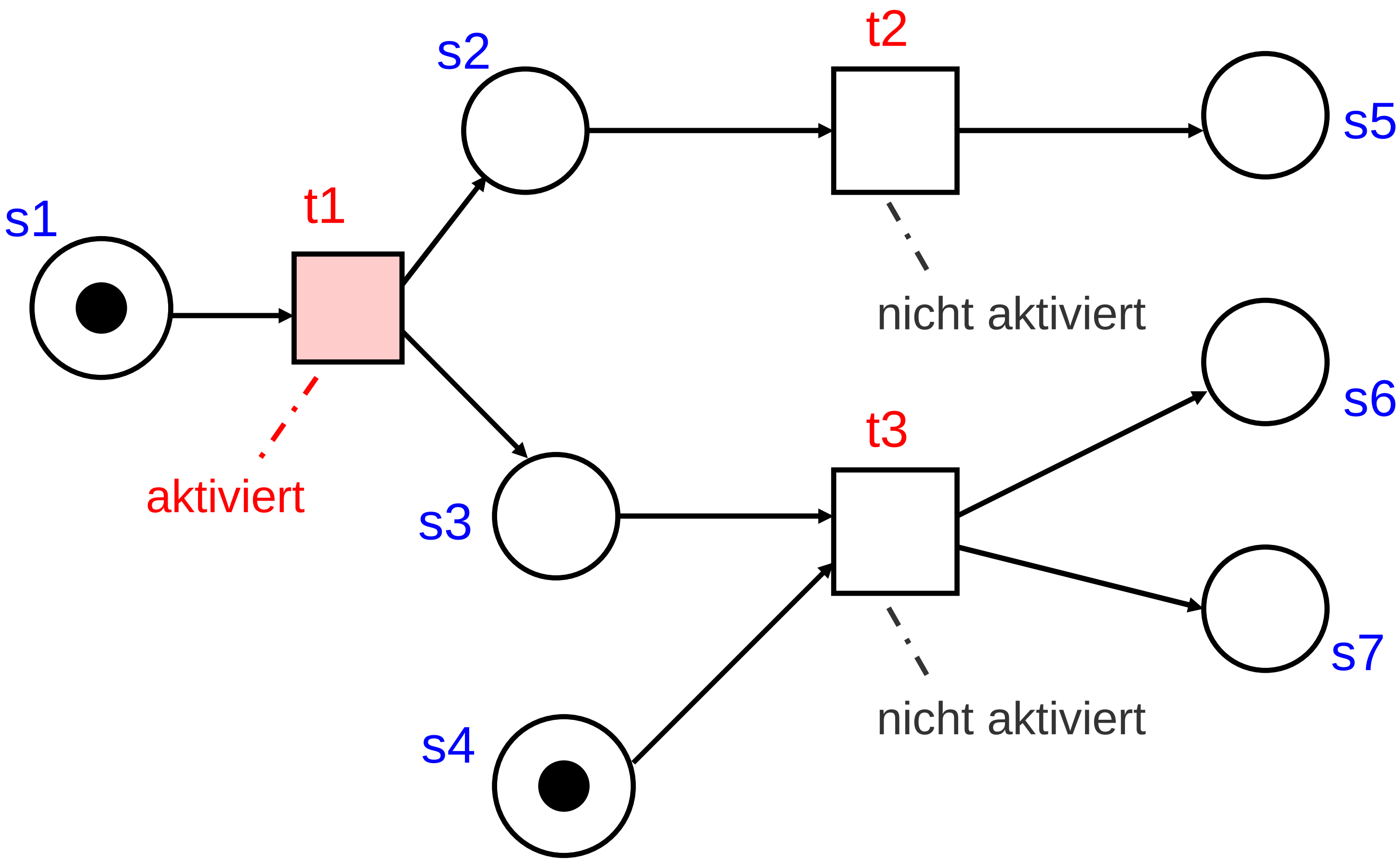
Beispiel:

- s1 Vorbedingung von t1
- s2 und s3 sind Nachbedingung von t1
- die Bedingungen s1 und s4 sind erfüllt



7.2.2 Das Schalten einer Transition

Eine Transition kann "**schalten**":
- Gegeben Ausgangsmarkierung: Marke jeweils in s1 und in s4
- Darin alle Vorbedingungen von t1 erfüllt, somit ist t1 "**aktiviert**"

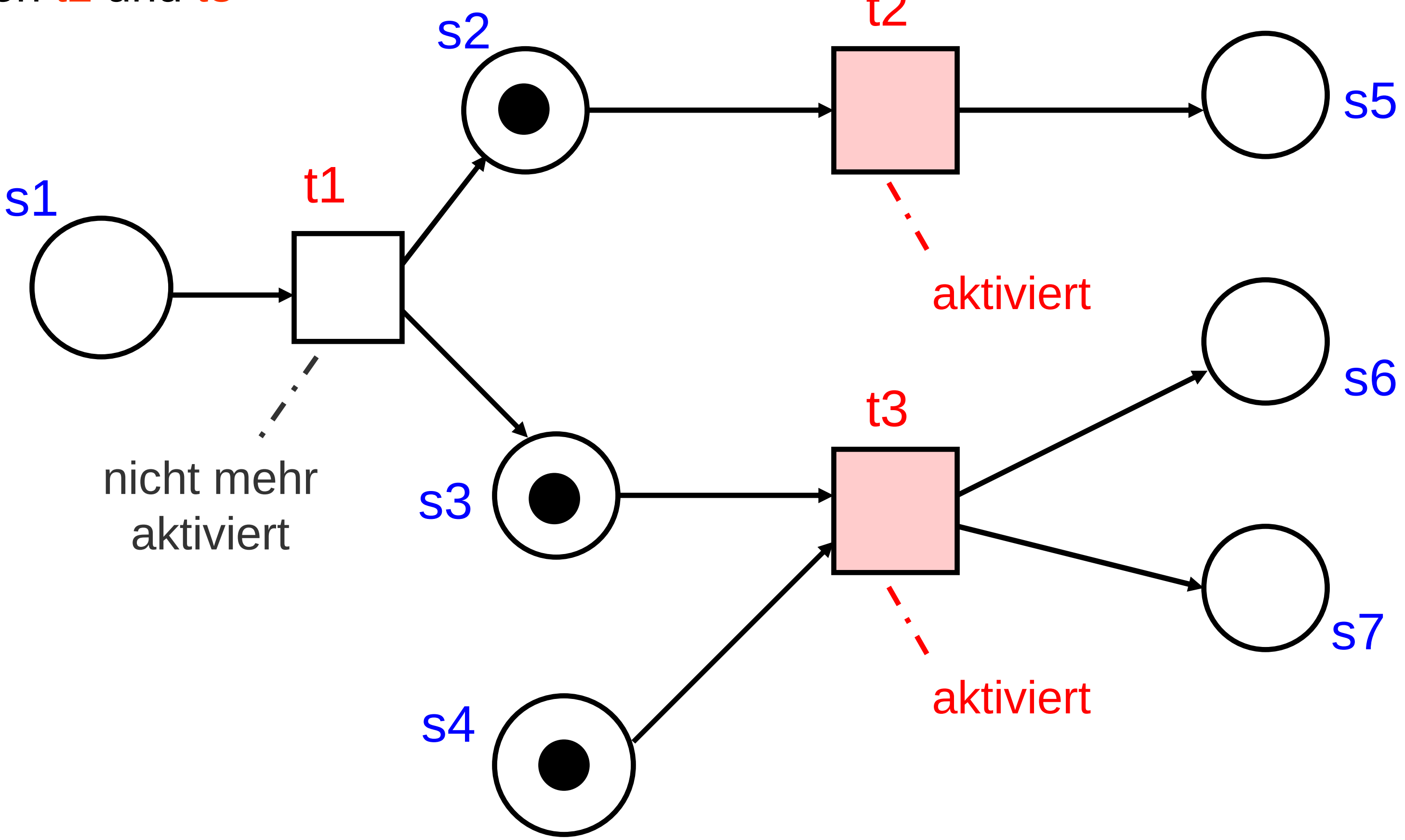


Schalten von Transition t1 $\Rightarrow$ nächste Folie

7.2.2 Das Schalten einer Transition

Folgemarkierung:

- Vorbedingung von t1 ist jetzt nicht mehr erfüllt, dafür sind alle Nachbedingungen von t1 erfüllt und somit die Vorbedingungen von t2 und t3



Jetzt können t2 oder t3 in beliebiger Reihenfolge (oder gleichzeitig) schalten.

7 Petrinetze

7.3.1 Grundlagen

Neben klassischen Petrinetzen existieren vielfältige Modellierungssprachen, die auf Petrinetzen beruhen.
(Sie werden deshalb auch oft als spezielle Petrinetztypen bezeichnet)

Gemeinsamkeit:

- Die **Systemkomponenten** sind durch **Stellen** und **Transitionen** und ihre **Verbindungen** dargestellt
- **Transitionen** beschreiben aktive (dynamische) Systemkomponenten
  - Aktionen,
  - Handlungen,
  - Transporte,
  - Transformationen,
  - Anweisungen,
  - Programme,
  - ...

7.3.1 Grundlagen

- **Stellen** beschreiben passive (statische) Systemkomponenten
  - Bedingungen,
  - Medien,
  - Materialbehälter,
    (z.B. Öltank: Marke gibt an ob dieser gefüllt ist;
    falls mehrere Marken erlaubt sind, können diese den Füllungsgrad angeben)
  - Datenträger,
    (z.B. Relationen in einer DB)
  - Puffer,
  - Nachrichtenkanäle,
  - ...

7.3.1 Grundlagen

- **Verbindungen** beschreiben
  (je nach der Transition)
  - Vor- und Nachbedingungen von Aktivitäten,
  - Start und Ziel von Transporten,
  - Eingabe und Ausgabe von Programmen,
  - ...

- **Marken** beschreiben **Zustände** von passiven Systemkomponenten
  - Gültigkeit von Bedingungen,
    (z.B. eine im Ablauf zuvor auszuführende Aktion ist beendet)
  - Füllungsgrad von Speichern,
    (siehe Bsp. Öltank;
    Marken können auch eine **bestimmte Struktur** haben, so dass evtl. eine Marke schon ausreicht, um den Füllungsgrad anzugeben)
  - Daten auf Datenträgern,
  - Nachrichten in Puffern,
  - ...

7.3.1 Grundlagen

Beispiel:

Bibliothek

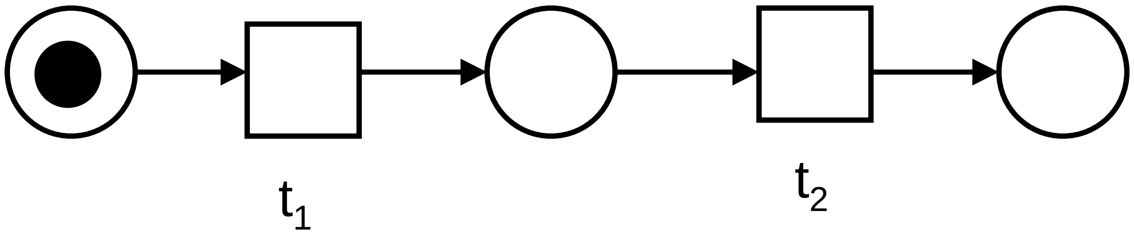
(VIKAR)

7.3.2 Was kann mit Petrinetzen modelliert werden?

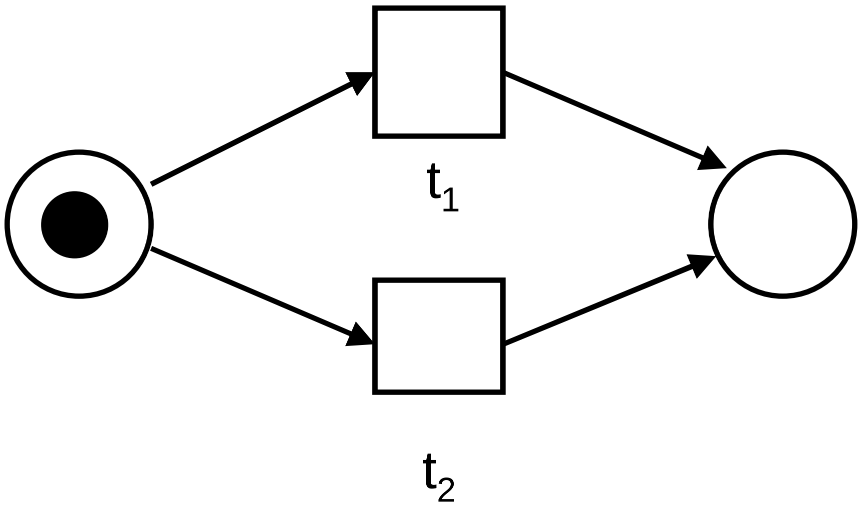
Kontrollstrukturen

Aktionen: t_1,t_2

1. sequentieller Ablauf: begin t_1; t_2 end;

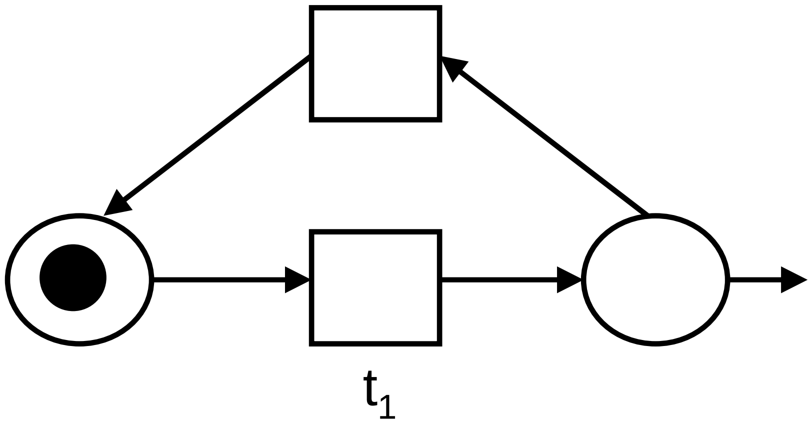


2. Alternative: if bed then t_1 else t_2;

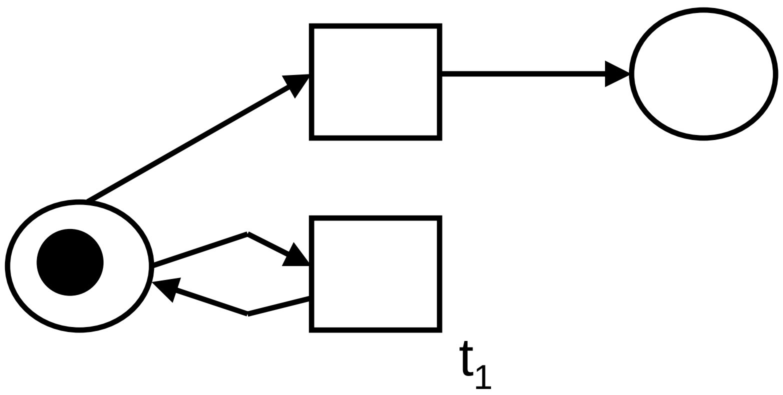


7.3.2 Was kann mit Petrinetzen modelliert werden?

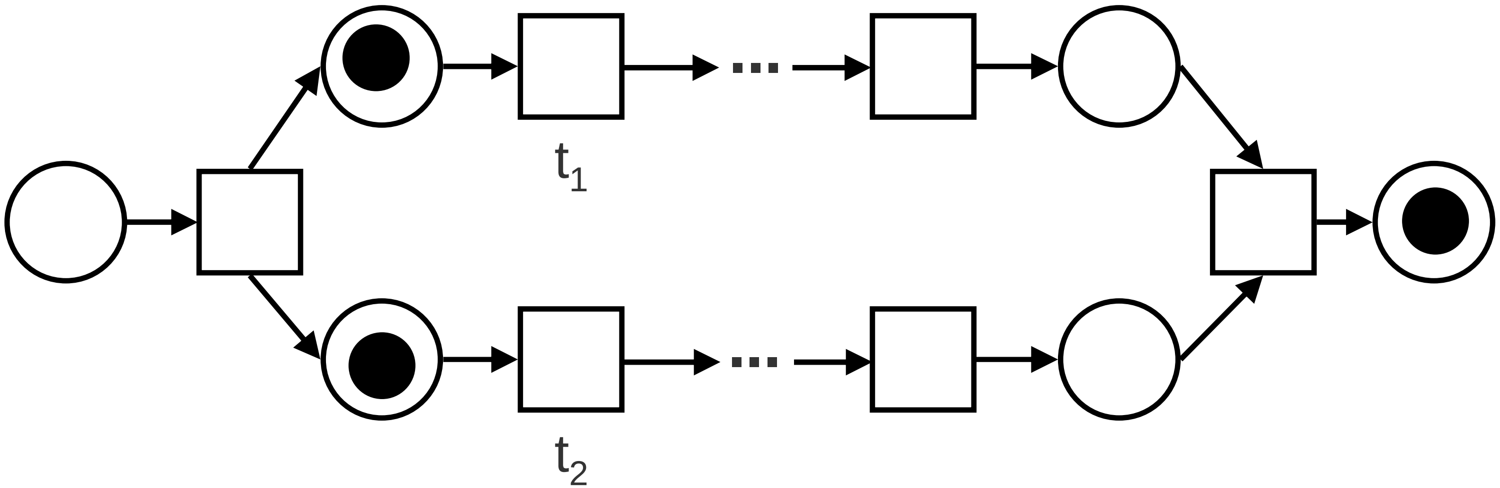
3. Wiederholung: repeat t_1 until bed;



while bed do t_1



4. Nebenläufigkeit und Synchronisation: begin t_1 ...|| t_2 ... end



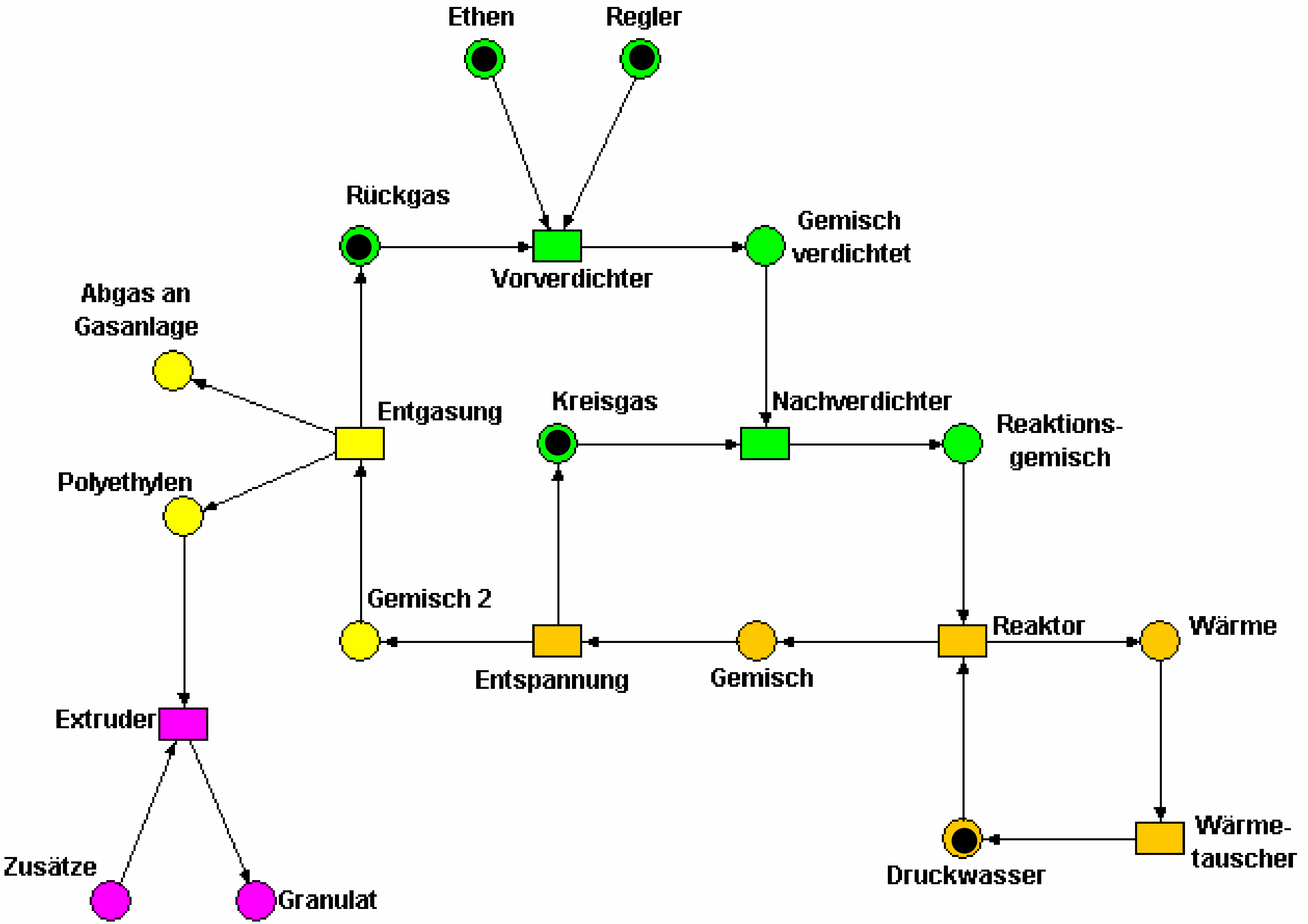
5. Eingabe von und Ausgabe zur Umwelt

7.3.2 Beispiel: Produktionsprozess in der Petrochemie

	Apparat bzw. Vorgang	Input	Output
1	**Vorverdichter**	Rückgas, Ethen, Regler	Gemisch verdichtet
2	**Nachverdichter**	Gemisch verdichtet, Kreisgas	Reaktionsgemisch
3	**Reaktor**	Reaktionsgemisch, Druckwasser	Wärme, Gemisch
4	**Wärmetauscher**	Wärme	Druckwasser
5	**Entspannung**	Gemisch	Gemisch2, Kreisgas
6	**Entgasung**	Gemisch2	Rückgas, Abgas an Gasanlage, Polyethylen
7	**Extruder**	Polyethylen, Zusätze	Granulat

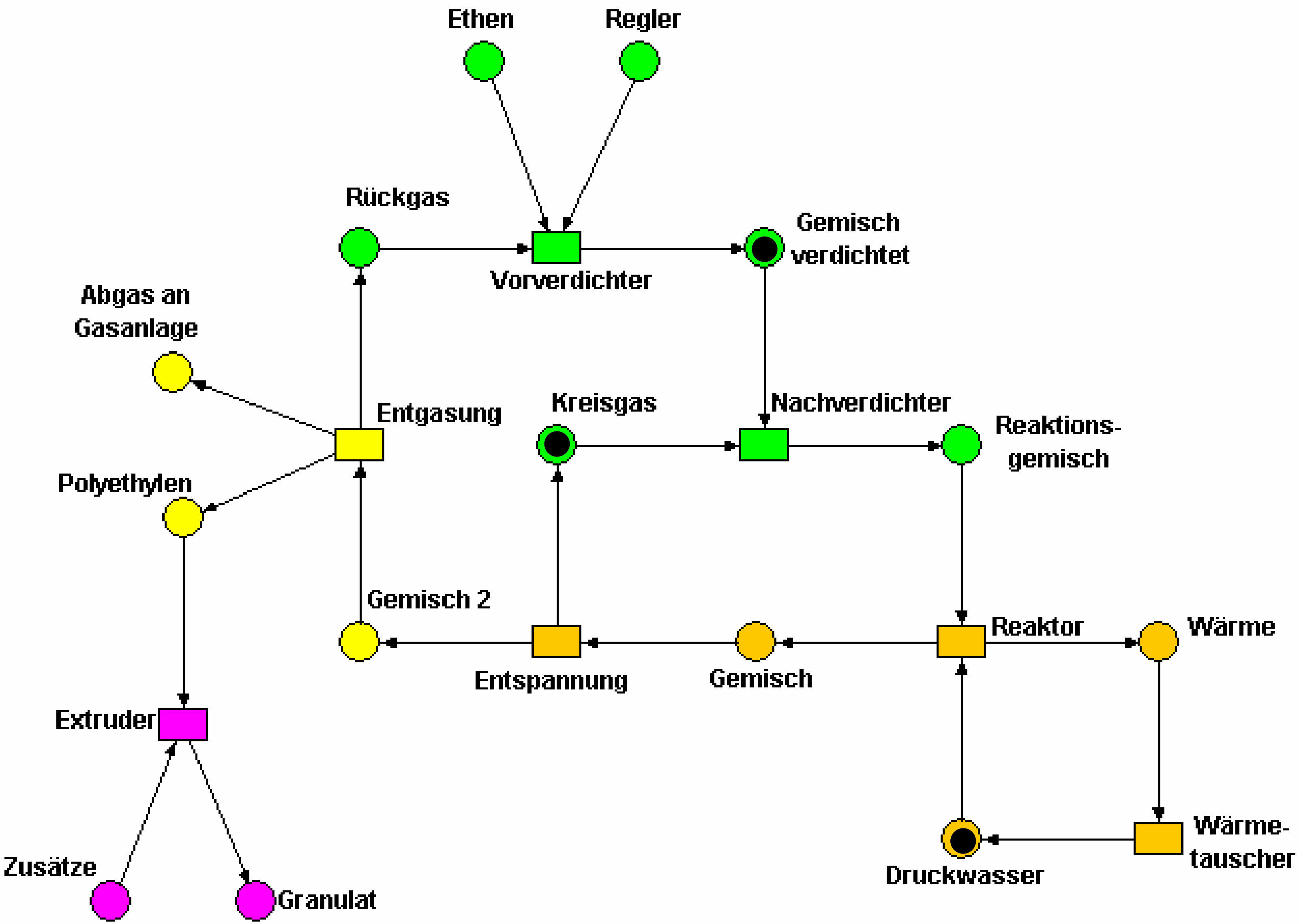
vgl. B. Rosenstengel, U. Winand, Petri-Netze, 4. Auflage, Vieweg 1991

7.3.2 Beispiel: Produktionsprozess in der Petrochemie



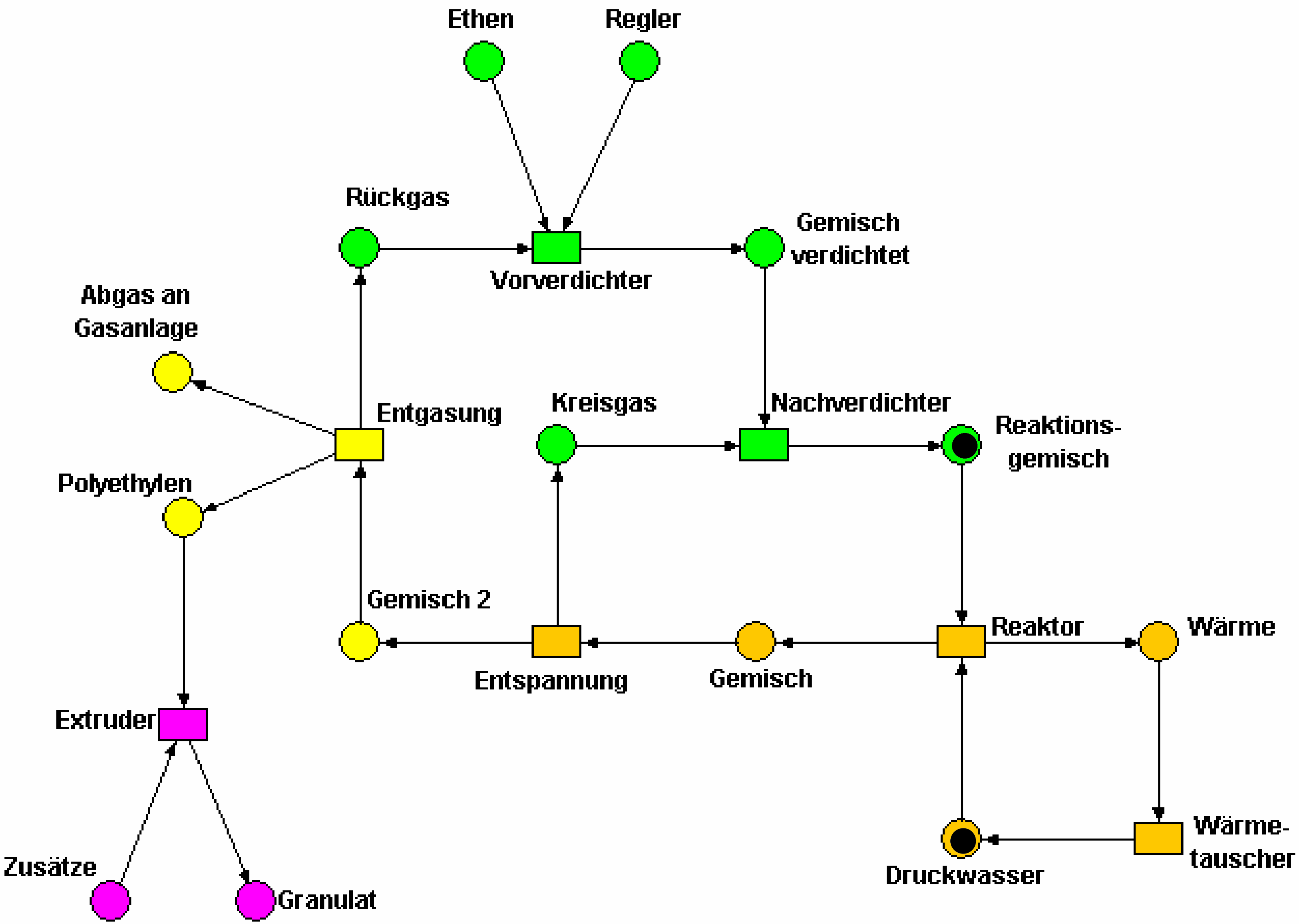
B. Rosenstengel, U. Winand, Petri-Netze, 4. Auflage, Vieweg 1991, Seite 124

7.3.2 Beispiel: Produktionsprozess in der Petrochemie



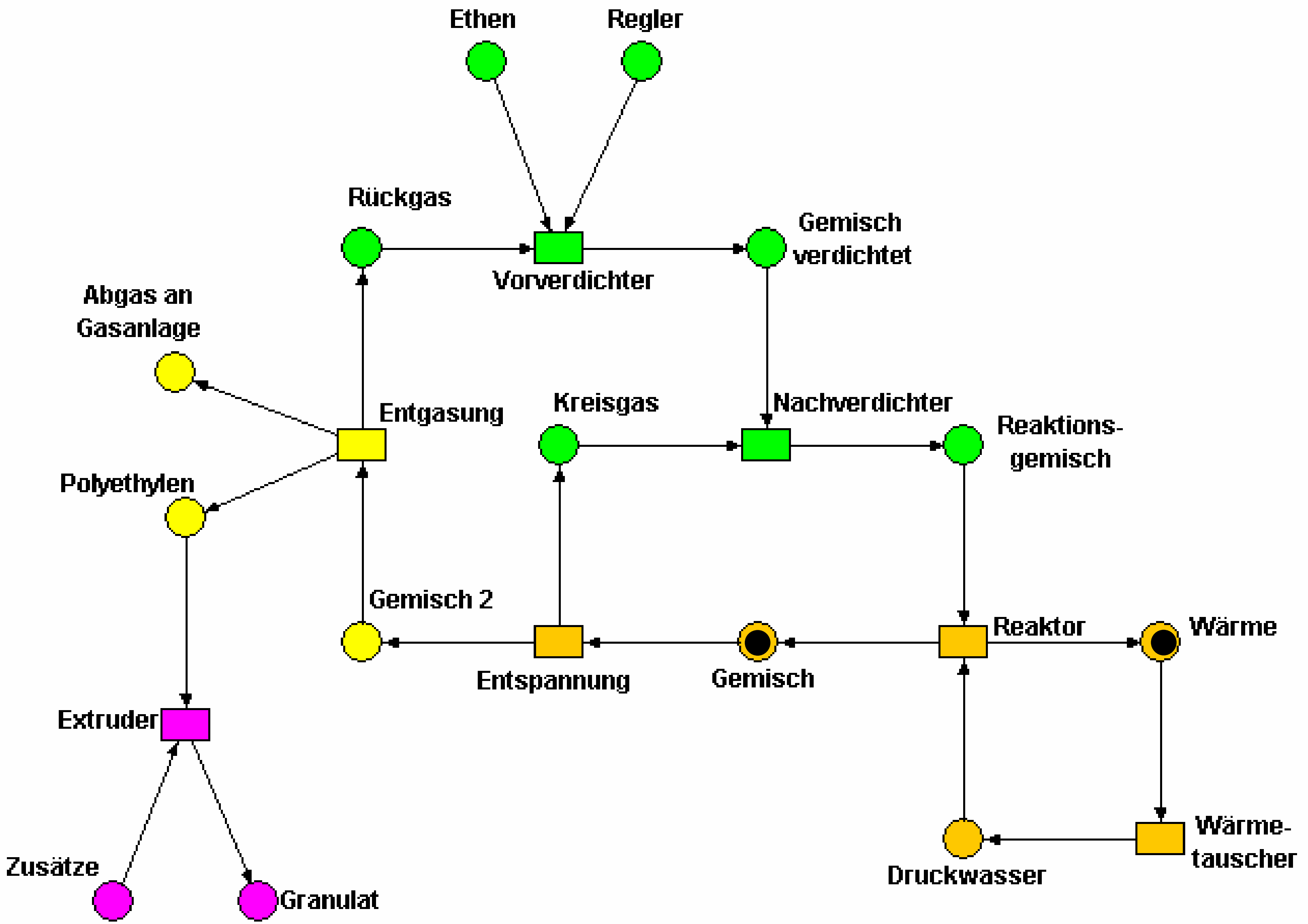
B. Rosenstengel, U. Winand, Petri-Netze, 4. Auflage, Vieweg 1991, Seite 124

7.3.2 Beispiel: Produktionsprozess in der Petrochemie



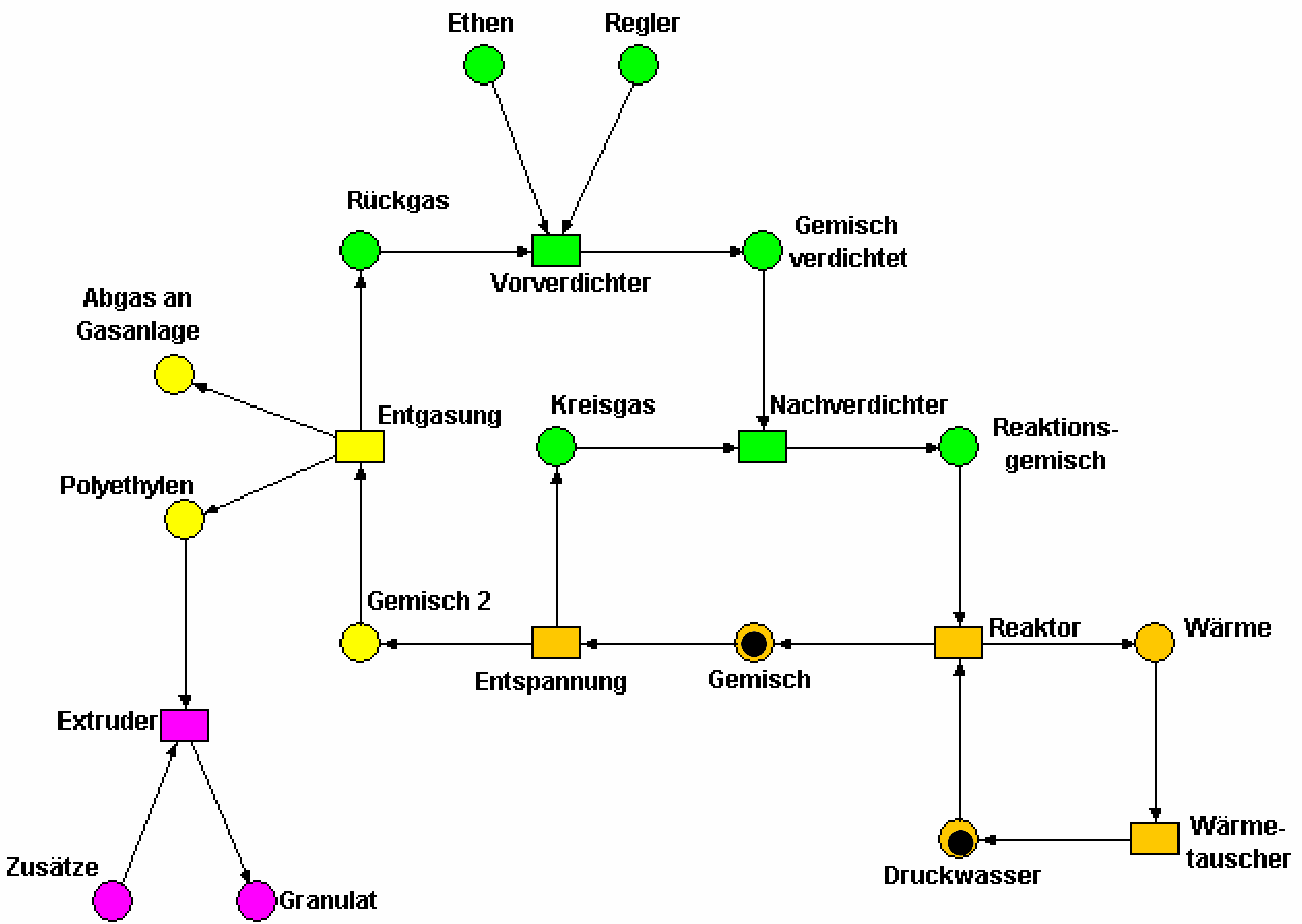
B. Rosenstengel, U. Winand, Petri-Netze, 4. Auflage, Vieweg 1991, Seite 124

7.3.2 Beispiel: Produktionsprozess in der Petrochemie



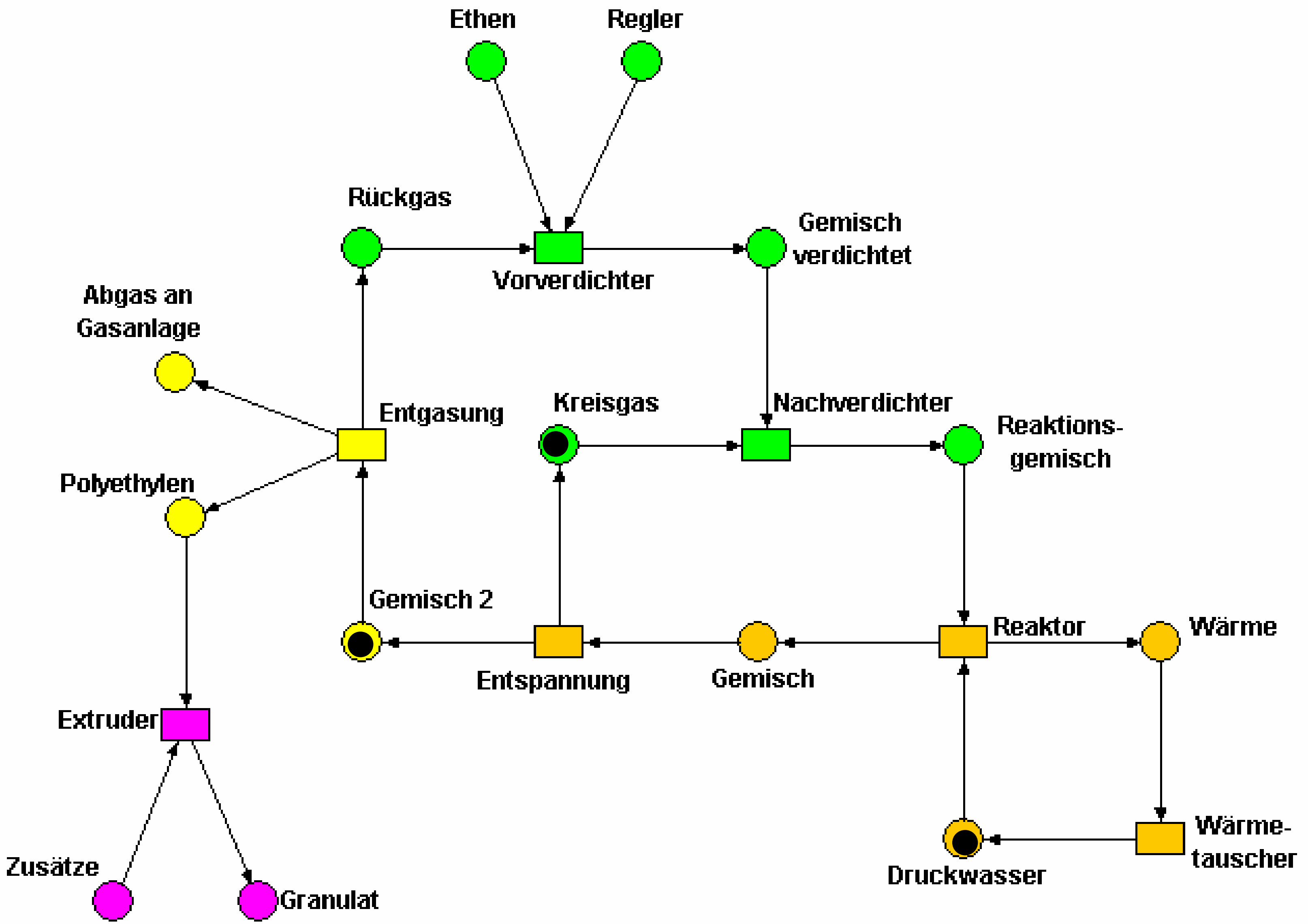
B. Rosenstengel, U. Winand, Petri-Netze, 4. Auflage, Vieweg 1991, Seite 124

7.3.2 Beispiel: Produktionsprozess in der Petrochemie



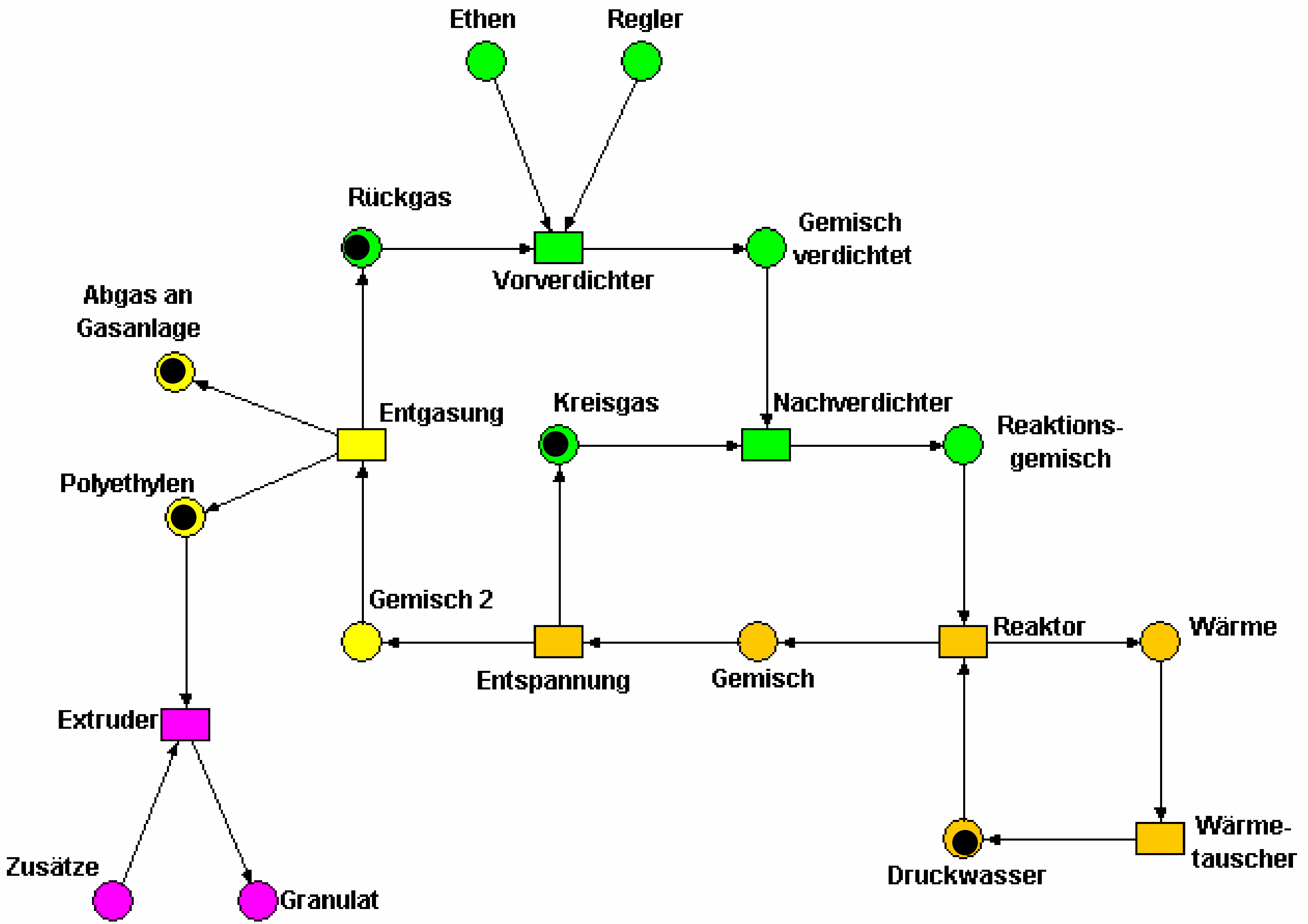
B. Rosenstengel, U. Winand, Petri-Netze, 4. Auflage, Vieweg 1991, Seite 124

7.3.2 Beispiel: Produktionsprozess in der Petrochemie



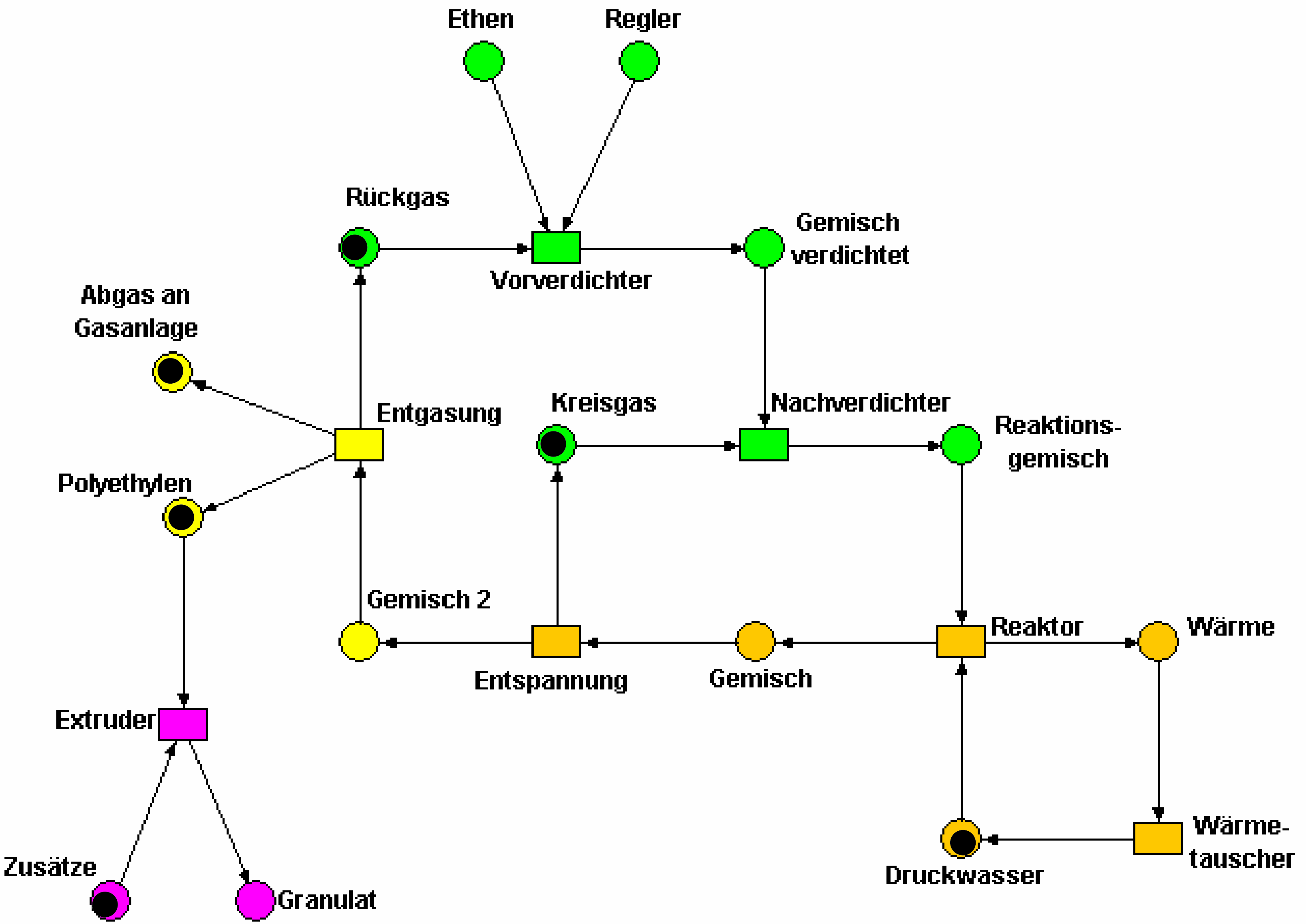
B. Rosenstengel, U. Winand, Petri-Netze, 4. Auflage, Vieweg 1991, Seite 124

7.3.2 Beispiel: Produktionsprozess in der Petrochemie



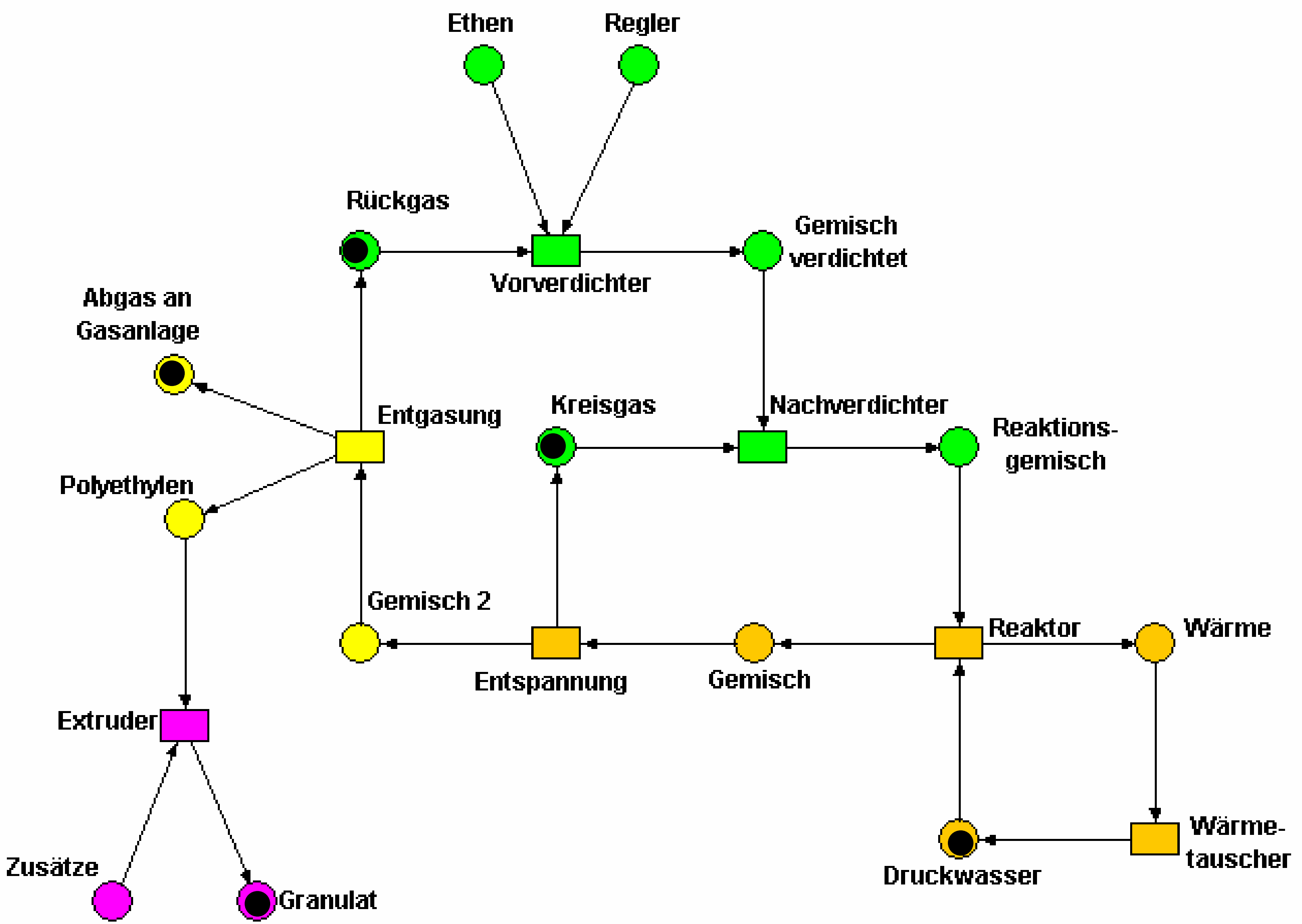
B. Rosenstengel, U. Winand, Petri-Netze, 4. Auflage, Vieweg 1991, Seite 124

7.3.2 Beispiel: Produktionsprozess in der Petrochemie



B. Rosenstengel, U. Winand, Petri-Netze, 4. Auflage, Vieweg 1991, Seite 124

7.3.2 Beispiel: Produktionsprozess in der Petrochemie



B. Rosenstengel, U. Winand, Petri-Netze, 4. Auflage, Vieweg 1991, Seite 124

7.3.3 Formale Definition von Petrinetzen

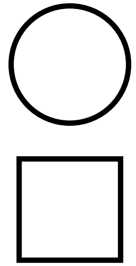
Definition: Struktur von Petrinetzen

Ein Petrinetz ist ein **Tripel N = (S,T,F)** mit

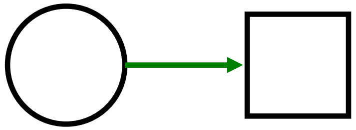
- S ,T sind endliche Mengen
- $S \cup T \neq \emptyset$
- $S \cap T = \emptyset$
- $F \subseteq (S \times T) \cup (T \times S)$ ist eine binäre Relation über $S \cup T$

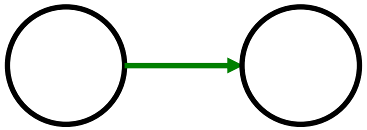
S: "Stellen"

T: "Transitionen"



F: $\longrightarrow$

erlaubt:  oder 

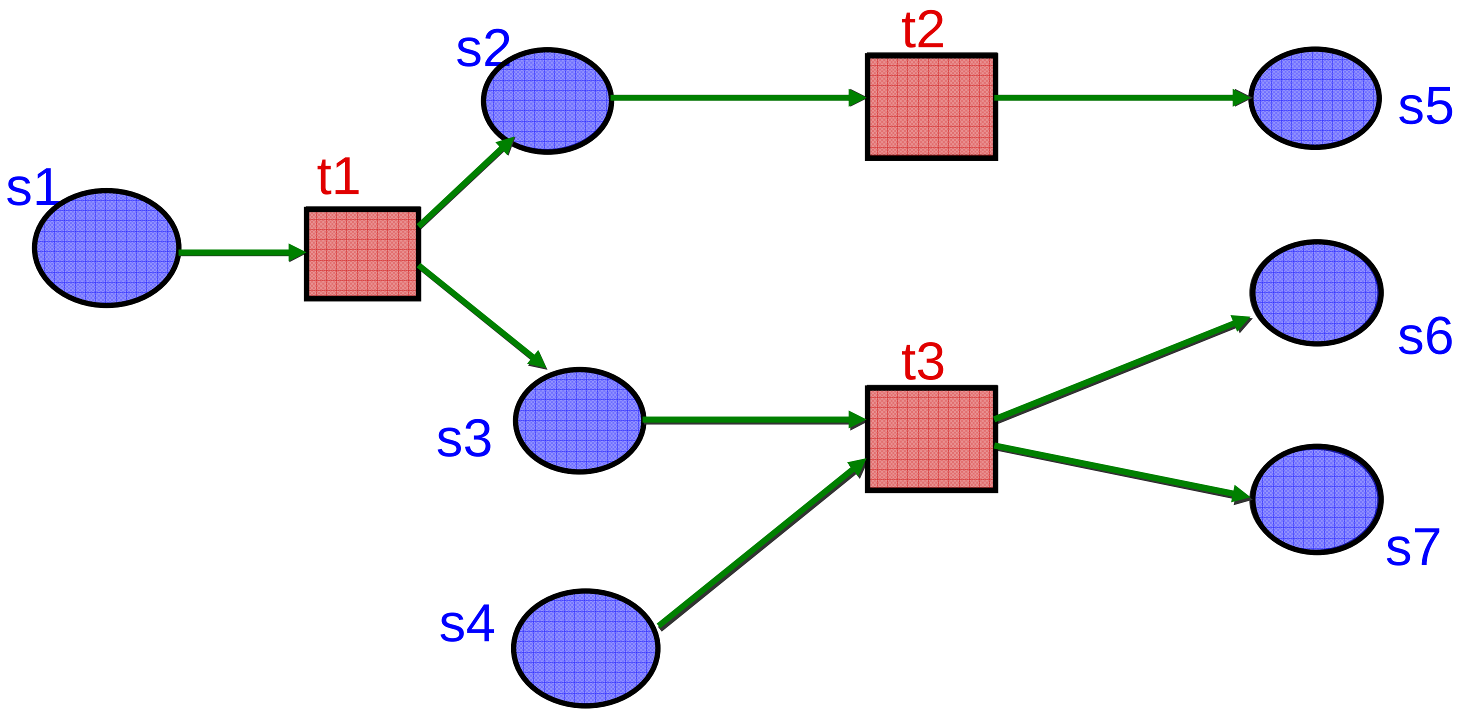
aber nie:  oder 

- Alle Stellen und Transitionen eines Netzes heißen Netzelemente
- Ein Petrinetz ist ein bipartiter Graph

WS06/07

Petrinetze

Beispiel:



S = {s1,s2,s3,s4,s5,s6,s7}

T = {t1,t2,t3}

F = {(s1,t1), (t1, s2), (t1,s3), (s2, t2), (t2, s5), (s3,t3), (s4, t3), (t3, s6), (t3, s7)}

7.3.4 Weitere Notationen

Vorbereich eines Netzelementes x

$\bullet x := \{y \mid (y,x) \in F\}$

Nachbereich eines Netzelementes x

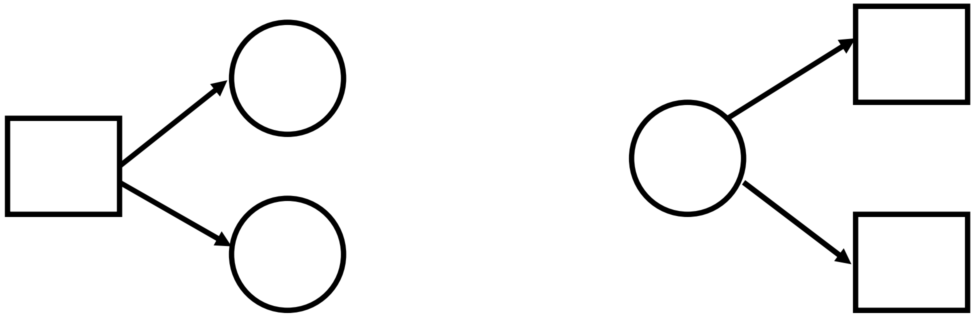
$x\bullet := \{y \mid (x,y) \in F\}$

7.3.4 Weitere Notationen

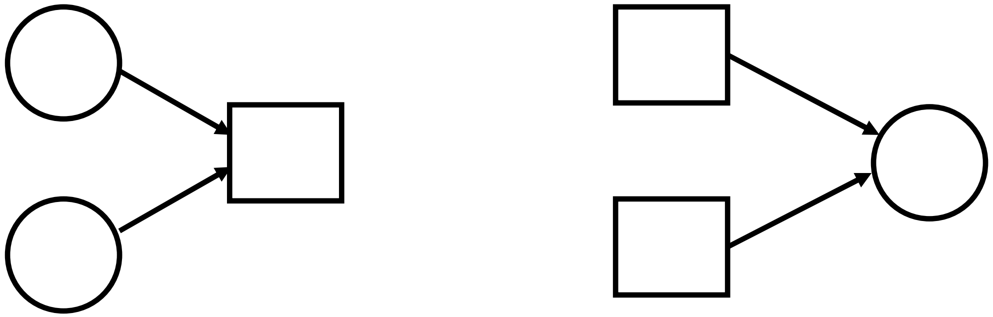
Definition: Verzweigungen

Ein Knoten x heißt

"vorwärtsverzweigt", falls |x•|> 1



"rückwärtsverzweigt", falls |•x|> 1



7.3.4 Weitere Notationen

Definition: Zusammenhängendes Netz

Ein Netz N = (S,T,F) heißt **nicht** zusammenhängend,
wenn **eine Zerlegung**

- **$S \cup T = X_1 \cup X_2$**
  ($X_1 = S_1 \cup T_1$ und $X_2 = S_2 \cup T_2$)
- **$X_1, X_2 \neq \emptyset$**
- **$X_1 \cap X_2 = \emptyset$**

existiert, so dass

- **$F \subseteq (X_1 \times X_1) \cup (X_2 \times X_2)$**
  (d.h. es gibt keine Kanten zwischen Elementen von X_1 und X_2 in N)

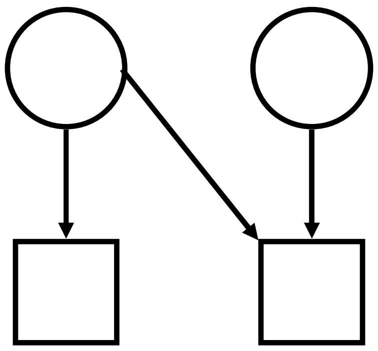
Wenn es keine solche Zerlegung gibt,

dann heißt ein Netz N = (S,T,F) **zusammenhängend**

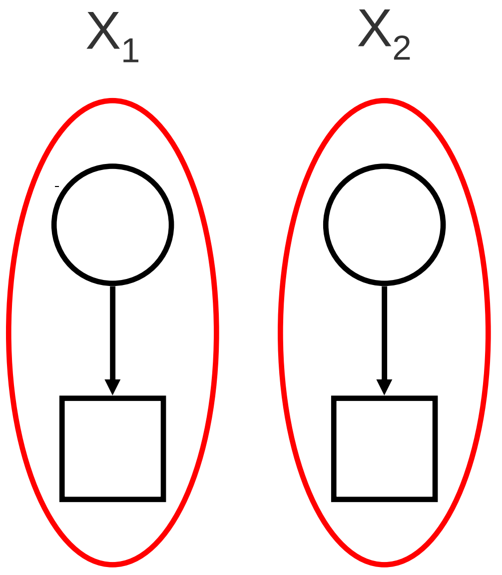
Andere Vorgehensweise: N ist zusammenhängend, wenn $\forall x,y \in (S \cup T)$ ein ungerichteter Pfad mit Anfangspunkt x und Endpunkt y existiert.

7.3.4 Weitere Notationen

Beispiel



zusammenhängend



nicht zusammenhängend

7.3.4 Weitere Notationen

Definition: Teilnetz

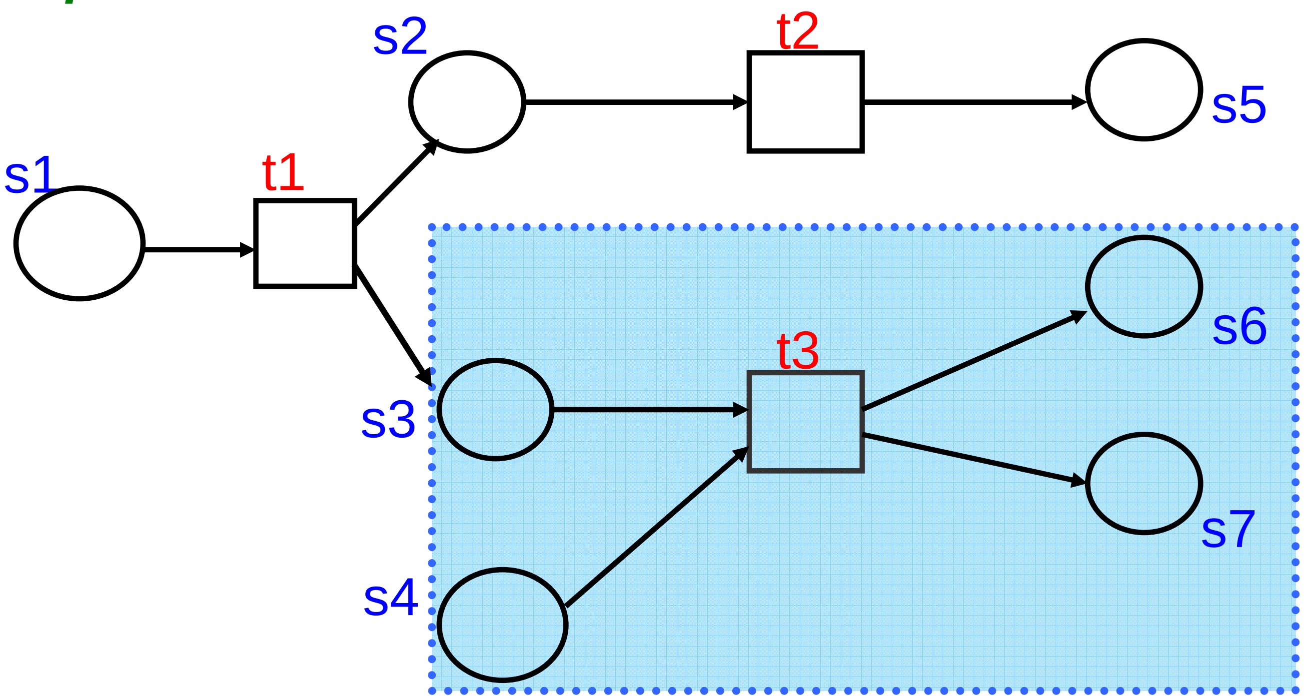
Sei N = (S, T, F) ein Petrinetz.

Ein Netz N' = (S', T', F') heißt **Teilnetz** von N, wenn

- **$S' \subseteq S$**
- **$T' \subseteq T$**
- **$F' = F \cap ((S' \times T') \cup (T' \times S'))$**

7.3.4 Weitere Notationen

Beispiel:



Ein Teilnetz N' eines Netzes N = (S, T, F) wird "generiert" durch die Menge seiner Elemente X:

$$N' = (S \cap X,\ T \cap X,\ F \cap (X \times X))$$

7.3.4 Weitere Notationen

Definition: Rand

Der **Rand eines Teilnetzes N'** (bzgl. des Gesamtnetzes N) ist definiert durch:

$$\{ x \in S' \cup T' \mid (\bullet x \cup x\bullet) \setminus (S' \cup T') \neq \emptyset \}$$

wobei •x, x• sich auf N bezieht.

(Der Rand besteht also aus allen Elementen des Teilnetzes, die eine Kante zum restlichen Gesamtnetz haben)

Ein Teilnetz N' heißt

"stellenberandet",
wenn sein Rand nur Stellen enthält

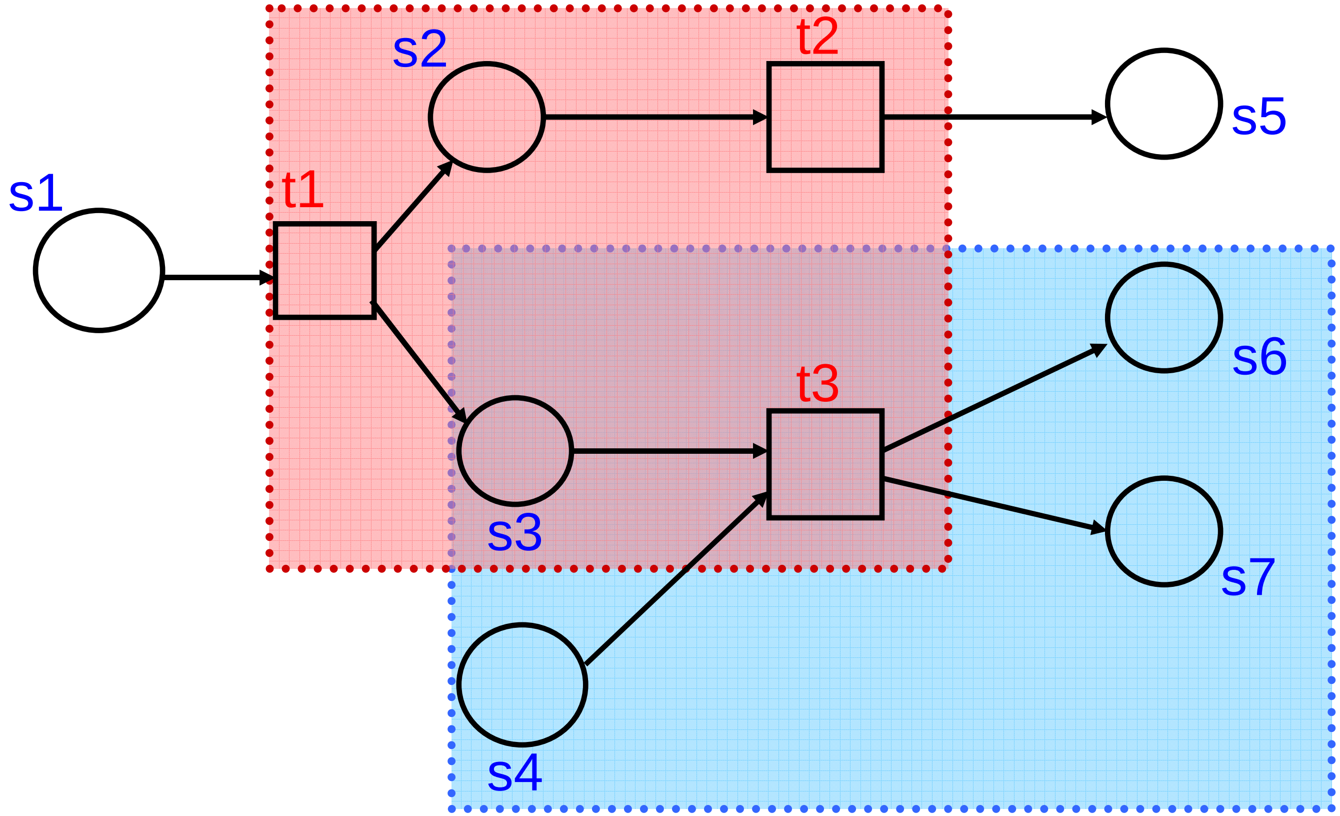
"transitionsberandet",
wenn sein Rand nur Transitionen enthält

7.3.4 Weitere Notationen

Beispiel:

(a) stellenberandetes Teilnetz (**Rand** = {s3})

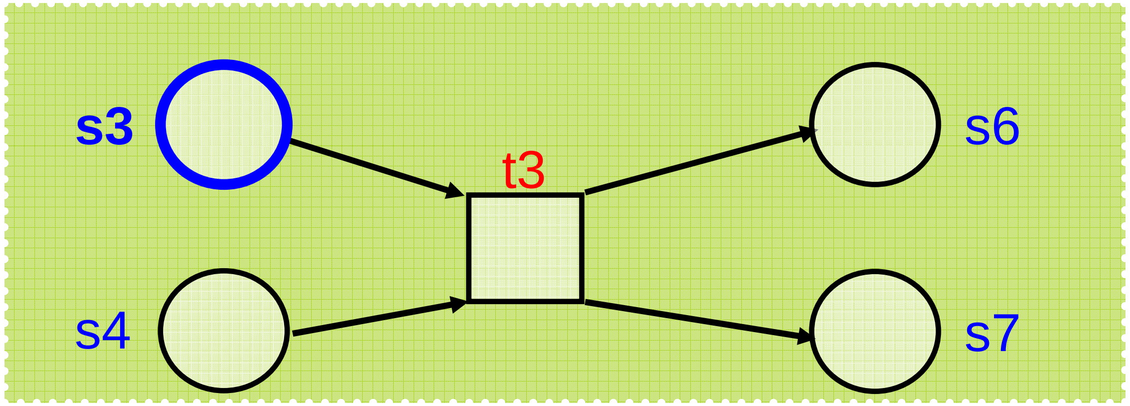
(b) transitionsberandetes Teilnetz (**Rand** = {t1,t2,t3})



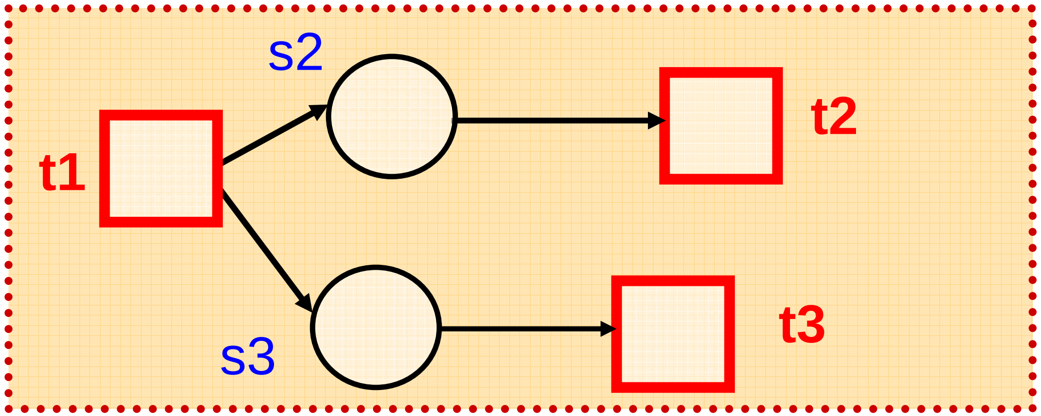
7.3.4 Weitere Notationen

Noch einmal:

(a) **stellen**berandetes Teilnetz (Rand = {s3})



(b) **transitions**berandetes Teilnetz (Rand {t1,t2,t3})



VIKAR
(Bibliothek)

7 Petrinetze

7.4 Netztransformationen

Definitionen

Veränderung des Detaillierungsgrades von Netzen.
Dies spielt eine Rolle bei der schrittweisen Systembeschreibung mit Netzen (siehe auch Beispiel: Bibliothek).

- **Vergröberung:** Ersetzung eines zusammenhängenden transitions- bzw. stellenberandeten **Teilnetzes** durch eine **Transition bzw. Stelle**.

  transisitionsber. Teilnetz $\rightarrow$ Transition
  stellenber. Teilnetz $\rightarrow$ Stelle

- **Verfeinerung:** Umkehrung der Vergröberung

7.4 Netztransformationen

Der Detaillierungsgrad eines Netzes kann auch durch **Netztransformationen** verändert werden:

- **Einbettung**: Erweiterung eines Netzes durch Hinzufügen von Kanten und Knoten, so dass das neue Netz das ursprüngliche als Teilnetz enthält.
- **Restriktion**: Umkehrung der Einbettung
- **Faltung**: Gleichartige Teil-Netze werden „aufeinandergefaltet“, wobei Knoten nur auf solche gleichen Typs gefaltet werden und die Flussrelation gewahrt bleibt.
- **Entfaltung**: Umkehrung der Faltung
- **Komposition**: Netze werden zusammengesetzt durch Verschmelzung von festgelegten Schnittstellen-Elementen des gleichen Typs.
- **Dekomposition**: Umkehrung der Komposition

7 Petrinetze

7.5.1 Grundlegende Definitionen

Einige Netztransformationen können durch sogenannte **"Netzmorphismen"** formalisiert werden

Definition: Netzmorphismus

Seien N = (S, T, F) und N' = (S', T', F') Netze.

Ein **Netzmorphismus** $\boldsymbol{\varphi : N \rightarrow N'}$ ist eine Abbildung

$$\boldsymbol{\varphi : (S \cup T) \rightarrow (S' \cup T')}$$

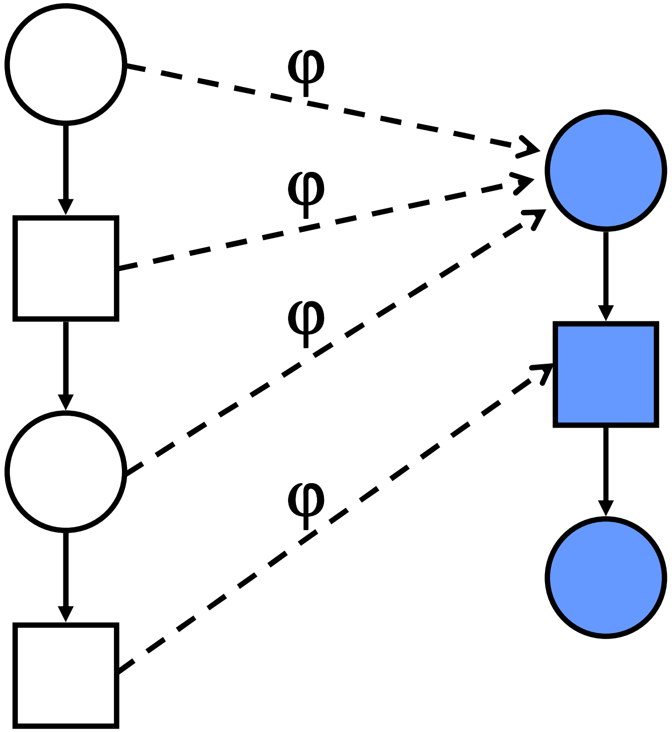
mit folgenden (strukturerhaltenden) Eigenschaften:

- Falls $\boldsymbol{(s, t) \in F \cap (S \times T)}$,
  dann ist entweder $\boldsymbol{(\varphi(s), \varphi(t)) \in F' \cap (S' \times T')}$
  oder $\boldsymbol{\varphi(s) = \varphi(t)}$
- Falls $\boldsymbol{(t, s) \in F \cap (T \times S)}$,
  dann ist entweder $\boldsymbol{(\varphi(t), \varphi(s)) \in F' \cap (T' \times S')}$
  oder $\boldsymbol{\varphi(t) = \varphi(s)}$

d.h. zu jeder Kante im Ursprung gibt es eine Entsprechung im Bild, oder aber die durch die Kante verbundenen Elemente des Ursprungs werden auf das selbe Element des Bilds abgebildet.

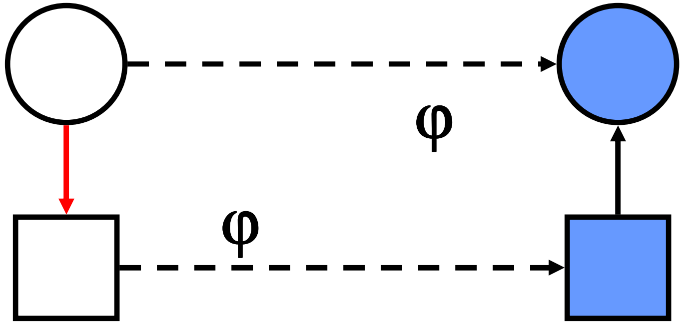
Schreibweise: $\varphi : N \rightarrow N'$

Beispiel 9-6: Netzmorphismus

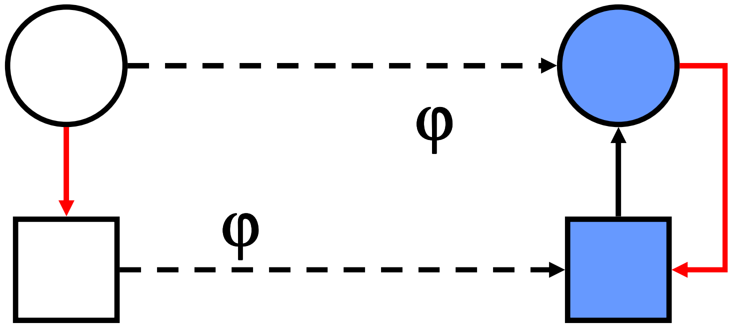


7.5.1 Grundlegende Definitionen

Beispiel 9-7: *Abbildung, die keinen Netzmorphismus darstellt.*

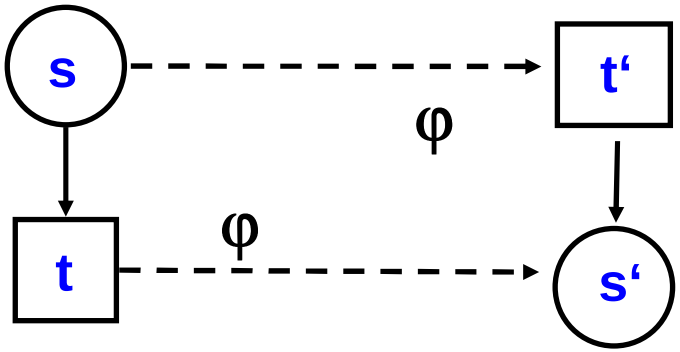


Abbildung, die einen Netzmorphismus darstellt.



7.5.1 Grundlegende Definitionen

Die folgende Abbildung ist wieder kein Netzmorphismus.



Sei φ Netzmorphismus:

- Wenn φ(s) = t', dann auch φ(t) = t'.
- Wenn φ(t) = s', dann auch φ(s) = s'.

Lemma

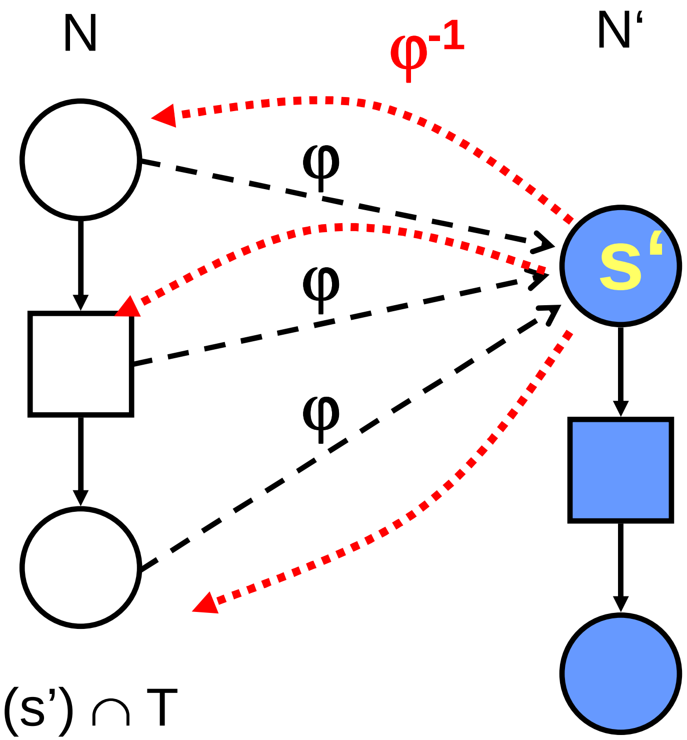
Sei φ: N $\rightarrow$ N' ein **Netzmorphismus**, sei s' eine Stelle von N'.

Dann gilt für das Urbild

$$\varphi^{-1}(s') := \{ x \in S \cup T \mid \varphi(x) = s' \}$$

von s': $\varphi^{-1}(s')$ ist leer oder

$\varphi^{-1}(s')$ generiert ein **stellenberandetes** Teilnetz von N,



d.h. genauer:

Wenn $S^{\sim} := \varphi^{-1}(s') \cap S$ und $T^{\sim} := \varphi^{-1}(s') \cap T$

dann gilt:

1. $F^{\sim} := F \cap (S^{\sim} \times T^{\sim} \cup T^{\sim} \times S^{\sim}) = \{(x,y) \in F \mid \varphi(x) = \varphi(y) = s'\}$
2. $(S^{\sim}, T^{\sim}, F^{\sim})$ ist ein **stellenberandetes** Teilnetz von N.

7.5.1 Grundlegende Definitionen

Beweis:

Annahme $\varphi^{-1}(s') \neq \emptyset$

Entweder: $\varphi^{-1}(s') \cap T = \emptyset$ => Rand nur Stellen

Oder : $\varphi^{-1}(s') \cap T \neq \emptyset$;

Sei $t \in \varphi^{-1}(s') \cap T$

zu zeigen: t gehört nicht zum Rand

Annahme: t gehört zum Rand

Dann: $\exists\, s \in \bullet t \cup t\bullet,\ s \notin \varphi^{-1}(s')$

Sei $s \in \bullet t$:

Entweder: $\varphi(s) = \varphi(t)$ Widerspruch

$$\left.\begin{array}{l} \text{Oder : } (\varphi(s), \varphi(t)) \in F' \text{ und } \varphi(t) \in T' \\ \varphi(t) = s' \in S' \end{array}\right\} \text{Widerspruch}$$

7.5.1 *Grundlegende Definitionen*

Lemma

Sei $\varphi : N \rightarrow N'$ ein **Netzmorphismus**, sei t' eine Transition von N'.

Dann gilt für das Urbild

$\varphi^{-1}(t') := \{x \in S \cup T \mid \varphi(x) = t'\}$ von t':

$\varphi^{-1}(t')$ ist leer oder

$\varphi^{-1}(t')$ generiert ein transitionsberandetes Teilnetz von N,

d.h.

$(\{s \in S \mid \varphi(s) = t'\},$	alle Stellen die auf t' abgebildet werden
$\{t \in T \mid \varphi(t) = t'\},$	alle Transitionen die auf t' abgebildet werden
$\{(x, y) \in F \mid \varphi(x) = \varphi(y) = t'\}$	alle Kanten zwischen Elementen, die auf t' abgebildet werden)

ist ein transitionsberandetes Teilnetz von N.

Beweis: Analog

7.5.2 *Vergröberung und Verfeinerung*

Vergröberungen sind **surjektive Netzmorphismen**, bei denen zusätzlich zu jeder Kante ein Urbild existiert.

Definition: Quotient

Seien N = (S,T,F), N’ = (S’,T’,F’) Netze.
Ein Netzmorphismus $\varphi : N \rightarrow N'$ heißt **Quotient** $:\Leftrightarrow$

- φ ist surjektiv
- $\forall\ (x',y') \in F'\ \ \exists\ (x,y) \in F$: $\varphi(x) = x'$ und $\varphi(y) = y'$

7.5.2 *Vergröberung und Verfeinerung*

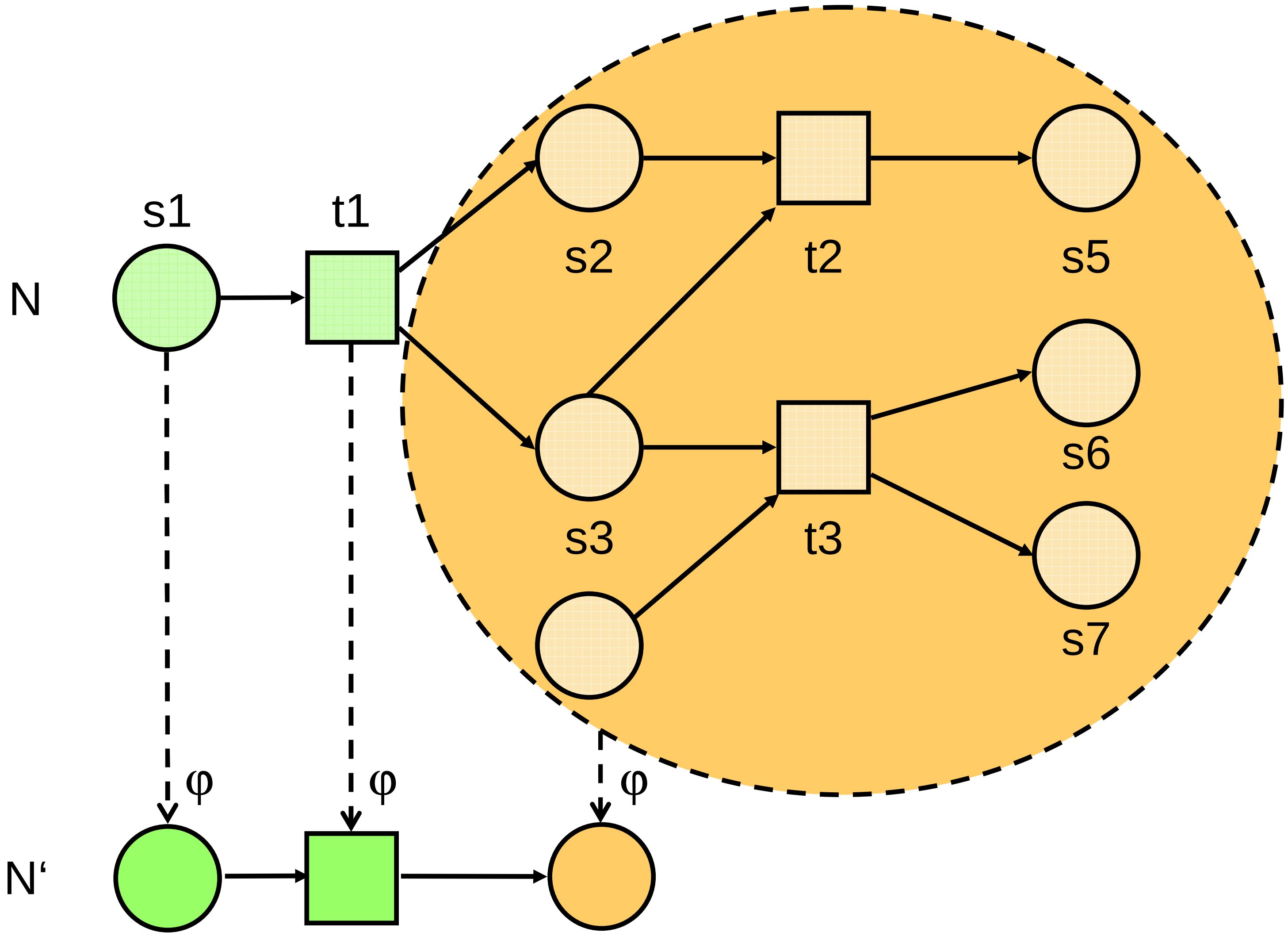
Definition: Vergröberung / Verfeinerung

N' = (S',T',F') ist **Vergröberung** von N = (S,T,F),
wenn ein **Quotient**
$\varphi : N \rightarrow N'$ existiert,
und
$\varphi^{-1}(x')$ für jedes Element $x' \in (S' \cup T')$
ein zusammenhängendes Teilnetz von N generiert.

Umgekehrt heißt N eine **Verfeinerung** von N'.

7.5.2 Vergröberung und Verfeinerung

Beispiel: Vergröberung / Verfeinerung



7.5.3 Einbettung

Eine **Einbettung** eines Netzes als Modell eines Ausschnitts eines realen Systems bedeutet:

- Ergänzung von weiteren Aspekten / Systemkomponenten,
- Vervollständigung,
- Einbeziehung der Umwelt.

7.5.3 Einbettung

Definition: Einbettung

Seien $N = (S,T,F)$, $N' = (S',T',F')$ Netze.
Ein Netzmorphismus $\varphi : N \rightarrow N'$ heißt **Einbettung** $:\Leftrightarrow$

- $\varphi: S \cup T \rightarrow S' \cup T'$ ist injektiv,
- $\forall (x',y') \in F',\ x', y' \in \varphi(S \cup T)\ \exists\ (x,y) \in F$:
  $\varphi(x) = x'$ und $\varphi(y) = y'$

Lemma

Für jedes Teilnetz N eines Netzes N' ist der **Netzmorphismus**

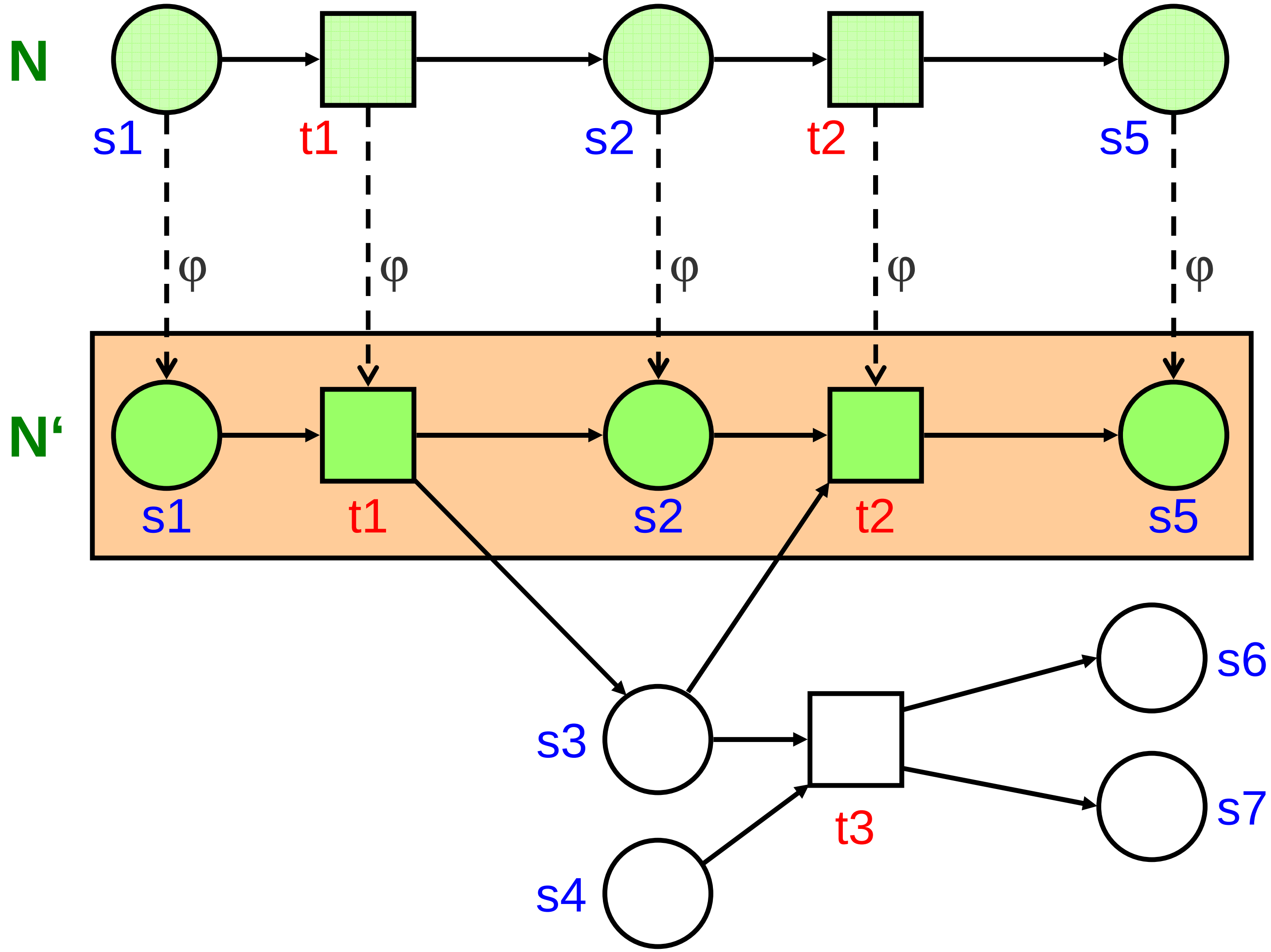
$\varphi : N \rightarrow N'$, $\varphi(x)=x$ für alle $x \in S \cup T$

eine Einbettung.

Dies muss nicht die einzige mögliche Einbettung sein**!**

7.5.3 Einbettung

Beispiel: Einbettung



7.5.4 Faltung

Eine **Faltung** ist ein **Quotient**

(d.h. surjektiv, jede Kante hat Urbild),

bei der Stellen auf Stellen und

Transitionen auf Transitionen abgebildet werden.

Definition: Faltung

Seien N = (S,T,F), N' = (S', T', F') Netze.

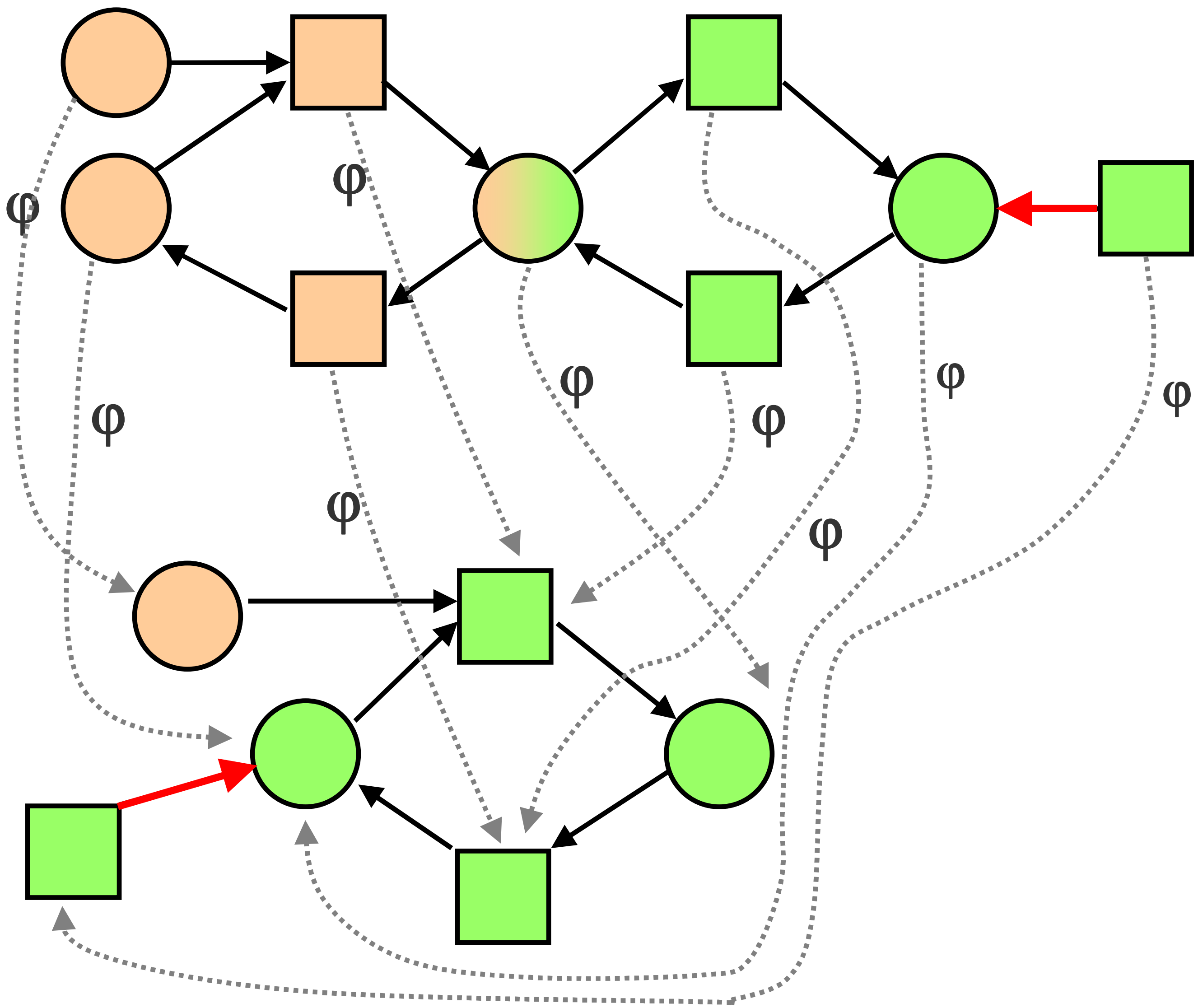
Eine Faltung ist ein **Quotient**

φ: N $\rightarrow$ N'

mit φ(S) = S', φ(T) = T'

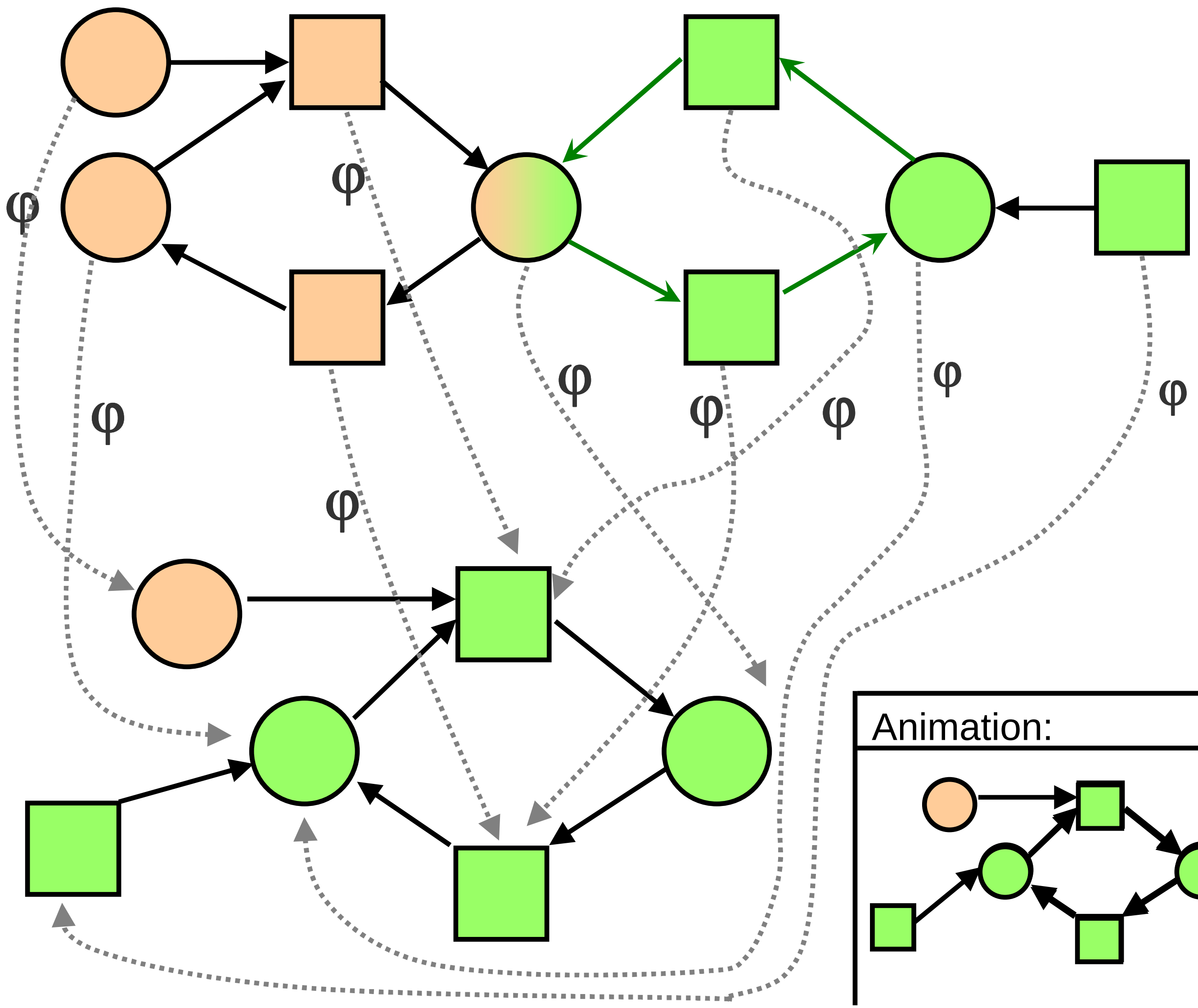
7.5.4 Faltung

Beispiel 1: Faltung

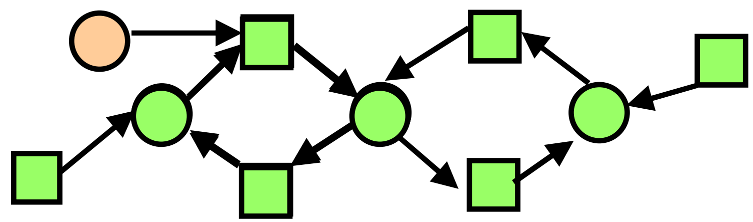


7.5.4 Faltung

Beispiel 2: Faltung



Animation:



7.5.5 Komposition

Netze werden zusammengesetzt durch **Verschmelzung** von festgelegten Schnittstellen-Elementen des gleichen Typs

VIKAR (Stellenkomposition)

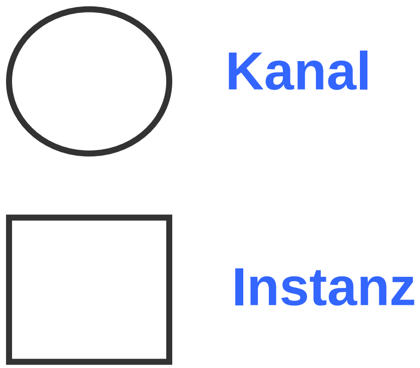
VIKAR (Transitionenkomposition)

Formalisiert werden Kompositionen durch spezielle Faltungen, die für jedes Teilnetz **injektiv** sind.

Zu jeder Komponente existiert eine **Einbettung** in das zusammengesetzte Netz.

7 Petrinetze

7.6 Kanal/Instanzen-Netze

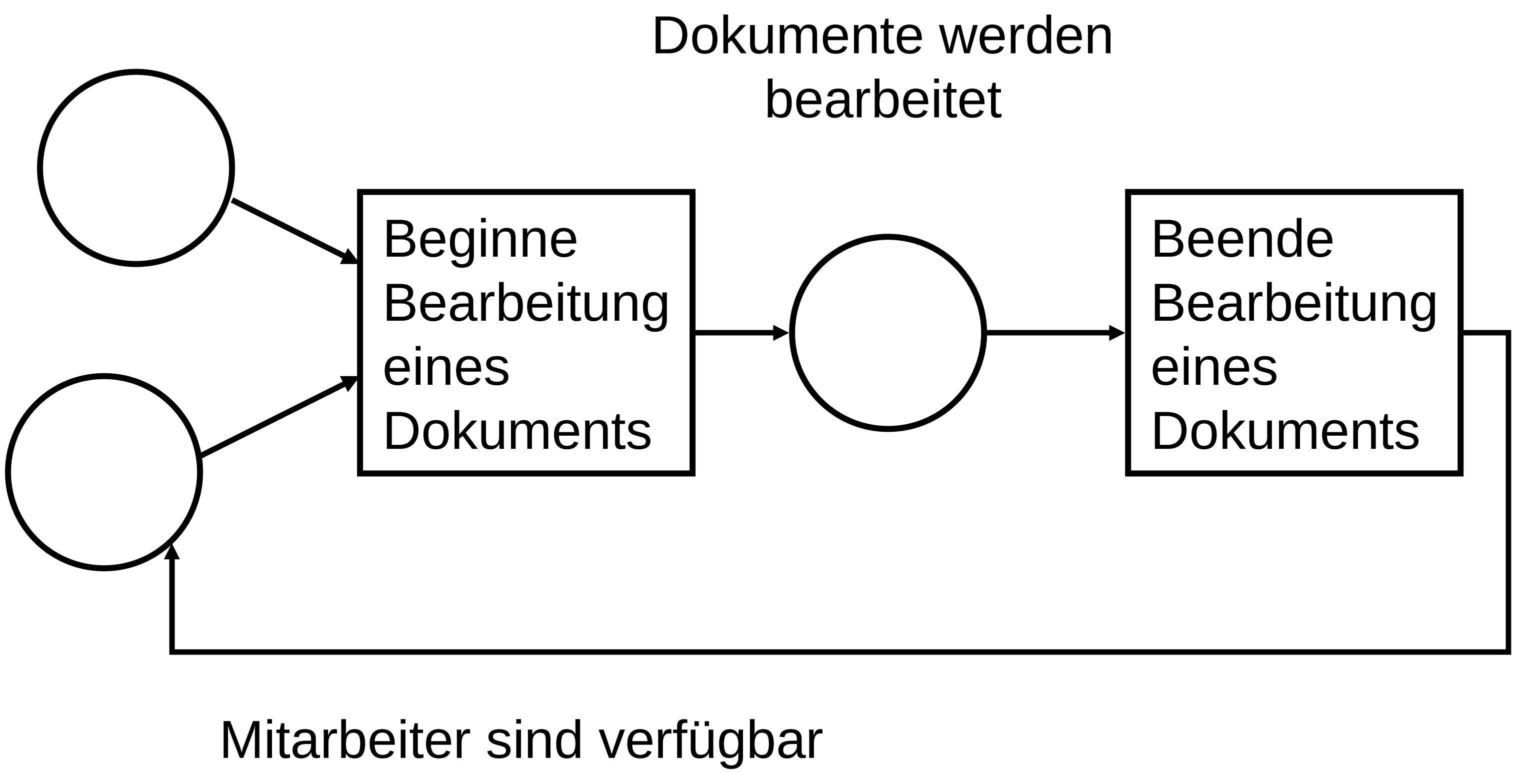


- umgangssprachliche und allgemeinverständliche Beschriftung der Netzelemente
(soll ihnen eine konkrete Bedeutung geben).
- keine Marken!

7.6 Kanal/Instanzen-Netze

Beispiel:

Dokumente sind zu bearbeiten



7.6 Kanal/Instanzen-Netze

Zweck

Durch Verfeinerung und Einbettung soll ein detailliertes Modell entwickelt werden.

Übergang (ab einer bestimmten Detaillierungstiefe)

- Transitionen (Instanzen) $\rightarrow$ Systemfunktionen
- Stellen (Kanäle) $\rightarrow$ Datentypen

Danach Übergang

- zur Verhaltensmodellierung
- zu höheren Netzen

Weiterhin erlauben Kanal-Instanzen-Netzen einen Überblick über kausale Strukturen des Systems.

7 Petrinetze

7.7.1 Definitionen

Stellen/Transitions-Netze (S/T-Netze) sind Netze mit

- Anfangsmarkierung m_0
- Schaltregel (bestimmt das Verhalten)

Definition: Markierung

a) Eine **Markierung** m eines Netzes N = (S, T, F) ist eine
Abbildung $m: S \to \mathbb{N}_0$ (mit $\mathbb{N}_0 = \{0,1,2,3,...\}$)

anders ausgedrückt:

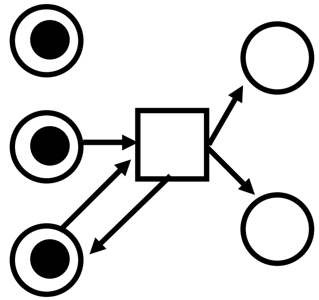
b) Eine **Markierung** m ist ein Vektor $m=(m_1, \ldots, m_n)$,

- wobei $n = |S|$ gilt und
- jedes $m_i \in \mathbb{N}_0$, für $i=1, \ldots, n$
- *D.h. m_i besagt, dass sich in der i-ten Stelle m_i Marken befinden*
- *Zusammenhang zu a) für Stelle s_i: $m(s_i) = m_i$*

7.7.1 Definitionen

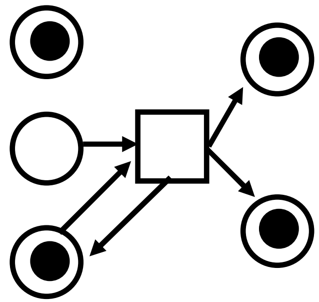
Definition: Schaltregel

Eine Markierung m „aktiviert“ eine Transition t, wenn m(s) > 0 für alle $s \in \bullet t$.



Falls t unter m aktiviert ist, kann t „schalten“; nach Schalten von t: Folgemarkierung m‘:

$$m'(s) = \begin{cases} m(s), & \text{falls } s \notin \bullet t \text{ und } s \notin t\bullet \\ m(s)-1, & \text{falls } s \in \bullet t \text{ und } s \notin t\bullet \\ m(s)+1, & \text{falls } s \notin \bullet t \text{ und } s \in t\bullet \\ m(s), & \text{falls } s \in \bullet t \text{ und } s \in t\bullet \end{cases}$$



Schreibweise: $m \rightarrow^{t} m'$ ("Schaltvorgang")

7.7.1 Definitionen

Ein **S/T-Netz N mit Markierung m**
wird durch das Paar (N, m) angegeben.

Die Anfangsmarkierung eines S/T-Netzes wird meist mit m_0 bezeichnet.

Definition: Schaltfolge

Sei m eine Markierung eines S/T-Netzes N,
$m \rightarrow^{t1} m_1$, $m_1 \rightarrow^{t2} m_2$, ..., $m_{n-1} \rightarrow^{tn} m_n$ Schaltvorgänge,

dann ist $\tau = t_1\, t_2\, t_3 \ldots t_n$
(erst t_1, dann t_2, dann ... dann t_n)

eine **Schaltfolge** von m nach m_n

Schreibweise: $m \rightarrow^{\tau} m_n$.

7.7.1 Definitionen

Wichtige Notationen

- $m \rightarrow^{\varepsilon} m$ — ε ist die leere Schaltfolge
- $m \rightarrow^{*} m'$ — $\exists$ Schaltfolge τ: $m \rightarrow^{\tau} m'$
  m' ist von m aus „erreichbar“
- [m> — Menge aller von m aus erreichbaren Markierungen

7.7.2 Markierungsgraph

- Das Verhalten eines S/T-Netzes (N,m_0) wird durch die Menge seiner Schaltfolgen beschrieben.
- Der Markierungsgraph ist eine kompakte Darstellung dieser Menge.

Definition: Markierungsgraph

Sei N = (S, T, F) ein S/T-Netz mit Anfangsmarkierung m_0

Der **Markierungsgraph** von (N, m_0) besteht aus der

- Knotenmenge $[m_0>$
- Menge beschrifteter Kanten $\{(m, t, m') \mid m \to^t m'\}$

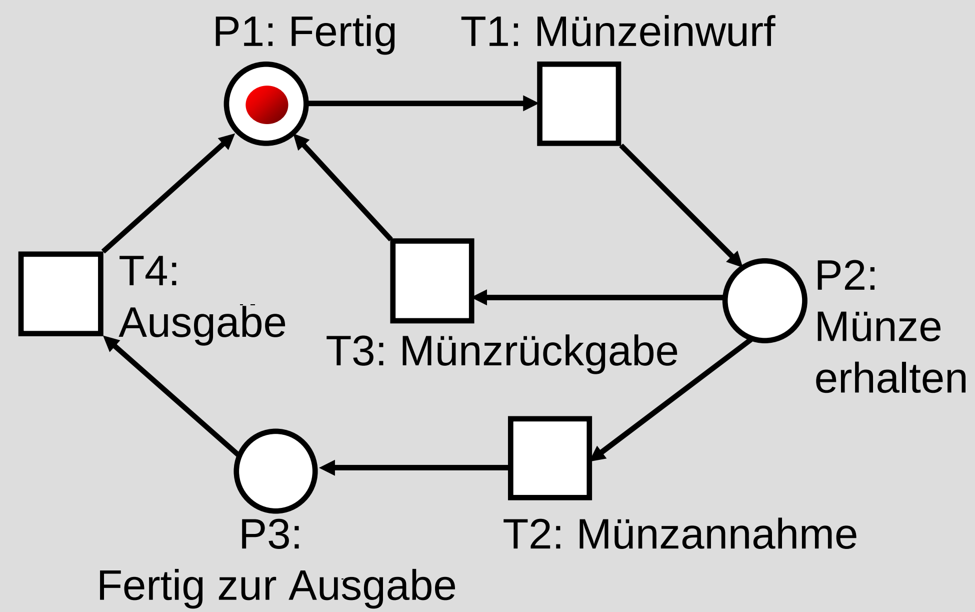
Schreibweise:

- Die Anfangmarkierung wird durch $\to m_0$ dargestellt

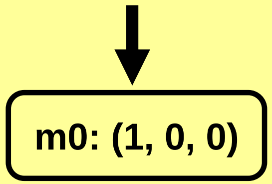
- Wir beschriften die Knoten mit dem Vektor der entspr. Markierung

7.7.2 Markierungsgraph

Geg.: Petrinetz



Zugehöriger Markierungsgraph

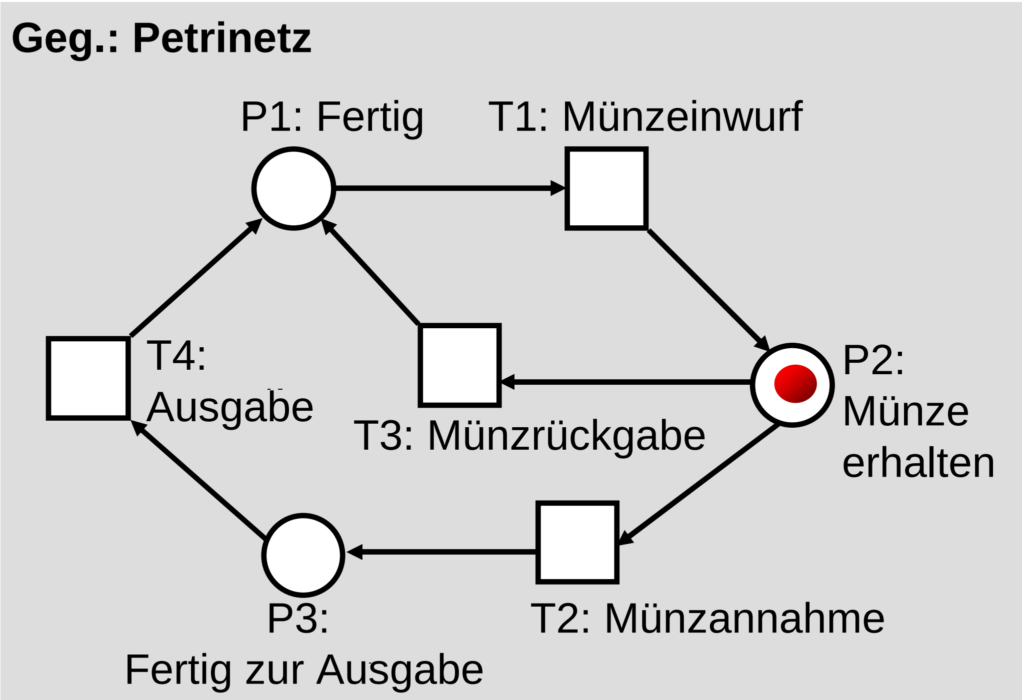


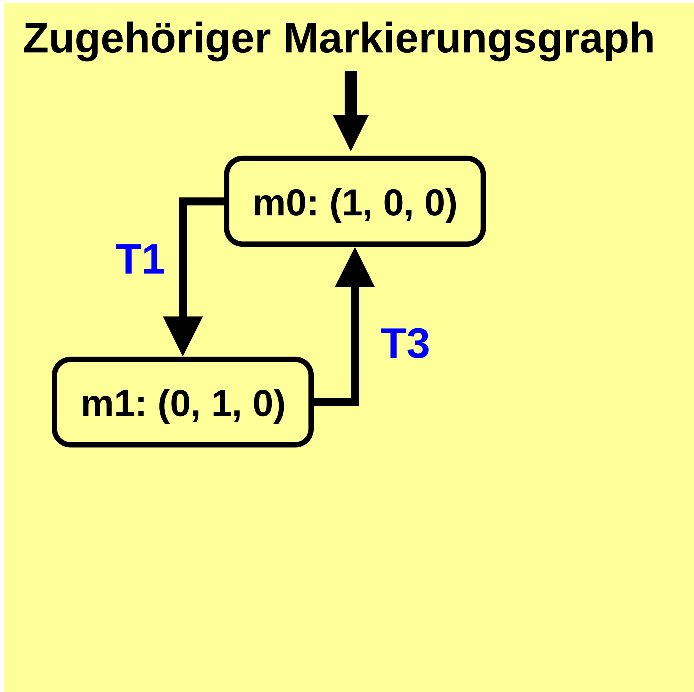
	Stellen			Transitionen				
Markierung	P1	P2	P3	T1	T2	T3	T4	nächste Markierung
m0	1	0	0	⊗				m1

X **Transition feuert**

O **Transition aktiviert**

7.7.2 Markierungsgraph





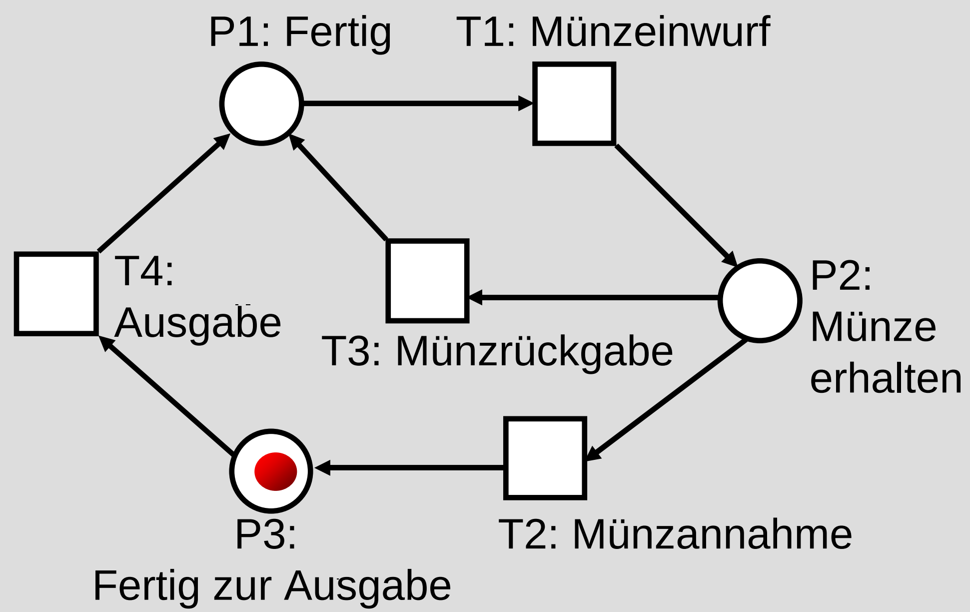
	Stellen			Transitionen				
Markierung	P1	P2	P3	T1	T2	T3	T4	nächste Markierung
m0	1	0	0	⊗				m1
m1	0	1	0		⊗	⊗		m0, m2

X **Transition feuert**

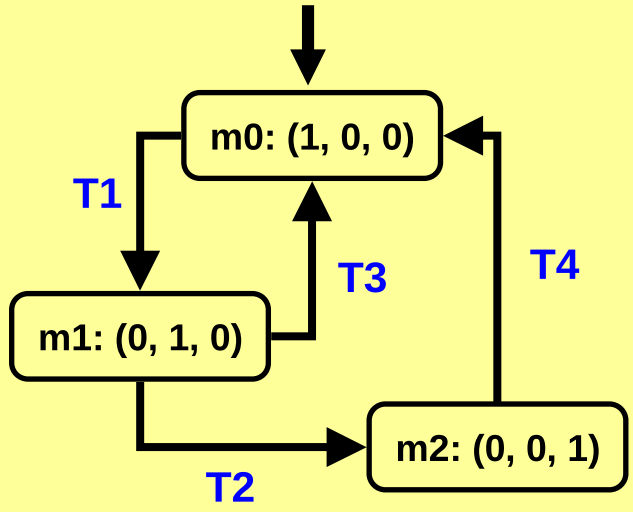
O **Transition aktiviert**

7.7.2 Markierungsgraph

Geg.: Petrinetz



Zugehöriger Markierungsgraph



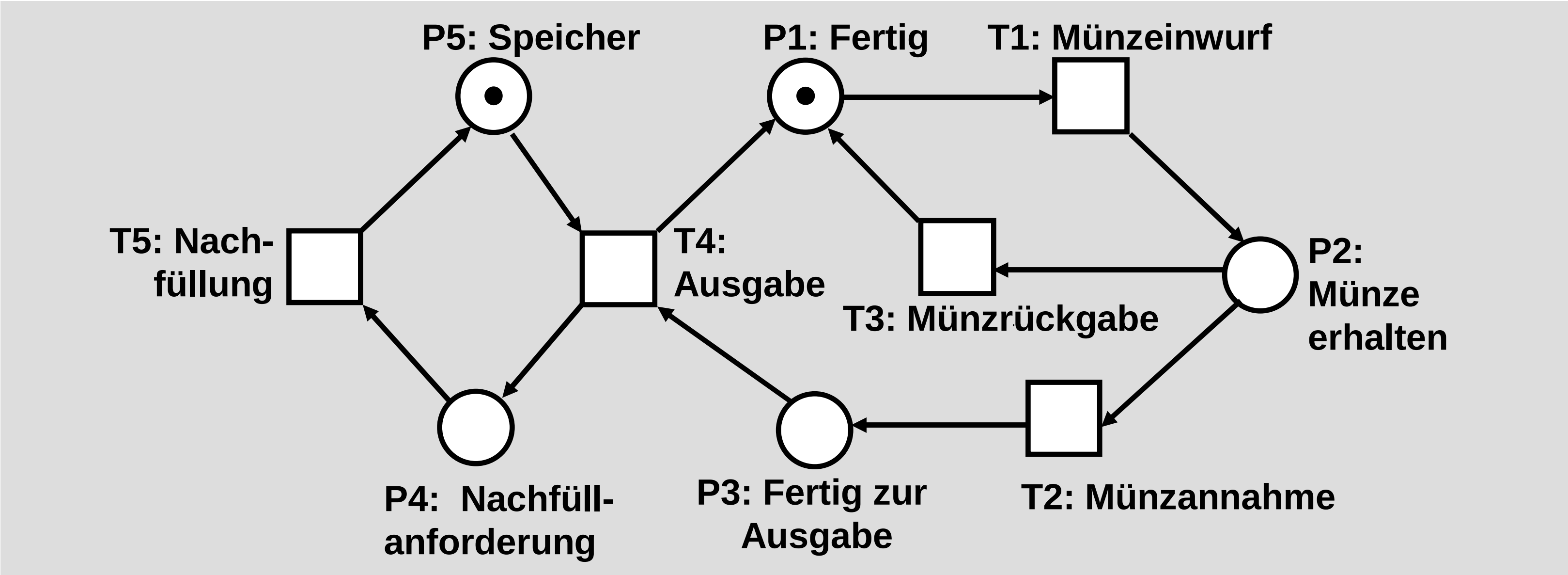
	Stellen			Transitionen				
Markierung	P1	P2	P3	T1	T2	T3	T4	nächste Markierung
m0	1	0	0	⊗				m1
m1	0	1	0		⊗	⊗		m2, m0
m2	0	0	1				⊗	m0

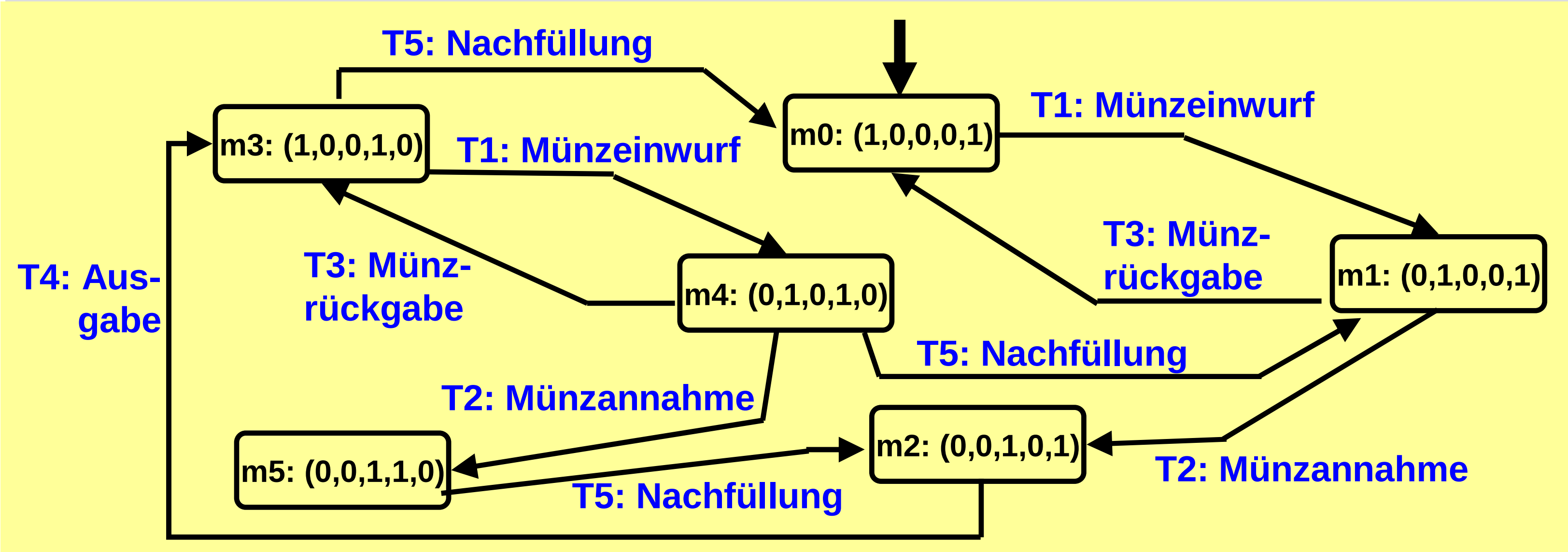
X **Transition feuert**

O **Transition aktiviert**

7.7.2 Markierungsgraph

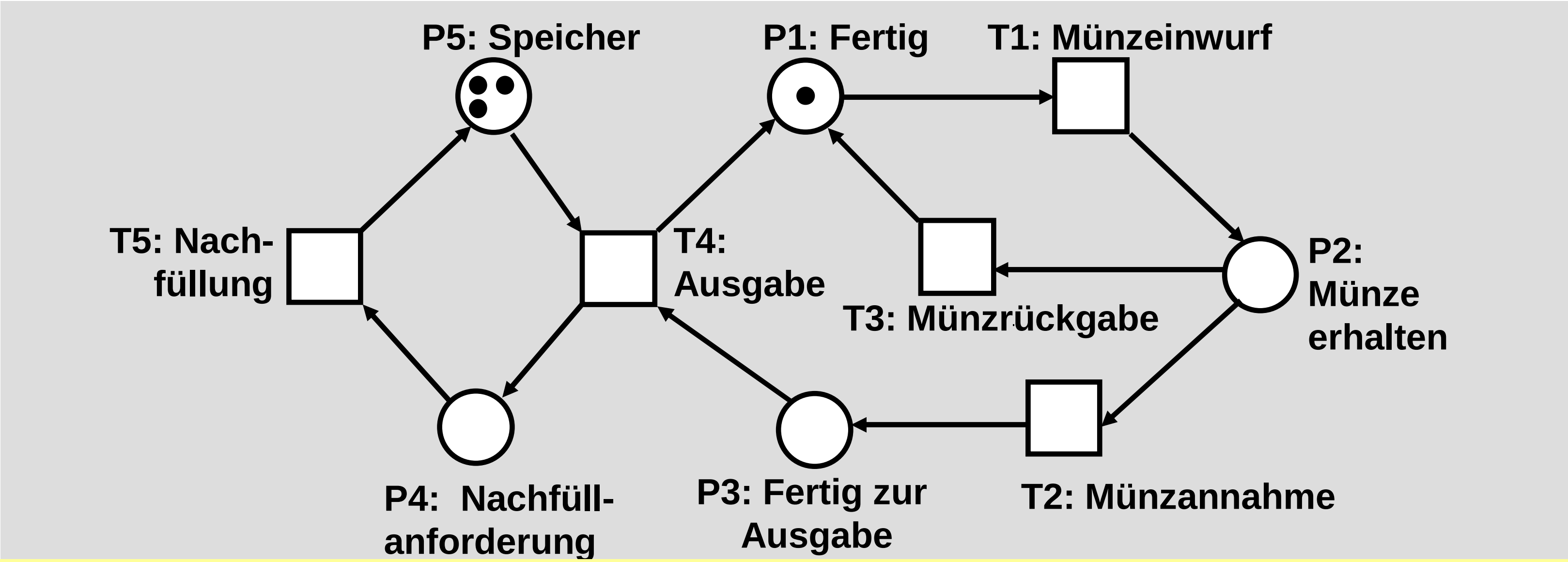
Beispiel:



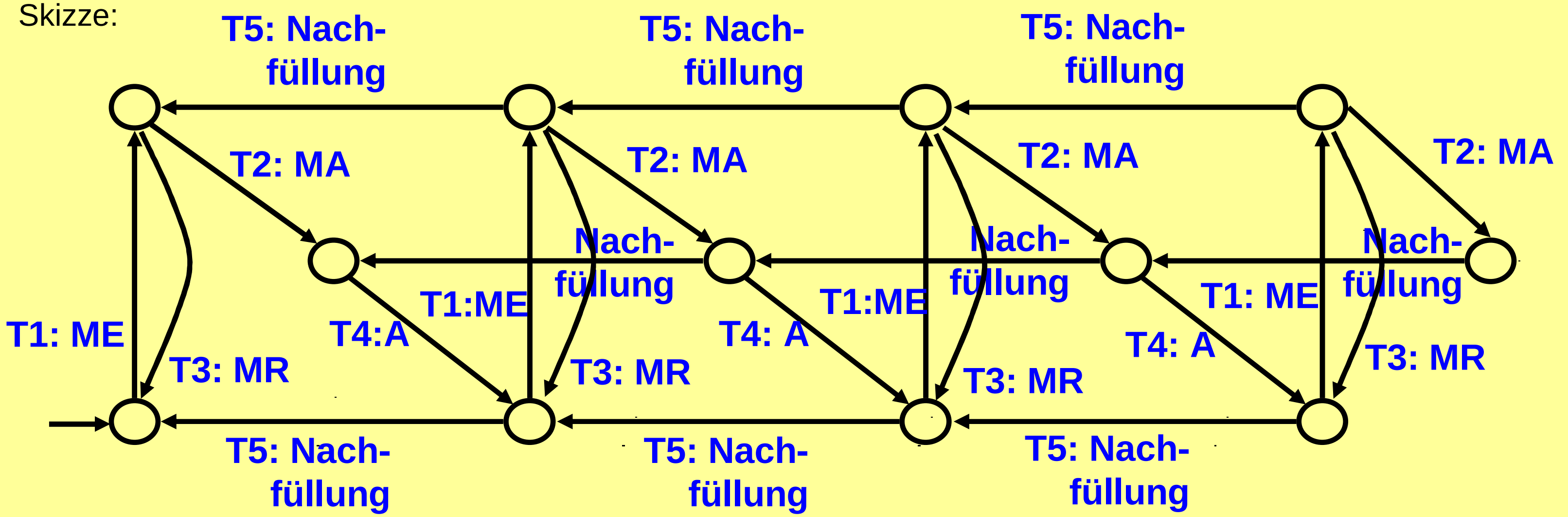


7.7.2 Markierungsgraph

Beispiel:

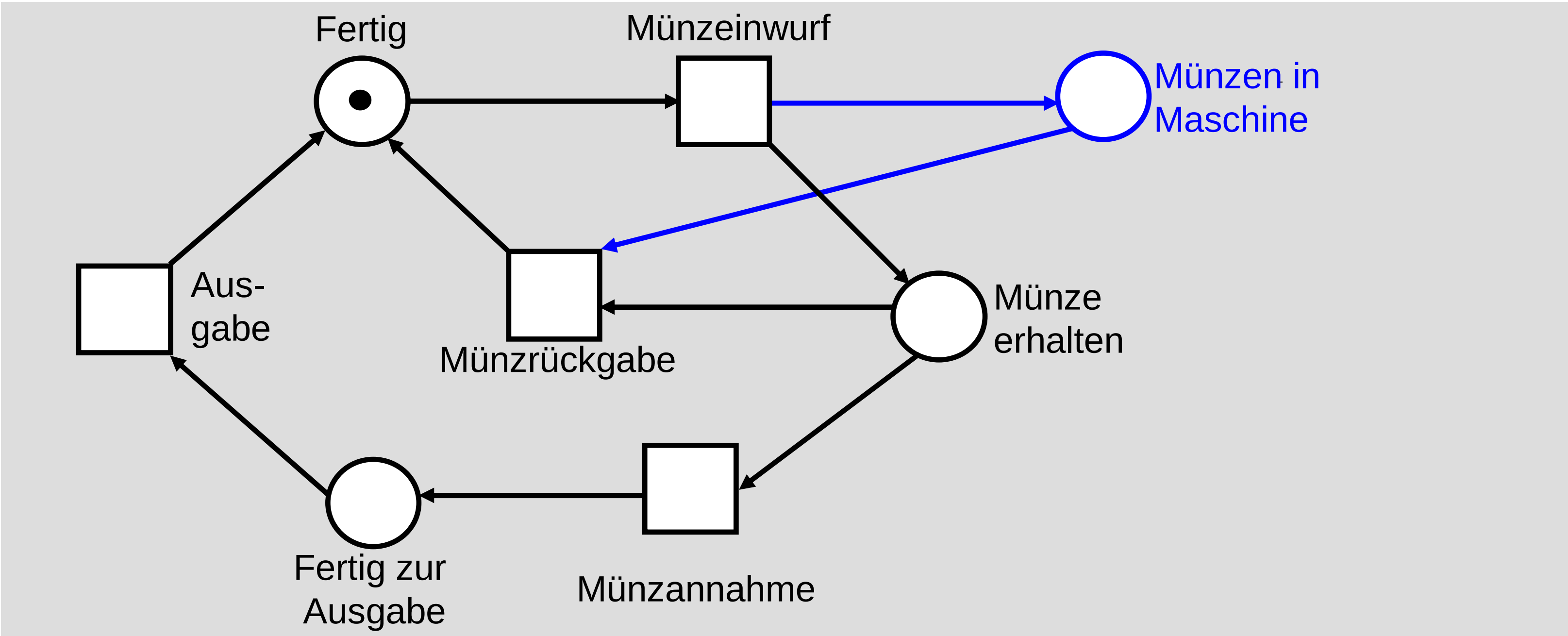


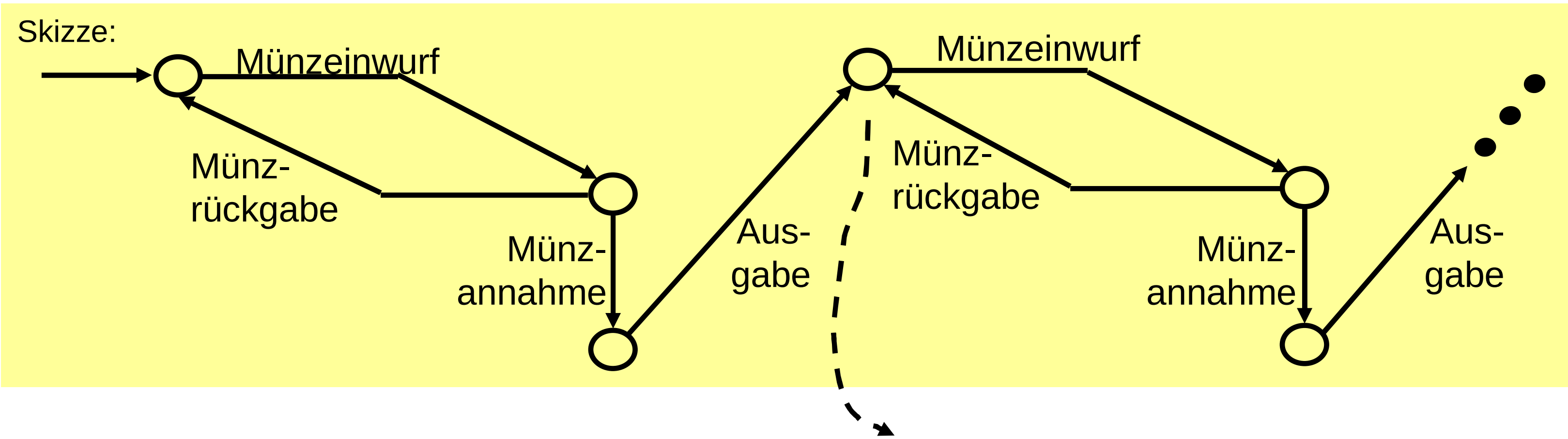
Skizze:



7.7.2 Markierungsgraph

Beispiel:





Ähnliche Ausgangssituation, jedoch +1 in Münzspeicher

Beschränkte Netze

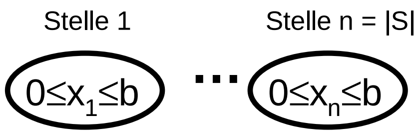
Ein Markierungsgraph ist **endlich** $\Leftrightarrow$
das S/T-Netz ist beschränkt

Definition: Beschränktheit

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

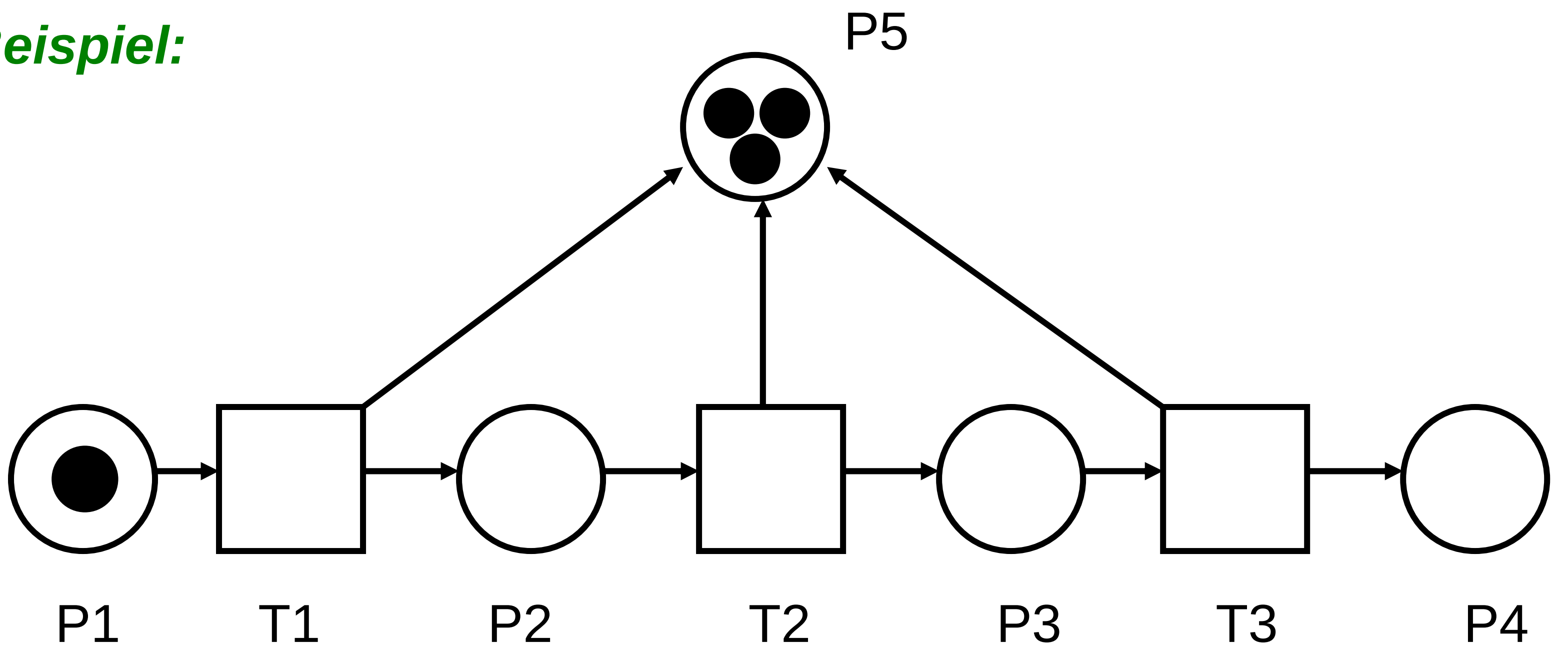
Zusammenhang zwischen Beschränktheit und Anzahl der Markierungen

- Hat (N, m_0) eine Schranke b,
  dann ist $|[m_0>| \leq (b+1)^{|S|}$.
- Ist $|[m_0>| < \infty$,
  dann ist $b := \max_{s \in S}(m_0(s)) + |[m_0>| - 1$
  eine geeignete Schranke



7.7.2 Markierungsgraph

Beispiel:



Max. Anzahl Marken in Stelle P5: $\max_{s \in S}(m_0(s)) = 3 = N$

$[m_0\rangle = \{(1,0,0,0,3), (0,1,0,0,4), (0,0,1,0,5), (0,0,0,1,6)\}$

d.h. $|[m_0\rangle| = 4$

$\Rightarrow$ b = 6 ist eine geeignete Schranke
(in diesem Fall sogar die kleinste mögliche)

Anz. Marken in P5:

N

$\left.\begin{array}{l} +\,1\ \ (m_0 \rightarrow m_1) \\ +\,1\ \ (m_1 \rightarrow m_2) \\ \ldots \end{array}\right\}\ |[m_0\rangle| - 1$

7.7.2 Markierungsgraph

Weitere Eigenschaften von beschränkten S/T-Netzen

- Für b=1 kann jede Stelle als zweiwertige Bedingung (erfüllt / nicht erfüllt) interpretiert werden.

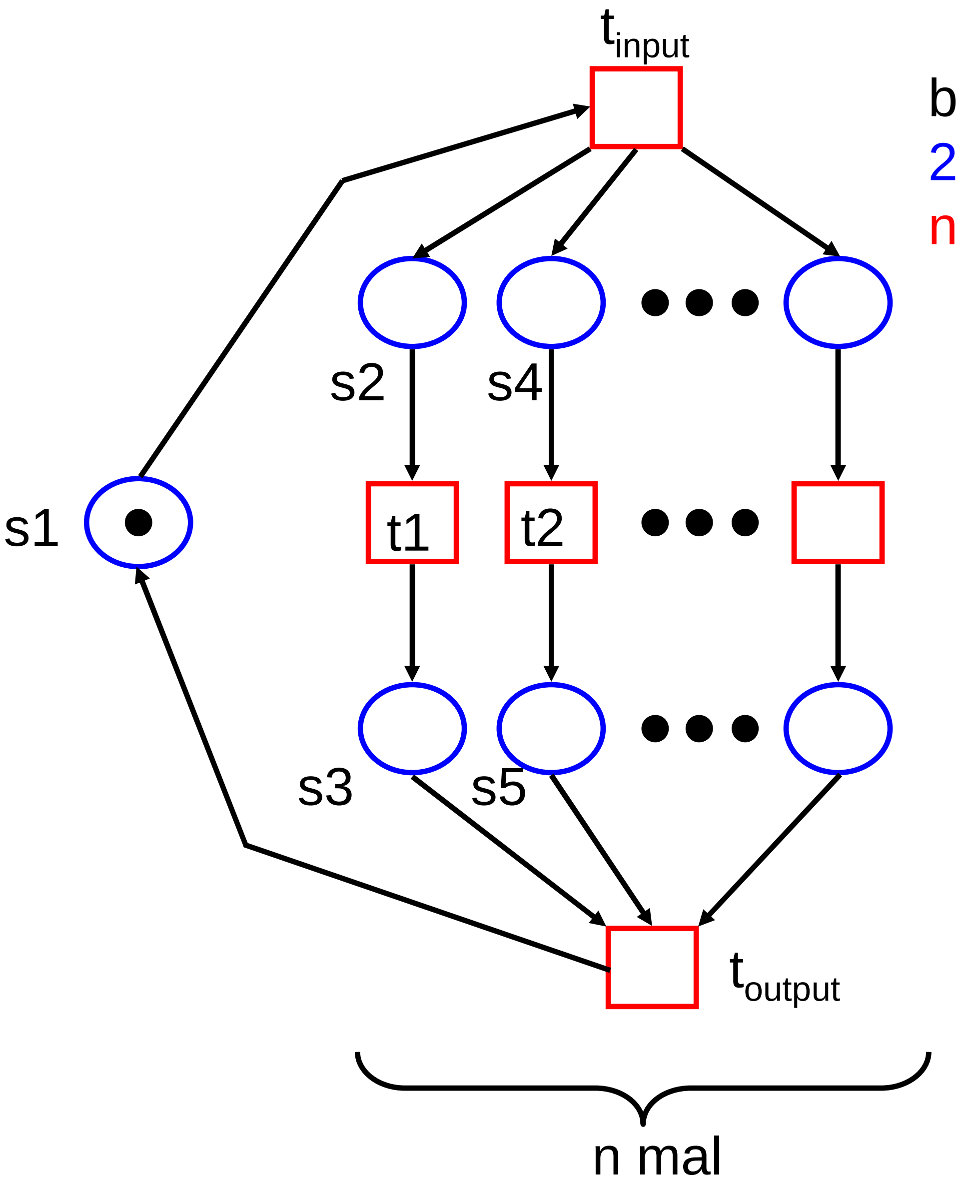
- Auch bei beschränkten S/T-Netzen wächst die Anzahl der erreichbaren Markierungen oft exponentiell mit der Größe des Netzes. (State-Space-Explosion)

  $\Rightarrow$

  Analyse des Verhaltens mittels der Konstruktion des Markierungsgraphen ist deshalb praktisch unmöglich.

7.7.2 Markierungsgraph

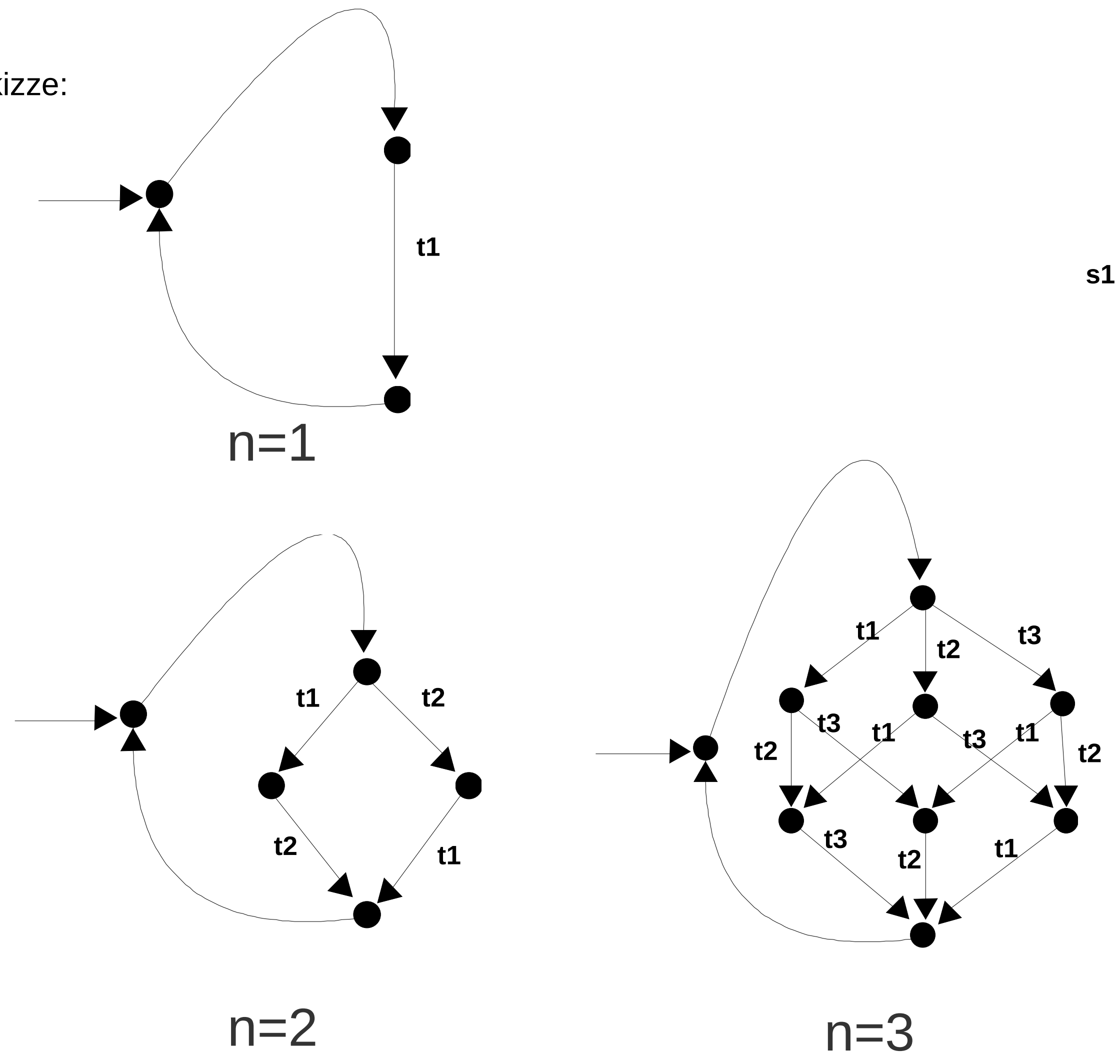
Beispiel:

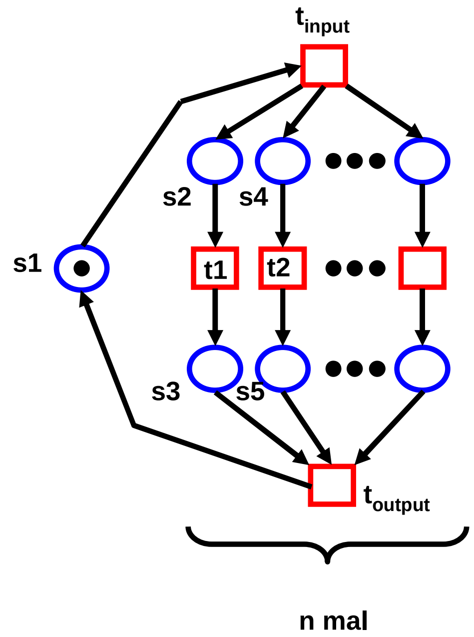


b = 1
2n+1 Stellen,
n+2 Transitionenen

7.7.2 Markierungsgraph

Skizze:





Für n=k hat der Markierungsgraph 2^k+1 Knoten, entsprechend viele Markierungen sind erreichbar.

7.7.3 Verhaltenseigenschaften von S/T-Netzen

Definitionen:

Ein S/T-Netz (N,m_0)

- **terminiert**,
  wenn die Menge der möglichen Schaltfolgen endlich ist.

- ist **verklemmungsfrei**,
  wenn jede erreichbare Markierung eine Transition aktiviert.

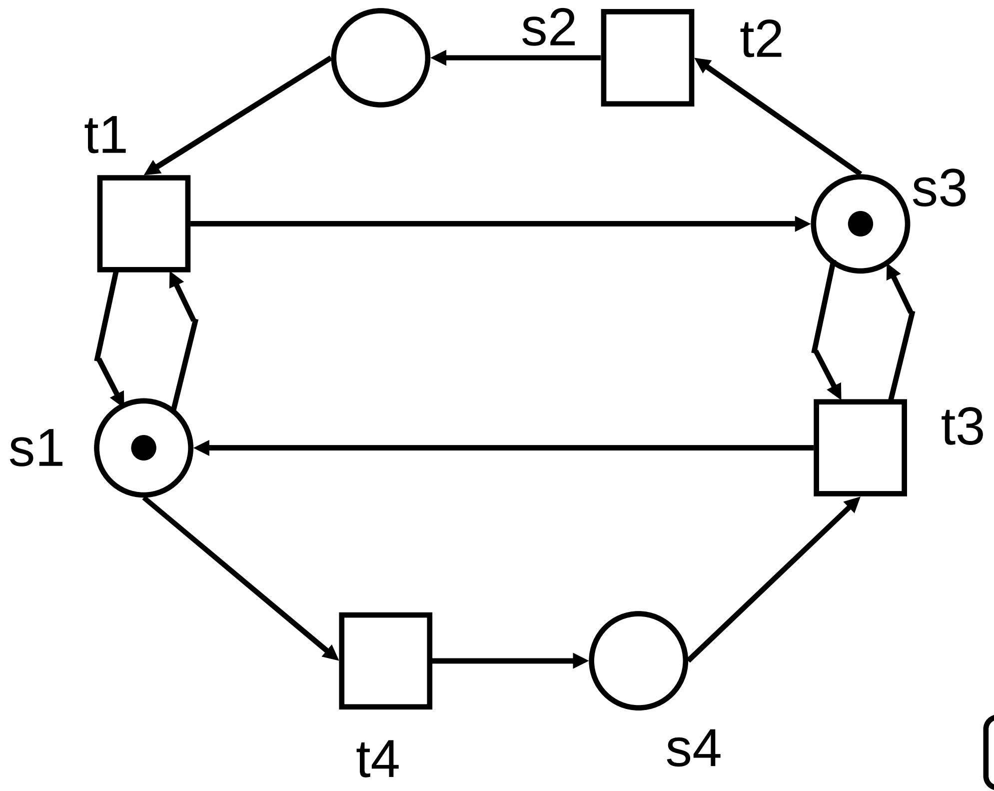
- ist **lebendig**,
  wenn von jeder erreichbaren Markierung ausgehend eine Schaltfolge existiert, die alle Transitionen umfasst.

- ist **rücksetzbar**,
  wenn von jeder erreichbaren Markierung wieder die Anfangsmarkierung m_0 erreichbar ist.

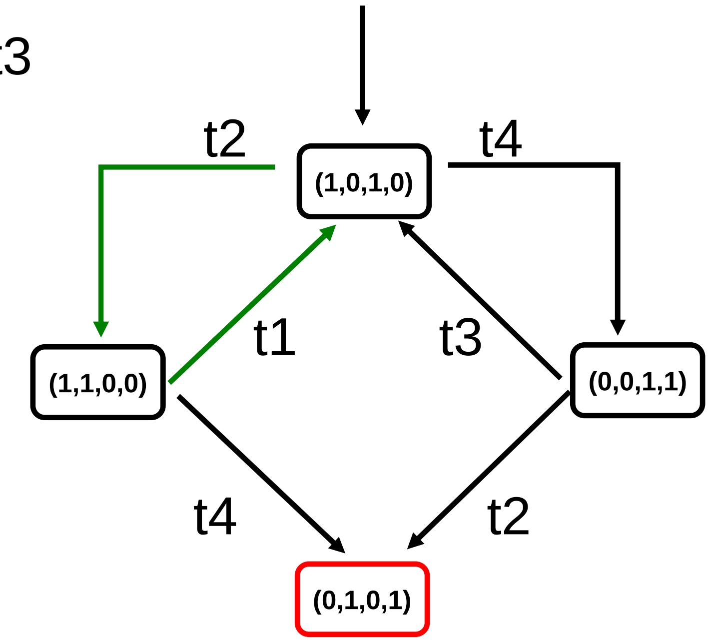
7.7.3 Verhaltenseigenschaften von S/T-Netzen

Beispiel:

Ein nichtterminierendes, nicht verklemmungsfreies S/T-Netz

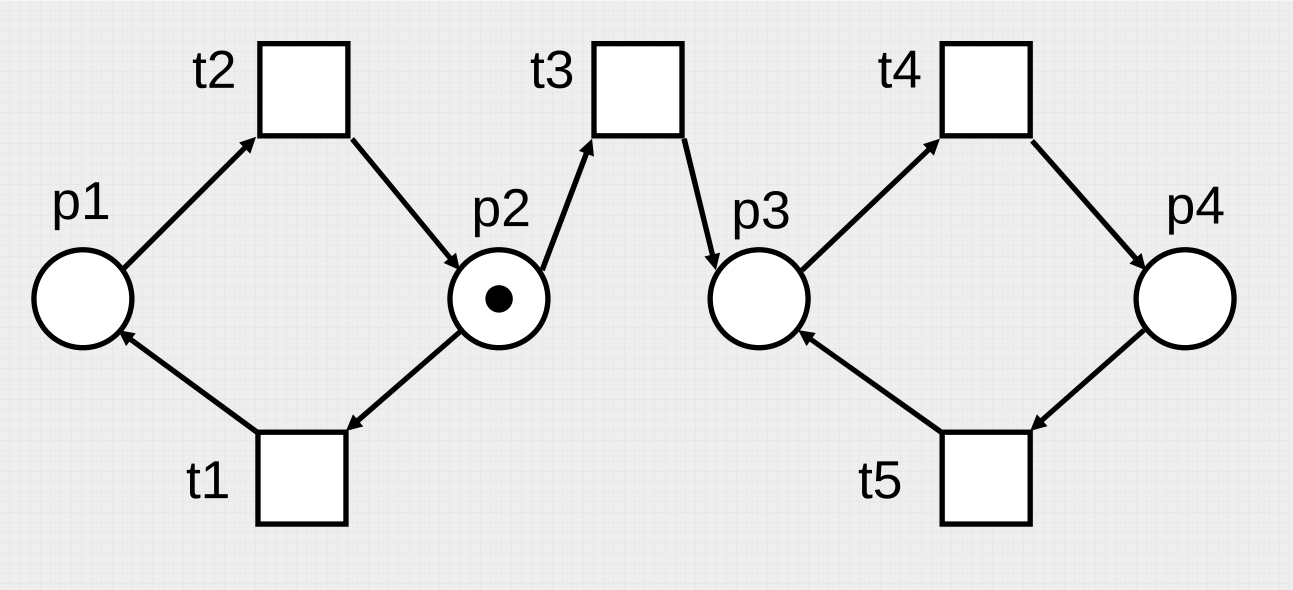


Markierungsgraph:

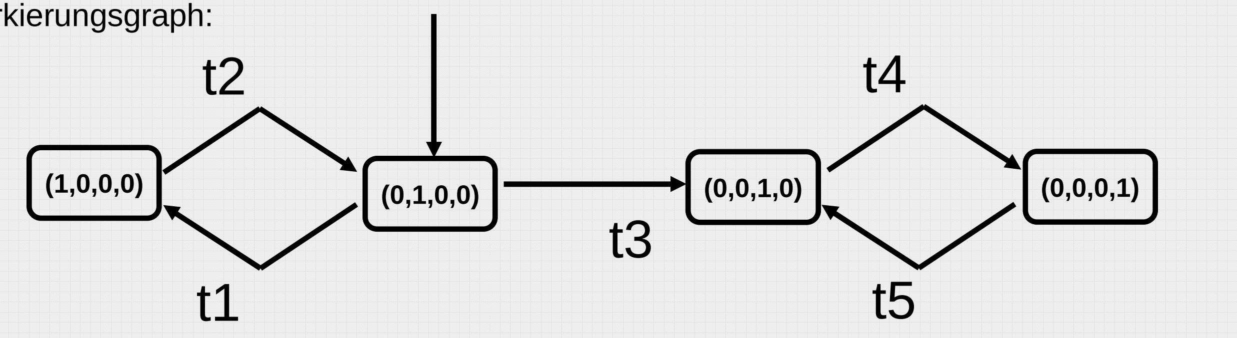


7.7.3 Verhaltenseigenschaften von S/T-Netzen

Beispiel:
Ein verklemmungsfreies, nichtlebendiges S/T-Netz



Markierungsgraph:



7.7.3 Verhaltenseigenschaften von S/T-Netzen

Lemmata:

(1) (N, m_0) ist verklemmungsfrei $\Leftrightarrow$
im zugehörigen Markierungsgraph hat jeder Knoten einen Nachfolger.

(2) (N, m_0) terminiert $\Leftrightarrow$
der zugehörige Markierungsgraph ist endlich und hat keine Zyklen.

(3) Kein terminierendes S/T-Netz ist verklemmungsfrei

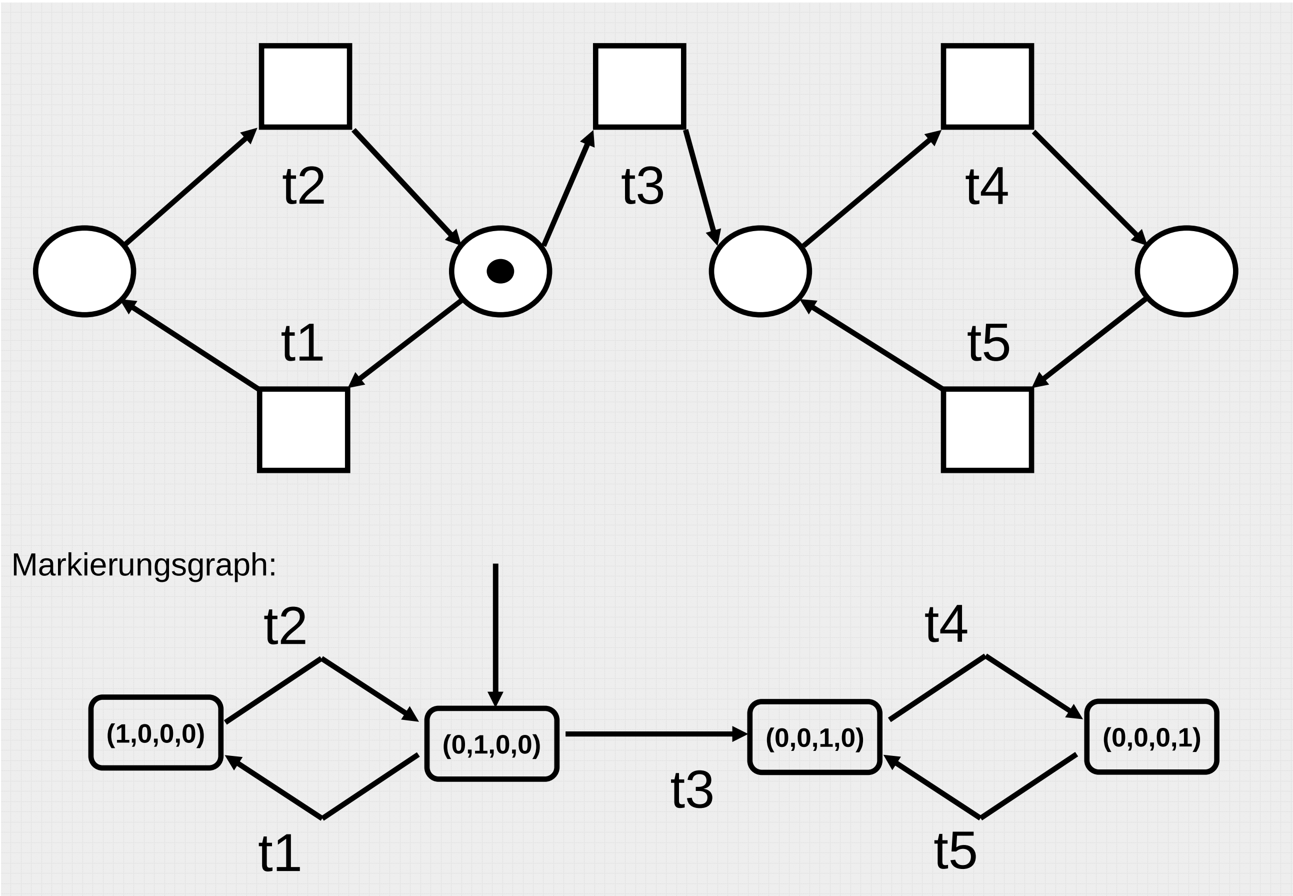
(4) Jedes lebendige S/T-Netz mit mindestens einer Transition ist verklemmungsfrei.

(5) (N, m_0) ist lebendig $\Leftrightarrow$
Jede Transition t kann – von jedem $m \in [m_0>$ ausgehend – immer wieder aktiviert werden.

(6) (N, m_0) ist rücksetzbar $\Leftrightarrow$
Der zugehörige Markierungsgraph ist stark zusammenhängend (mit Zyklen überdeckt; vgl. Folie 105).

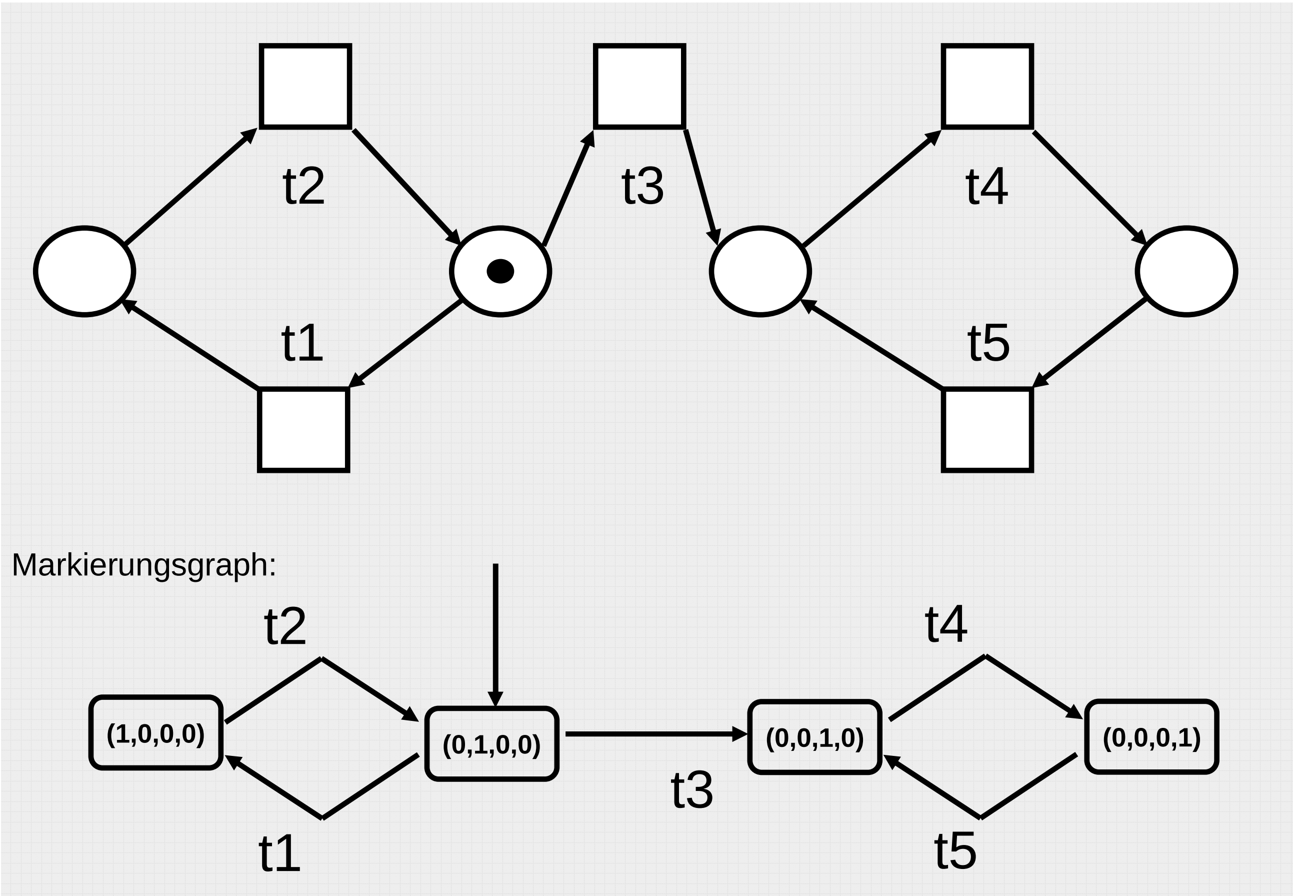
Beispiel:

Transitionen, die ab einem bestimmten Punkt nicht mehr schalten können (Lemma (5))



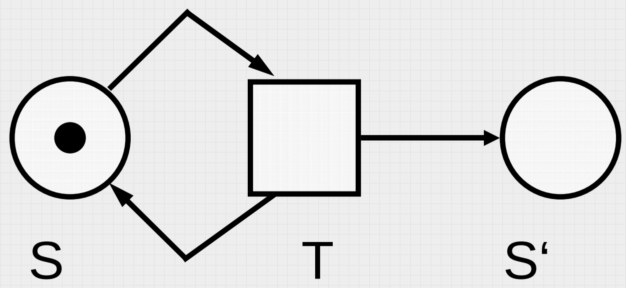
Beispiel:

beschränktes (b=1) nichtlebendiges S/T-Netz, das nicht rücksetzbar ist (Lemma (6))

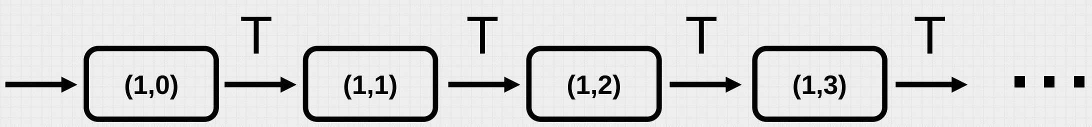


7.7.3 Verhaltenseigenschaften von S/T-Netzen

Unbeschränktes, lebendiges S/T-Netz, das nicht rücksetzbar ist (Lemma (6))

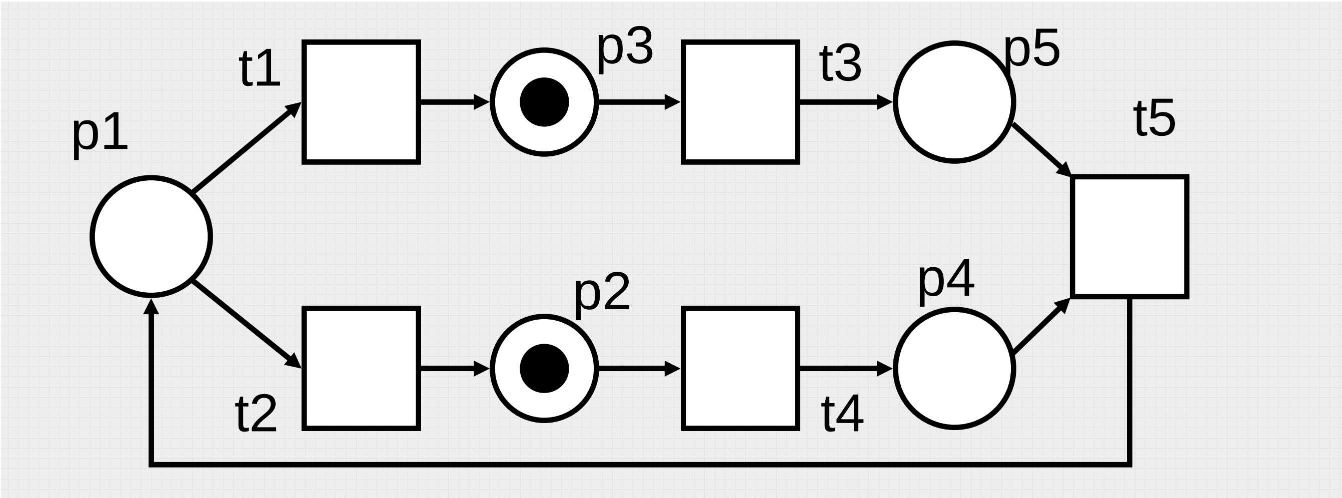


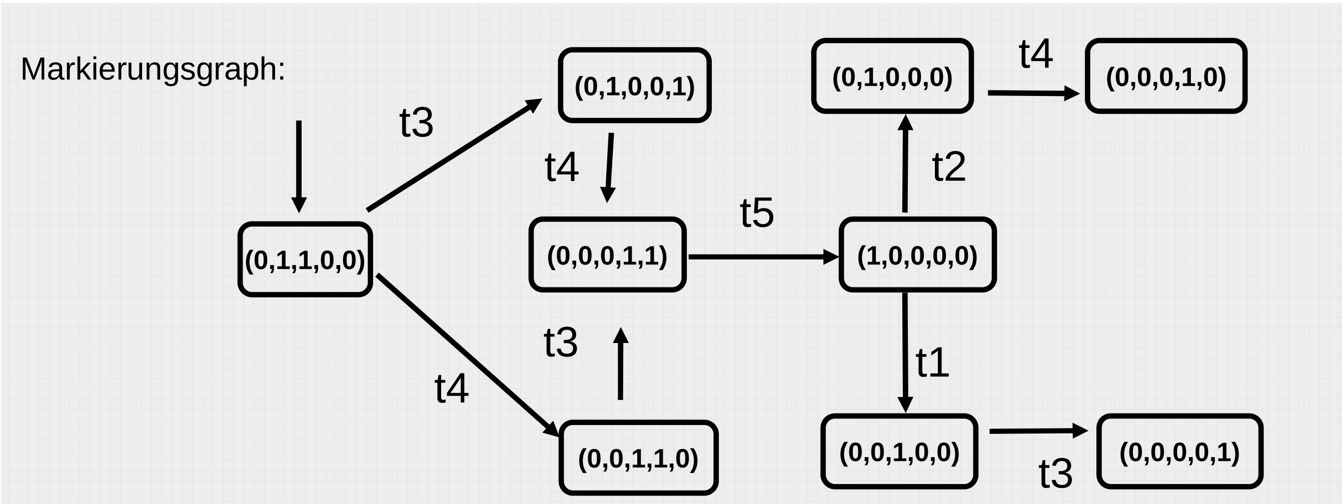
Markierungsgraph:



7.7.3 Verhaltenseigenschaften von S/T-Netzen

Ein beschränktes (b=1) S/T-Netz, das nicht rücksetzbar ist (Lemma (6))

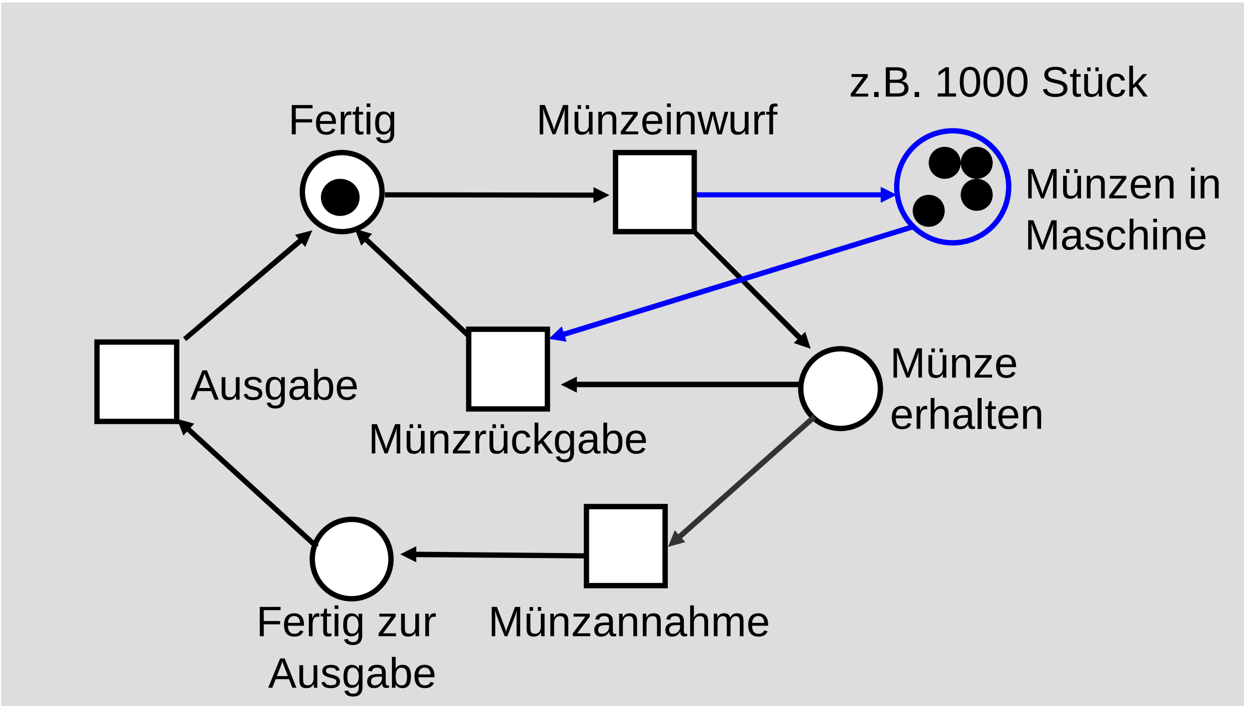




7.7.4 Kapazitätsbeschränkungen

Nur in den seltensten Fällen sind in der Realwelt eine beliebige Anzahl Marken in einer Stelle sinnvoll.

Beispiel: Münzautomat

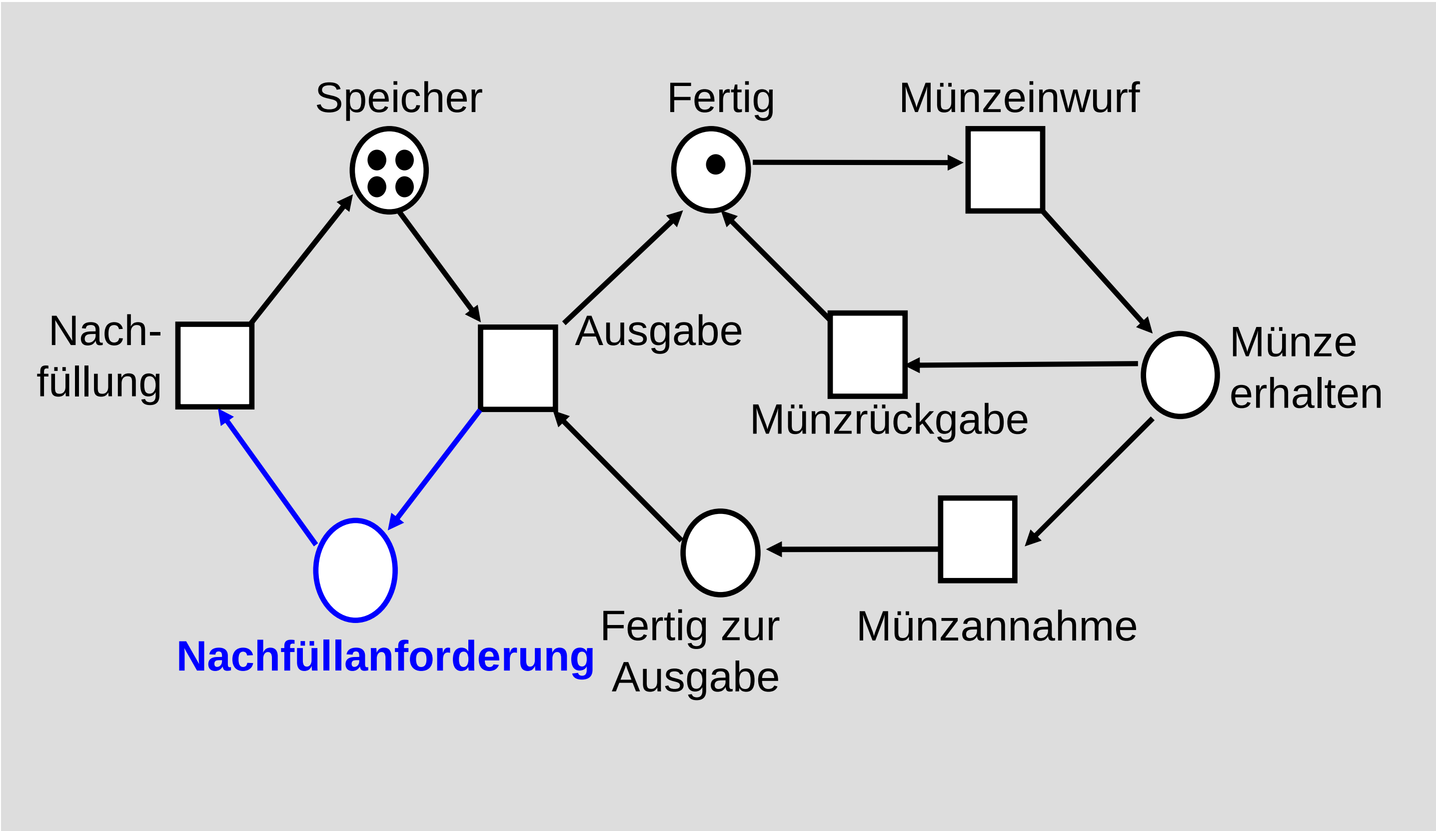


Ein realer Automat wird nur endlich viele Münzen aufnehmen können. Obiges Beispiel sieht diese Einschränkung jedoch nicht vor.

7.7.4 Kapazitätsbeschränkungen

Erzwingung von Schranken ...

... durch Komplementstellen



7.7.4 Kapazitätsbeschränkungen

Erzwingung einer endlichen Schranke durch Komplementstellen

Definition: Komplementstelle

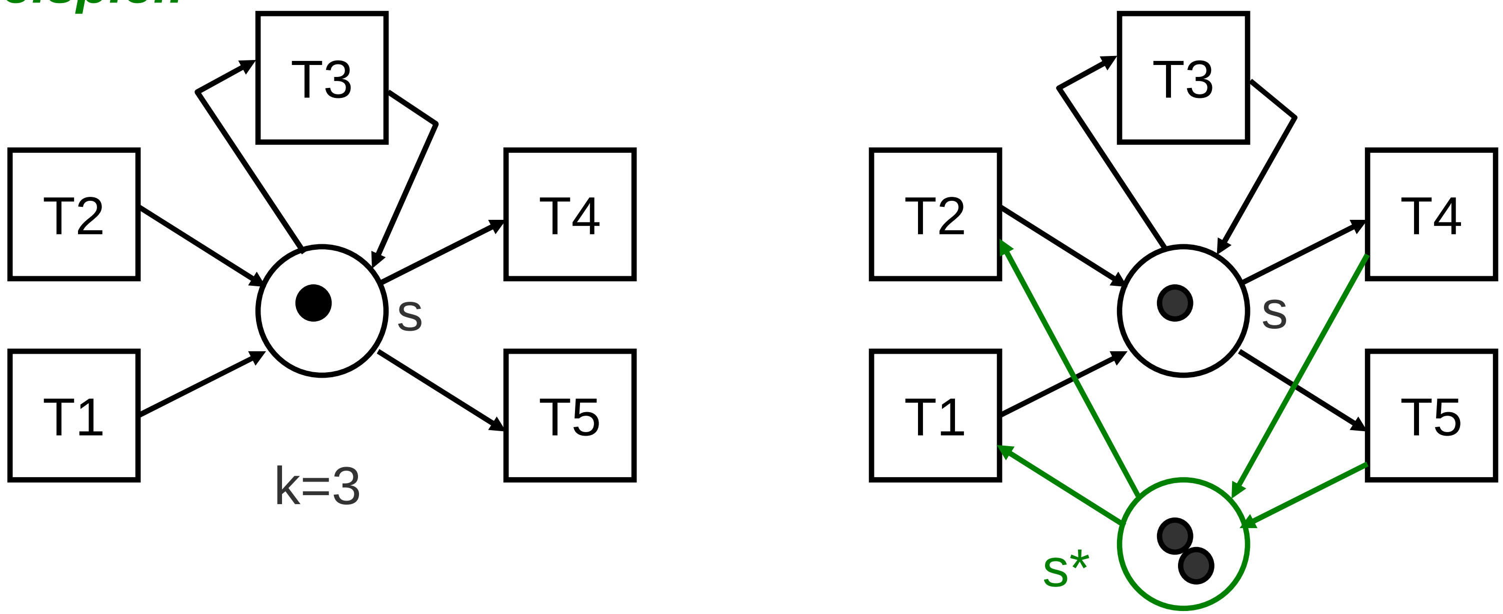
Wenn s eine Stelle ist, so wird eine neue Stelle s*, mit folgenden Eigenschaften:

- **s*• = •s \ s•**
- **•s* = s• \ •s**
- **m_0(s*) = k(s) - m_0(s),**
  dabei ist k(s) die gewünschte Schranke für s

als die **Komplementstelle von s** bezeichnet.

7.7.4 Kapazitätsbeschränkungen

Beispiel:



•s={T1,T2,T3}

s•={T3,T4,T5}

s*• = •s \ s• = {T1,T2}

•s* = s• \ •s = {T4,T5}

⇒ T1 oder T2 können nicht mehr Marken in s ablegen, als durch s* zugelassen

7.7.4 Kapazitätsbeschränkungen

Lemma

Jede erreichbare Markierung m erfüllt die Gleichung

$$m(s) + m(s^*) = k(s)$$

Beweis

Man betrachte $\bullet s \cup s\bullet$.

Es gilt $(\bullet s^* = s\bullet \setminus \bullet s) \wedge (s^*\bullet = \bullet s \setminus s\bullet)$

$\Rightarrow$

Jeder Schaltvorgang lässt die Zahl $m(s) + m(s^*)$ unverändert,
und somit ist nach der Definition von Komplementstellen
$m_0(s^*) + m_0(s) = k(s)$. (es gilt $m_0(s^*) = k(s) - m_0(s)$)

Korollar

Jede erreichbare Markierung m erfüllt $m(s) \leq k(s)$.

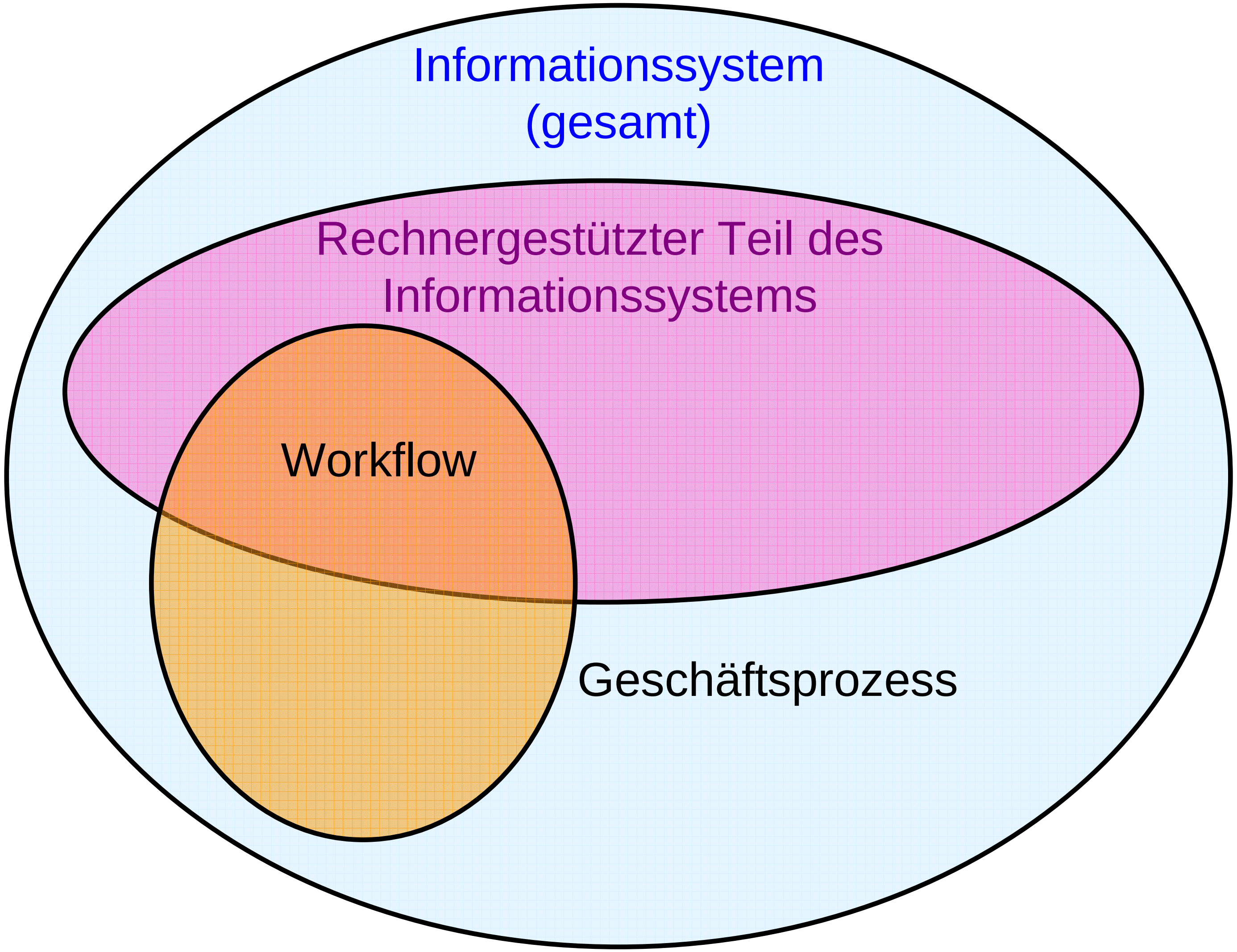
Beweis

$0 \leq m(s^*)$

7 Petrinetze

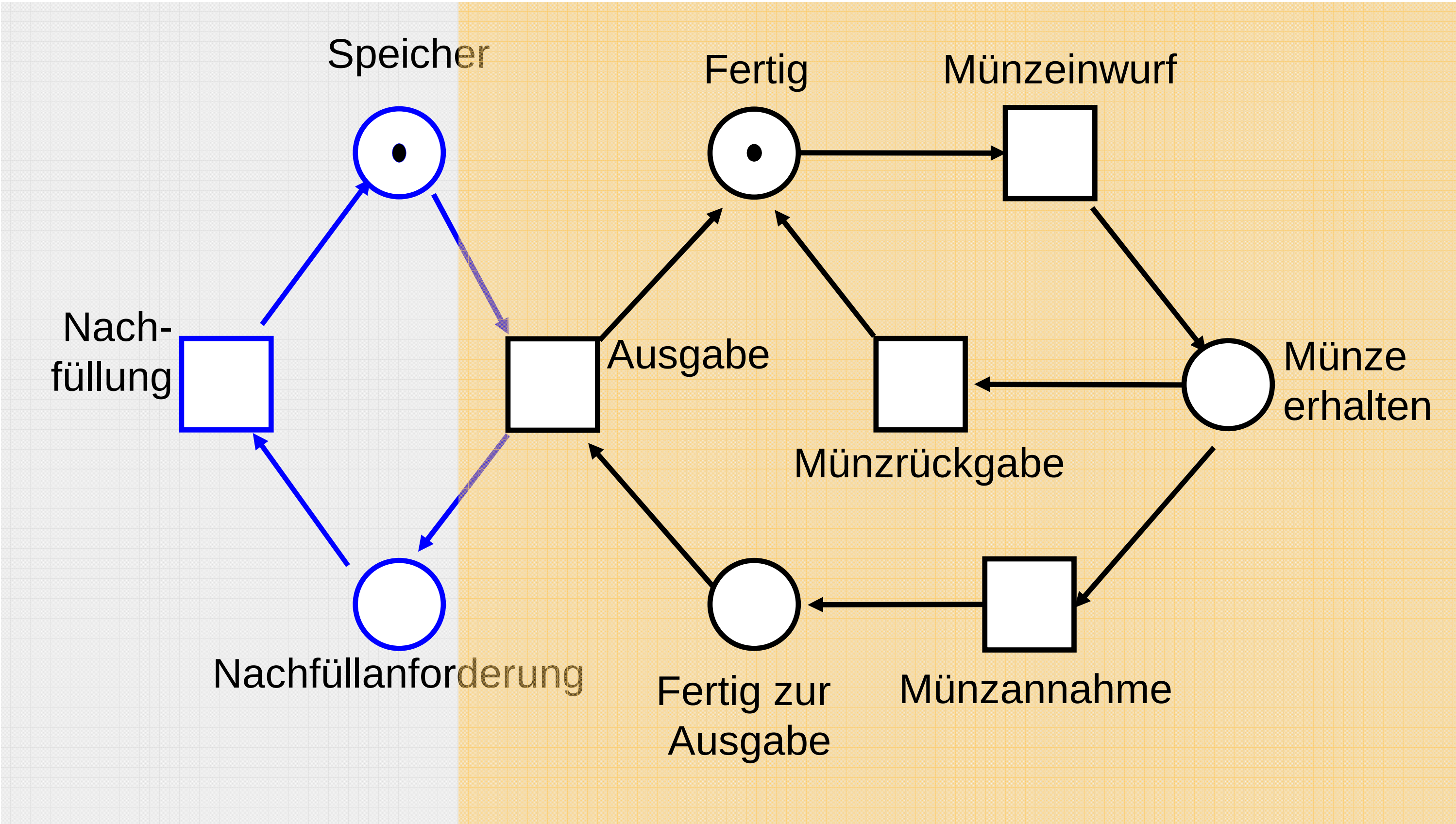
7.8.1 Überblick

Was wird durch Petrinetze modelliert?



7.8.1 Überblick

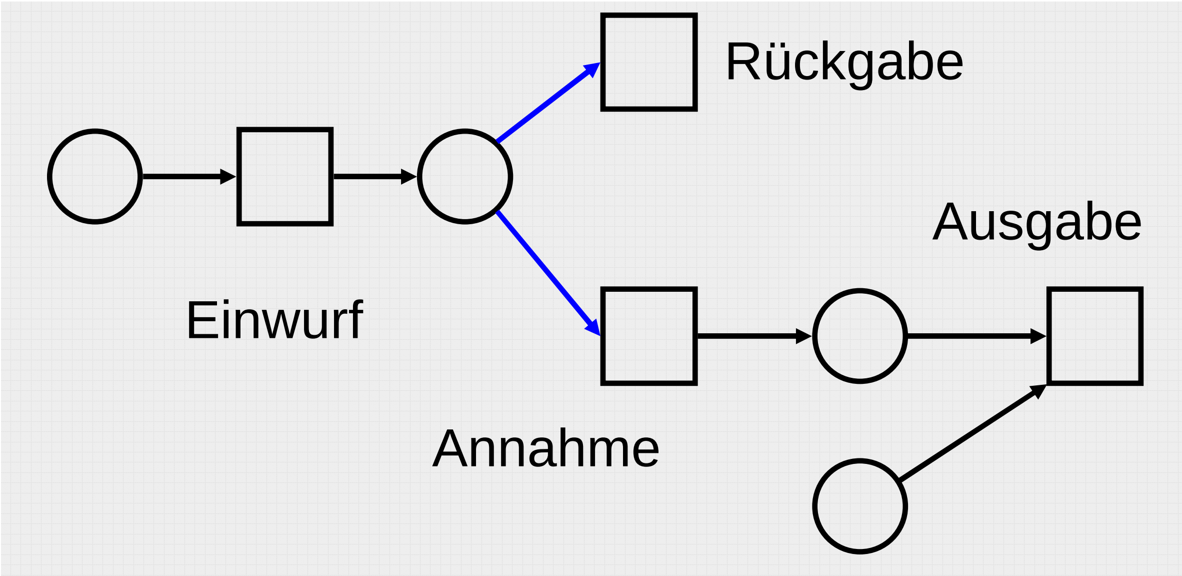
Gesamtsystem (Systemebene)



Rechnergestützter Teil des Informationssystems

7.8.1 Überblick

Prozesse:



initiiert durch Münzeinwurf

7.8.1 Überblick

- Petrinetz als Modell von Informationssystemen sollte verklemmungsfrei oder (besser) lebendig sein
- Petrinetz als Modell von Prozessen sollte **nicht** verklemmungsfrei oder (besser) terminierend sein.

Vgl. Definition aus 7.7.3

7.8.2 Ziele der Modellierung

(1|2)

Ziele der Modellierung:

- **Validierung**: Ist das Modell richtig bzgl. der Realität/Vorstellung
  - grafische Darstellung
  - Simulation des Verhaltens

- **Simulation**: Erzeugung einiger / aller Abläufe
  - sequentielle Abläufe **/** nichtsequentielle Abläufe
  - animierte Simulation **vs.**
  - Ablauferzeugung im Hintergrund
    (z.B. quantitative Simulation)
    (z.B. Laufzeitverhalten)
  - zur Validierung **bzw.**
  - zur Verifikation (im Sinne von Tests)

7.8.2 Ziele der Modellierung

- **Analyse**: Ermittlung von Eigenschaften
  (z.B. Lebendigkeit, Rücksetzbarkeit)
  - Validierung
  - Hilfsmittel zur Verifikation
- **Automatische Verifikation**: Ermittlung spezieller Eigenschaften
  (gegeben Modell + gewünschte Eigenschaften;
  Frage: Sind Eigenschaften im Modell erfüllt?)
  - z.B. nachdem eine Münze in den Automat geworfen wurde, wird sie entweder zurückgegeben oder etwas wird ausgegeben
- **Semiautomatische Verifikation**: Beweisverfahren
  (gegeben Modell + gewünschte Eigenschaften + Beweisschritte;
  Frage: Wie sieht ein Beweis aus?)
  - Proof Checker

Methoden für Analyse und Verifikation

- Erzeugung des Markierungsgraphen
- Strukturtheorie

Strukturtheorie

Gewinnung von Aussagen über das Verhalten in einem Netz mit Hilfe der

- statischen Netzstruktur
- der Anfangsmarkierung

(d.h. ohne den Markierungsgraph zu konstruieren)

7.8.3 Methoden für Analyse und Verifikation

Satz: Zusammenhangssatz

Jedes lebendige und beschränkte zusammenhängende S/T-Netz ist stark zusammenhängend.

Definition: Stark zusammenhängend

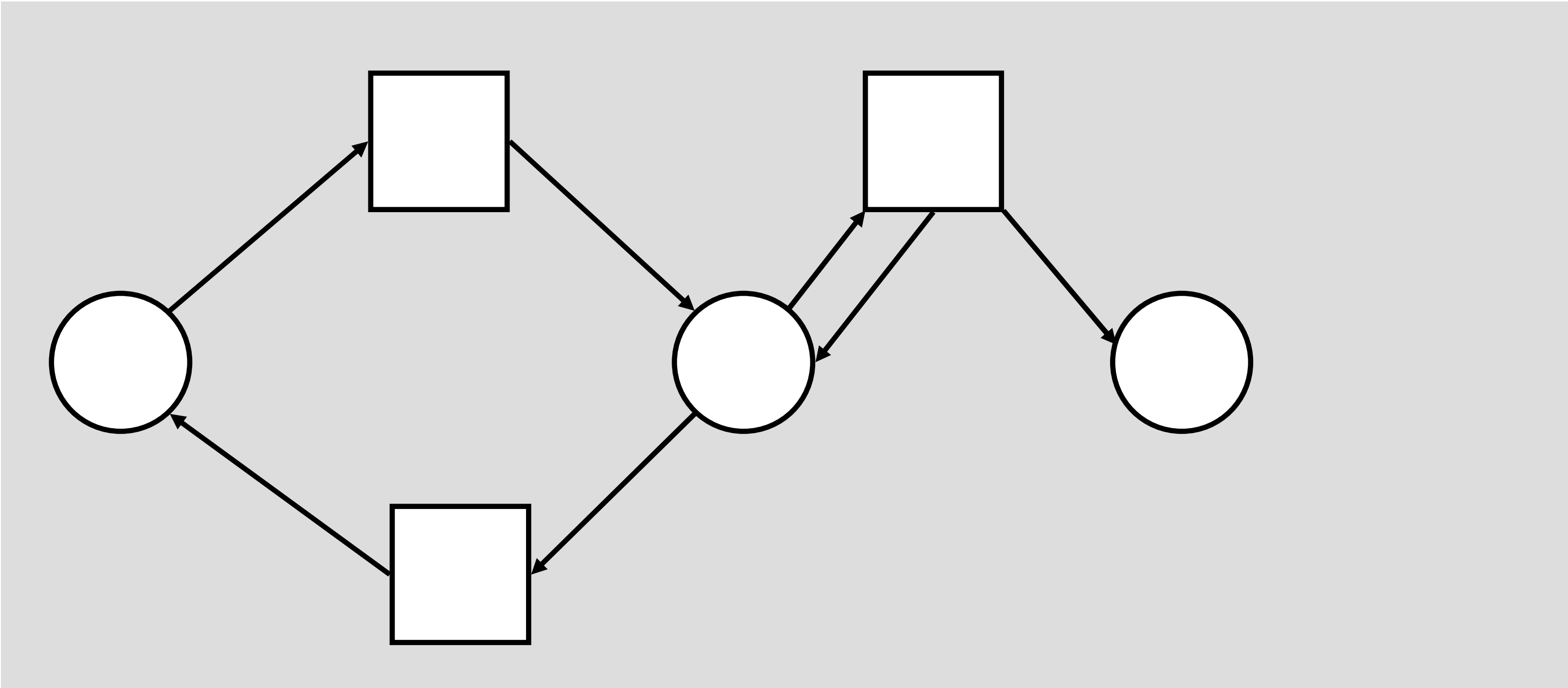
N = (S, T, F) heißt stark zusammenhängend :$\Leftrightarrow$

für je zwei Elemente $x, y \in S \cup T$, $x \neq y$
gibt es eine Sequenz $(z_1, z_2), (z_2, z_3), \ldots, (z_{n-1}, z_n) \in F$ $(2 \leq n)$,
so dass $x = z_1$ und $y = z_n$.

Vgl.: "zusammenhängend" und Beispiele aus 7.7.3

7.8.3 Methoden für Analyse und Verifikation

Das folgende Netz ist zusammenhängend, aber nicht stark zusammenhängend:



Wegen des Zusammenhangssatzes ist es daher unter keiner Anfangsmarkierung lebendig und beschränkt.

7 Petrinetze

7.9 Ausblick

- Weitere Details zu Petrinetzen
(Anwendungsgebiete, Klassifikation, Bezug zu formalen Sprachen, Programmierung, u.v.a.m.)
in T. Murata, Petri Nets: Properties, Analysis and Applications, Proc. IEEE, Vol 77(4), April 1989

- Es gibt noch weitere Methoden, um Petrinetze zu beschreiben bzw. zu untersuchen, z.B. mit Hilfe linearer Algebra:

  – Markierungen sind Vektoren
  – Petrinetz als Inzidenzmatrix modelliert
  – Schalten von Transitionen als Matrix-Vektor-Produkt modelliert
  – weitere Details in Workflow-Vorlesung (SS 2006) bzw. angegebener Literatur

Literatur

James L. Peterson: **Petri Nets**.
ACM Computing Surveys (CSUR), Volume 9, Issue 3, September 1977

Erhältlich unter:

http://portal.acm.org/ft_gateway.cfm?id=356702&type=pdf&coll=ACM&dl=ACM&CFID=28996548&CFTOKEN=25198609

Der Link funktioniert nur innerhalb des Uni-Netzes der Universität Karlsruhe.

(siehe auch Seite zur Vorlesung - Literatur)

Literatur

- James L. Peterson: **Petri Nets**. ACM Computing Surveys (CSUR), Volume 9, Issue 3, September 1977
- T. Murata: **Petri nets: Properties, analysis and applications**. Proceedings of the IEEE, Volume 77, Issue 4, April 1989, pp. 541 - 580
- Wolfgang Reisig: **A Primer in Petri Net Design**. Springer Verlag, 1992
- J. Desel, J. Esparza: **Free Choice Petri Nets**. Cambridge University Press, 1995
- Bernd Baumgarten: **Petri-Netze. Grundlagen und Anwendungen**. Spektrum Akademischer Verlag, 1996
- W. van der Aalst, J. Desel, A. Oberweis (Eds.): **Business Process Management: Models, Techniques, and Empirical Studies**. Lecture Notes in Computer Science 1806, Springer Verlag Heidelberg, 2000