Vorlesung

Softwarekonstruktion

im Wintersemester 2014/15

Prof. Dr. Jan Jürjens

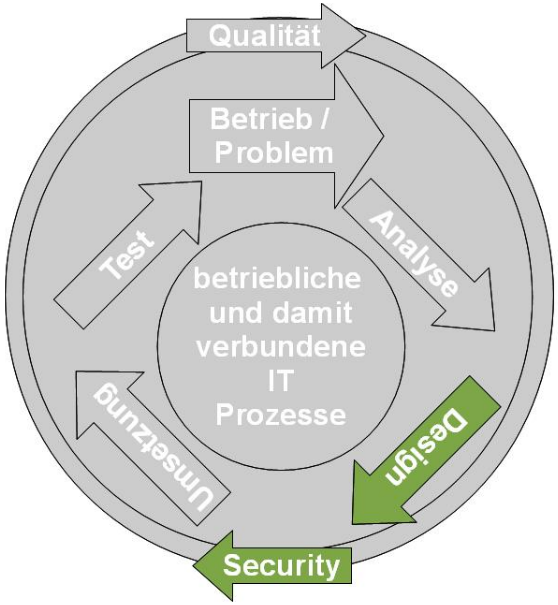
TU Dortmund, Fakultät Informatik, Lehrstuhl XIV

Teil 1.4: Petrinetze

v. 05.12.2014

Einordnung
1.4 Petrinetze

- Modellgetriebene SW-Entwicklung
  - Einführung
  - Modellbasierte Softwareentwicklung
  - OCL
  - Ereignisgesteuerte Prozesskette (EPK)
  - Petrinetze
  - Eclipse Modeling Framework (EMF)
- Qualitätsmanagement
- Testen



Inkl. Beiträgen von Prof. Volker Gruhn, Jutta Mülle und Dr. Silvia von Stackelberg.

Literatur (s. Vorlesungswebseite):

[Rei10] W. Reisig: Petrinetze. Vieweg, 2010.
- Teil I

Einleitung Petrinetze

Vorheriger Abschnitt:

- GP-Modellierungsnotationen **EPK**.
  → Intendiertes Modellverhalten informell diskutiert.

Automatische Verarbeitung (z.B. Analyse, Simulation) der GP-Modelle benötigt präzise Definition des Ausführungsverhaltens.

Verschiedene Ansätze: Abstract State Machines, Petrinetze, ...

- Z.B.: **Ausführungssemantik** von **UML 2-Aktivitätsdiagrammen** mit Petrinetzen definiert.

=> Dieser Abschnitt:

Einführung in Petrinetze

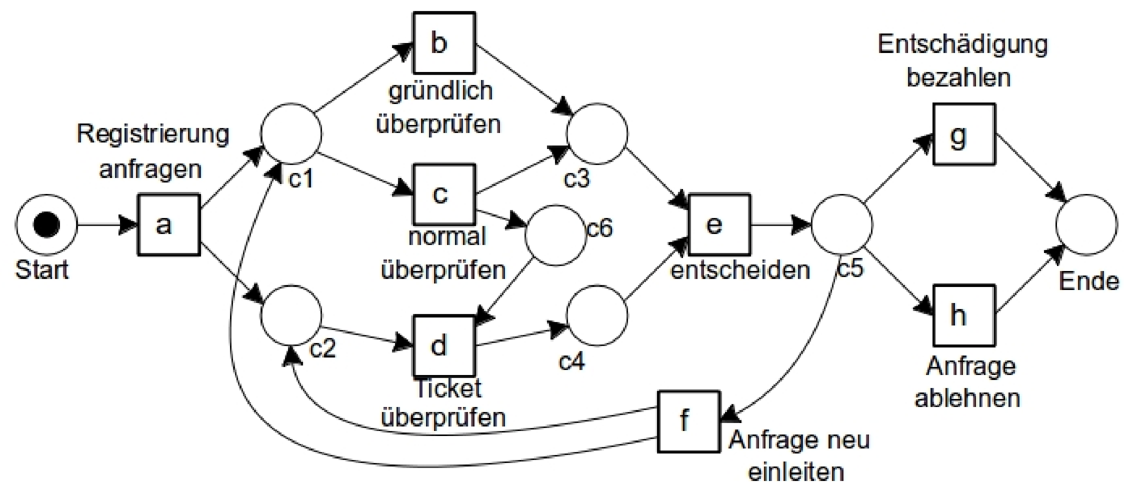
1.4 Petrinetze

1.4 Petrinetze

Petrinetz Syntax
Ausführung
Analyse von Systemen

Petrinetze

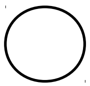
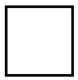
- Modellierung, Analyse, Simulation von dynamischen Systemen mit **nebenläufigen** und **nichtdeterministischen** Merkmalen.
- Erlauben die Beschreibung von **Kontroll- und Datenfluss**.
- Benannt nach Carl Adam Petri (Dissertation "Kommunikation mit Automaten", 1962).

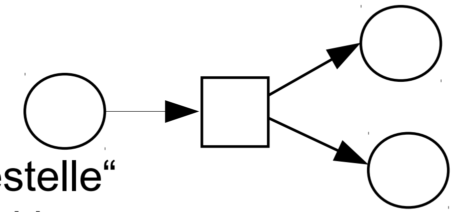


Vorsicht: Es existieren heute viele Varianten.

Petrinetz (informell)

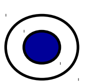
Statische Komponente: Bipartiter gerichteter Graph, bestehend aus:

- zwei Sorten von **Knoten**:
  - **Stelle**: Zwischenablage von Informationen 
  - **Transition**: Verarbeitung von Informationen 
- **Kanten**: verbinden Stellen mit Transitionen oder umgekehrt (**nie** Stellen mit Stellen oder Transitionen mit Transitionen !).

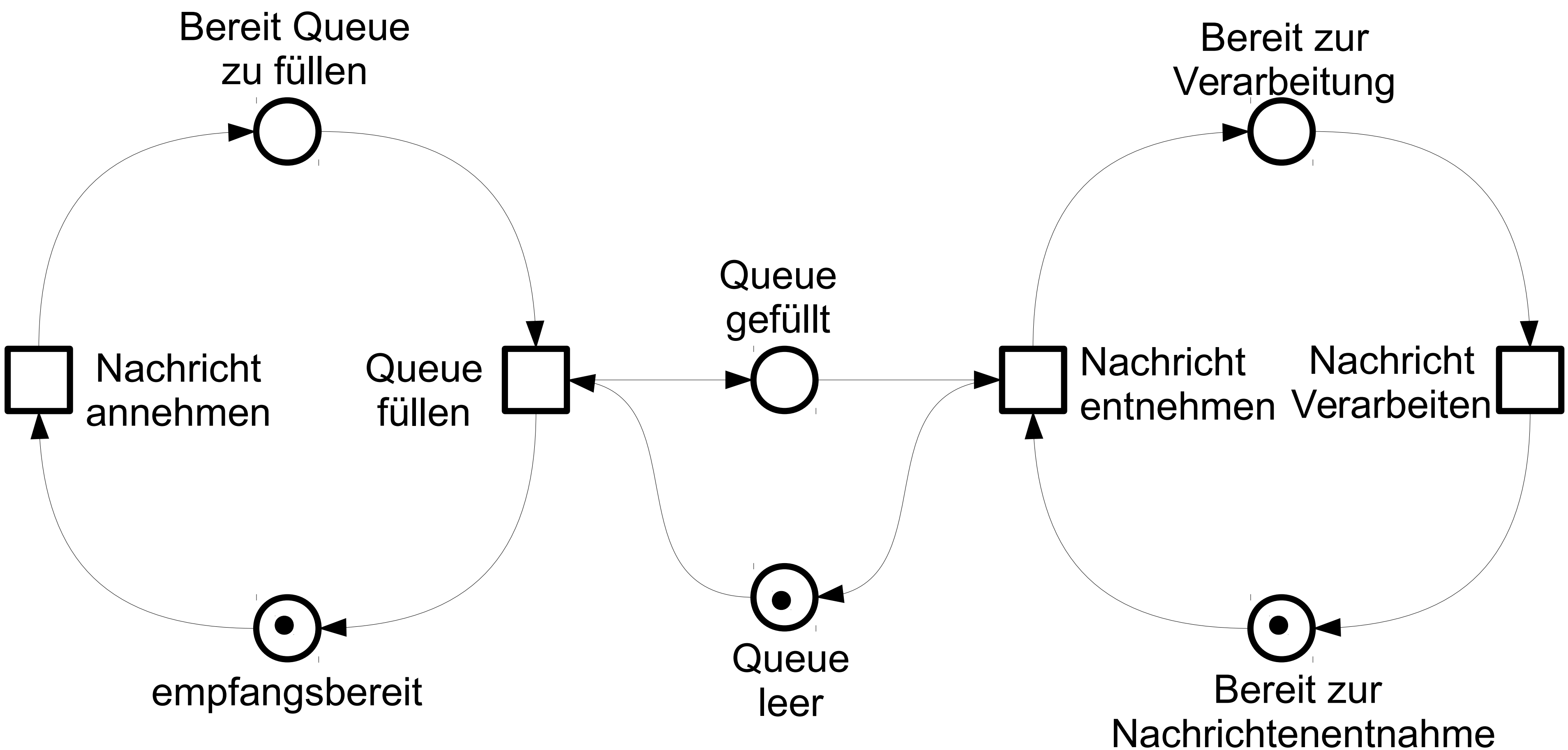


Dynamische Komponente:

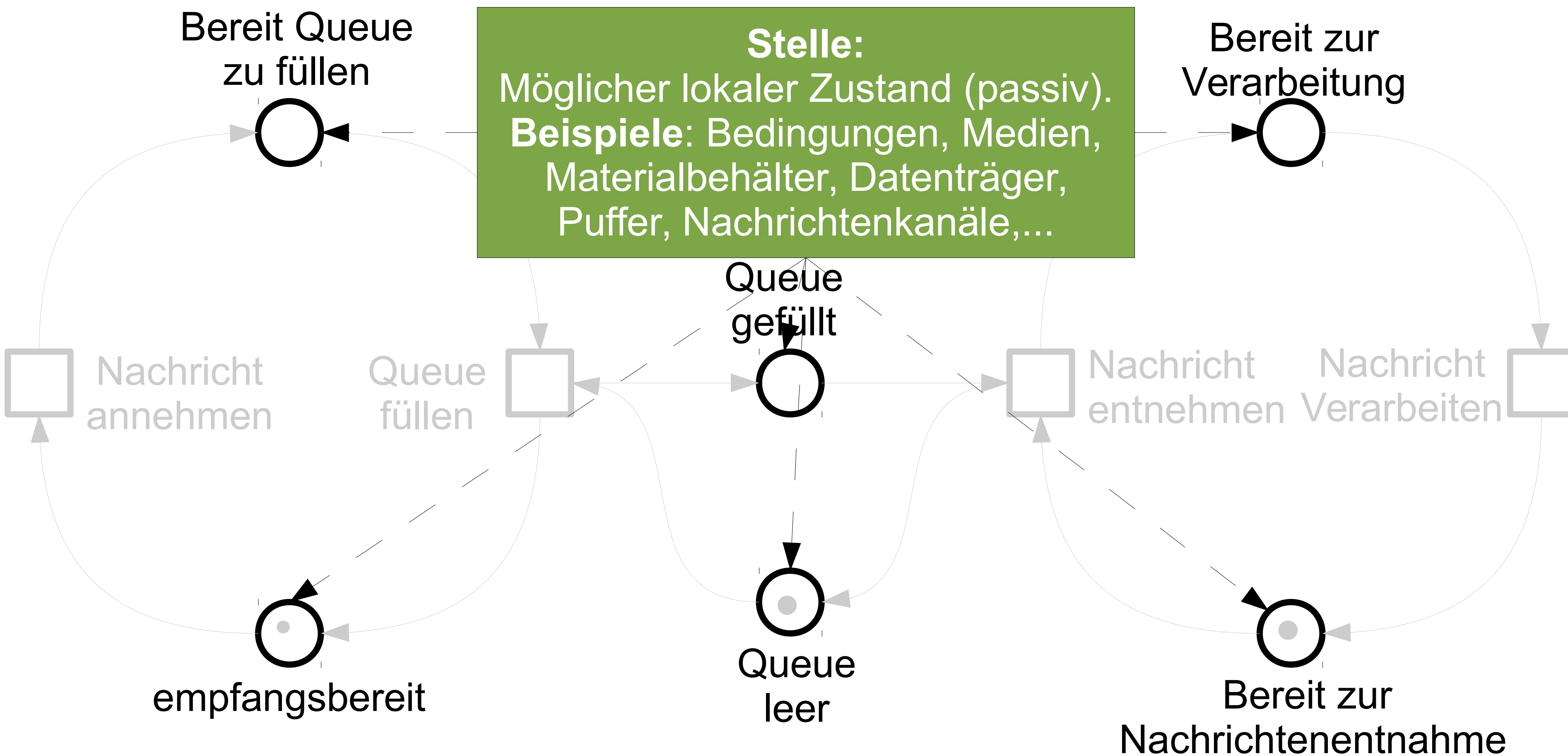
- Marken („Token“): Stellen werden mit Objekten belegt.
  - Durchlauf der Marken durch Petrinetz beschreibt **dynamisches Verhalten** des Systems.



Petrinetz: Beispiel Nachrichten-Queue



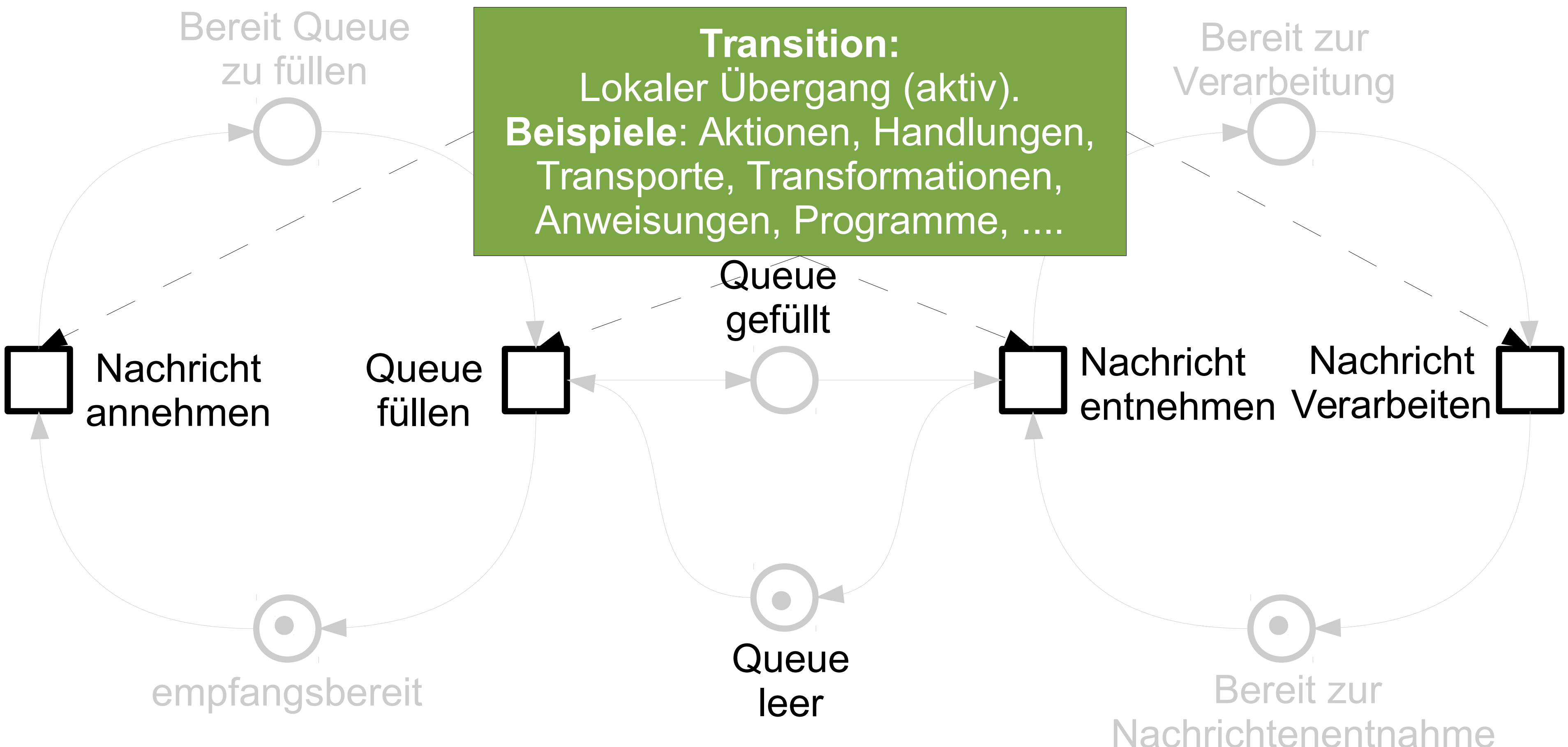
Petrinetz-Syntax: Stelle



Stelle:
Möglicher lokaler Zustand (passiv).
Beispiele: Bedingungen, Medien, Materialbehälter, Datenträger, Puffer, Nachrichtenkanäle,...

Petrinetz-Syntax: Transition

Transition:
Lokaler Übergang (aktiv).
Beispiele: Aktionen, Handlungen, Transporte, Transformationen, Anweisungen, Programme, ...



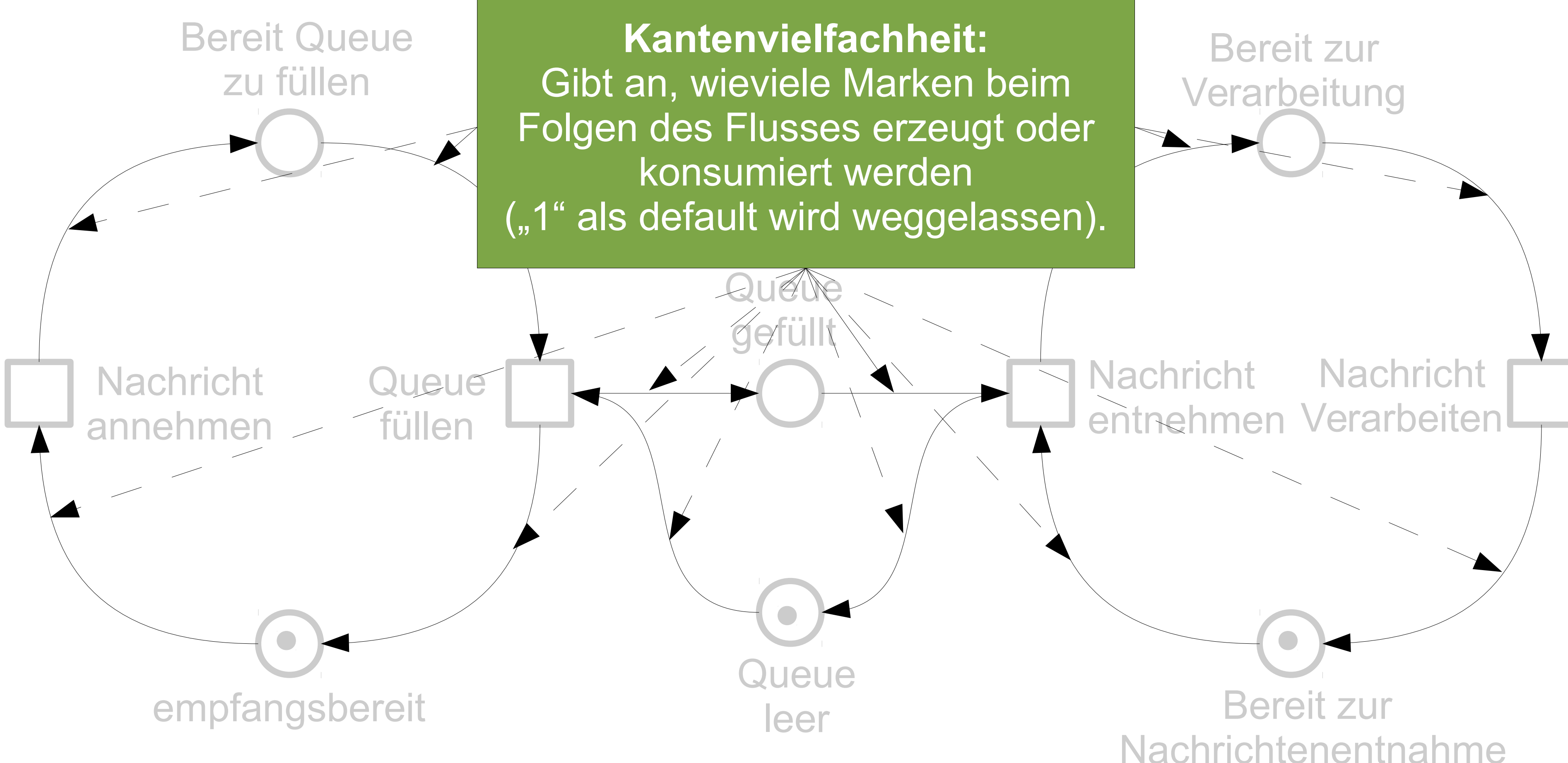
Petrinetz-Syntax: Kanten

Kanten:
Fluss (automatisch).
Beispiele: Vor- und Nachbedingungen von Aktivitäten, Start und Ziel von Transporten, Eingabe und Ausgabe von Programmen,...

Petrinetz-Syntax: Marken

Marken:
Beispiel: Zustände einer Bedingung, Gültigkeit von Bedingungen, Füllungsgrad von Speichern, Daten auf Datenträgern, Nachrichten in Puffern, ...

Petrinetz-Syntax: Kantenvielfachheit



Kantenvielfachheit:
Gibt an, wieviele Marken beim Folgen des Flusses erzeugt oder konsumiert werden („1“ als default wird weggelassen).

Syntaxdefinition Petrinetz

Gegeben:

- S: endliche Menge von **Stellen**
- T: endliche Menge von **Transitionen**

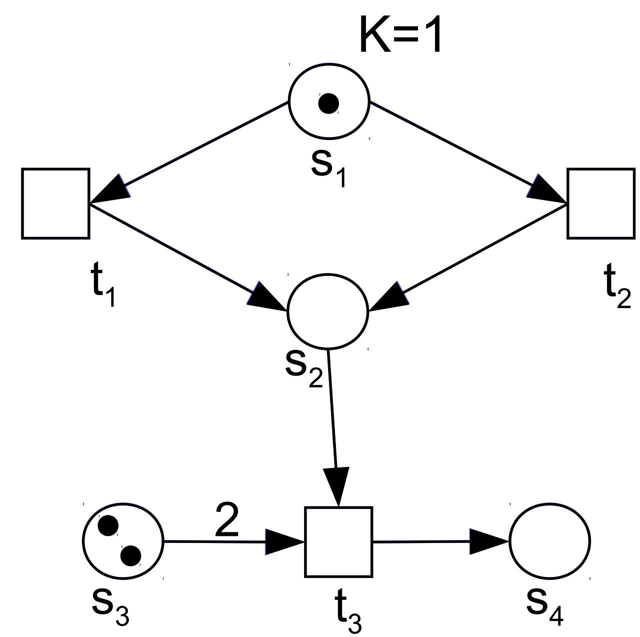
  mit: $\mathbf{S \neq \emptyset}$,

  $\mathbf{T \neq \emptyset}$ **und**

  $\mathbf{S \cap T = \emptyset}$
- F: Menge von Kanten

  mit: $F \subseteq (S \times T) \cup (T \times S)$

  (binäre Relation).



$S = \{s_1, s_2, s_3, s_4\}$

$T = \{t_1, t_2, t_3\}$

$F = \{(s_1,t_1), (s_1,t_2), (t_1,s_2), (t_2,s_2), (s_2,t_3), (s_3,t_3), (t_3,s_4)\}$

Syntaxdefinition Petrinetz

Gegeben:

- K: **Kapazität**
  („Fassungsvermögen der Stellen“)

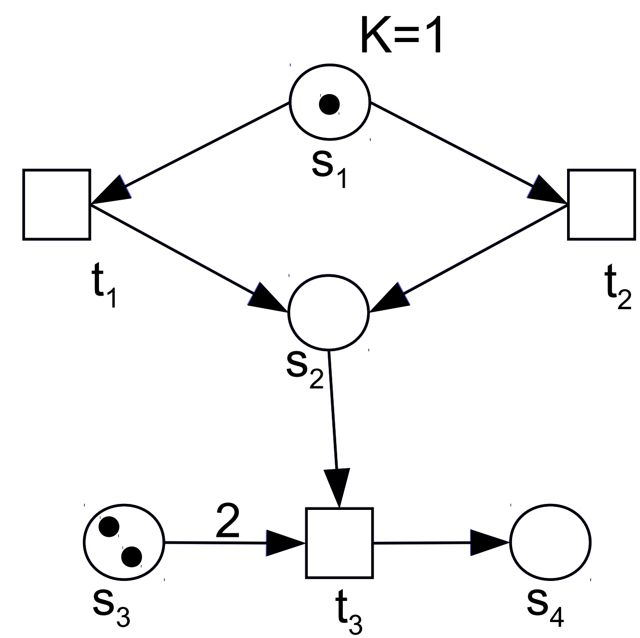
  mit: $K: S \rightarrow \mathbb{N} \cup \{\infty\}$

  Default-Kapazität ∞

- W: **Kantenvielfachheit**

  mit: $W: F \rightarrow \mathbb{N} \setminus 0$

  Default-Kantengewicht 1



$K(s_1)=1$; $K(s_2)=K(s_3)=K(s_4)=\infty$

$W(s_1,t_1)= W(s_1,t_2)= W(s_2,t_1)=$
$W(s_2,t_2)= W(s_2,t_3)= W(t_3,s_4)= 1,$

$W(s_3,t_3)= 2$

Syntaxdefinition Petrinetz

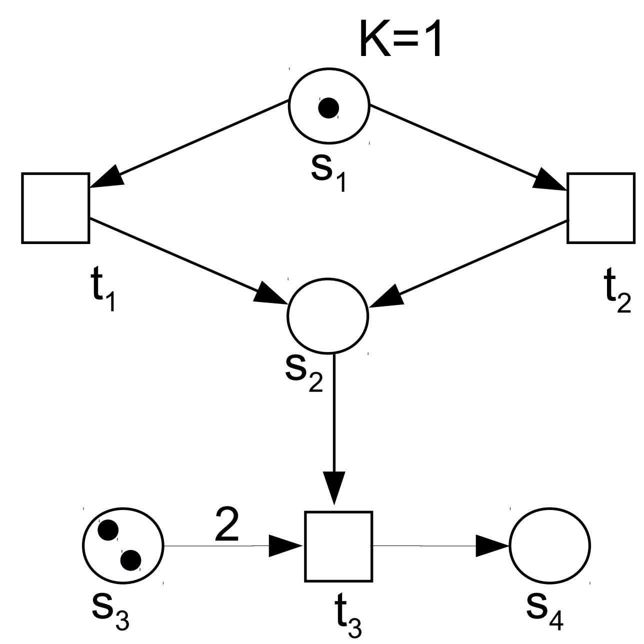
Gegeben:

- M_0 : **Globaler Startzustand**

  („Anfangsmarkierung")

  mit: $M_0 : S \rightarrow \mathbb{N}$

Dann: (S, T, F, W, K, M_0) ist **Petrinetz**



$M_0(s_1)= 1,$
$M_0(s_2)= 0,$
$M_0(s_3)= 2,$
$M_0(s_4)= 0$

1.4 Petrinetze

1.4 Petrinetze

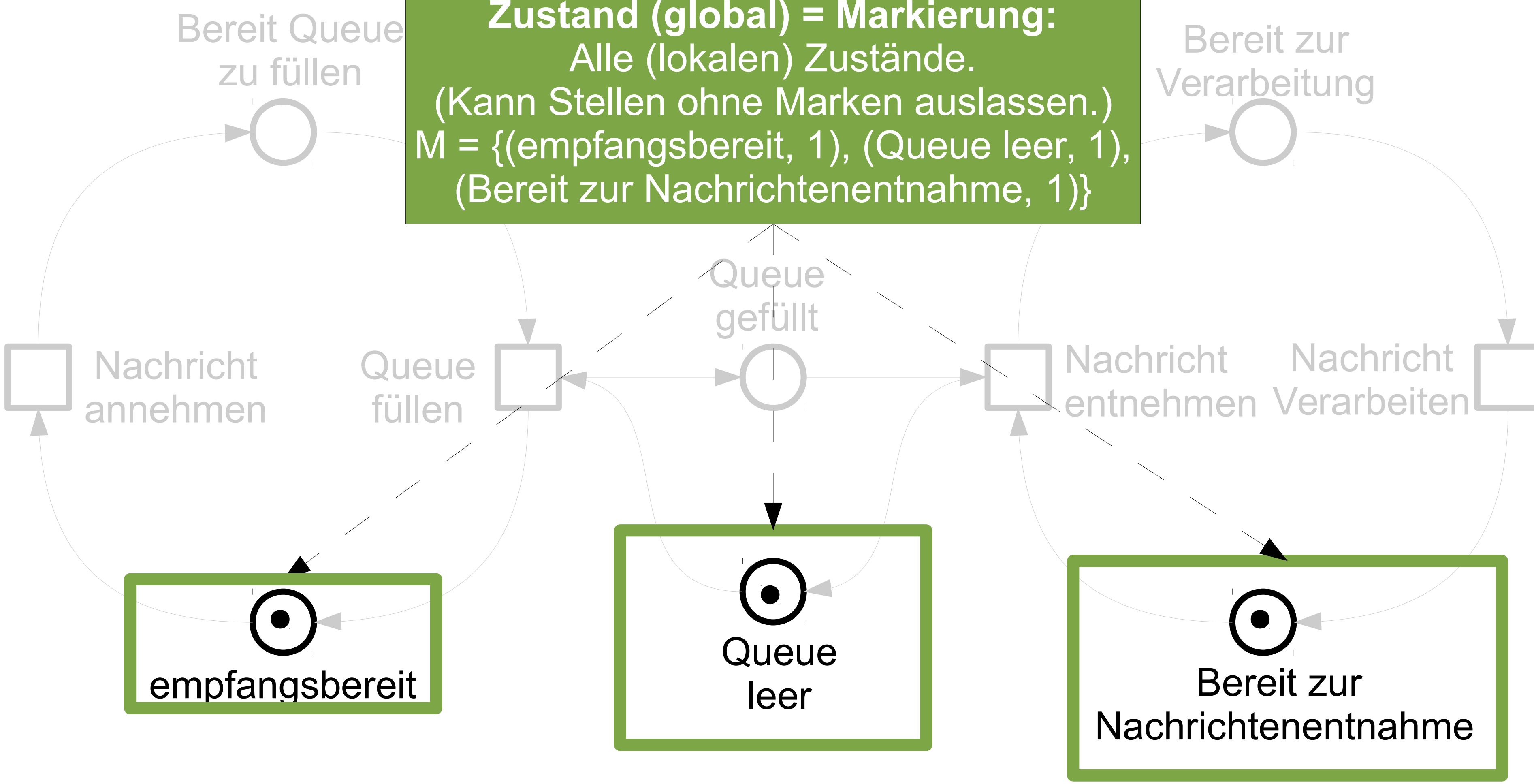
- Petrinetz
- Ausführung
- Analyse von Systemen

Petrinetz-Ausführung: Zustand (lokal)

Zustand (lokal):
(Stelle, Marke(n))
Interpretation je nach Bedeutung der Stelle
Beispiel:
$M_{empfangsbereit}$ = (empfangsbereit,1)

Petrinetz-Ausführung: Zustand (global)

Zustand (global) = Markierung:
Alle (lokalen) Zustände.
(Kann Stellen ohne Marken auslassen.)
M = {(empfangsbereit, 1), (Queue leer, 1), (Bereit zur Nachrichtenentnahme, 1)}



Petrinetz: Markierung

Markierung: Verteilung Marken auf Stellen (aktueller Systemzustand).

Markierung M:

mit: $M : S \to \mathbb{N}$

Markierungen müssen Kapazitäten respektieren,

d.h. für jede Stelle $s \in S$ gilt: $M(s) \leq K(s)$.

Initiale Markierung: Anfangszustand eines Netzes.

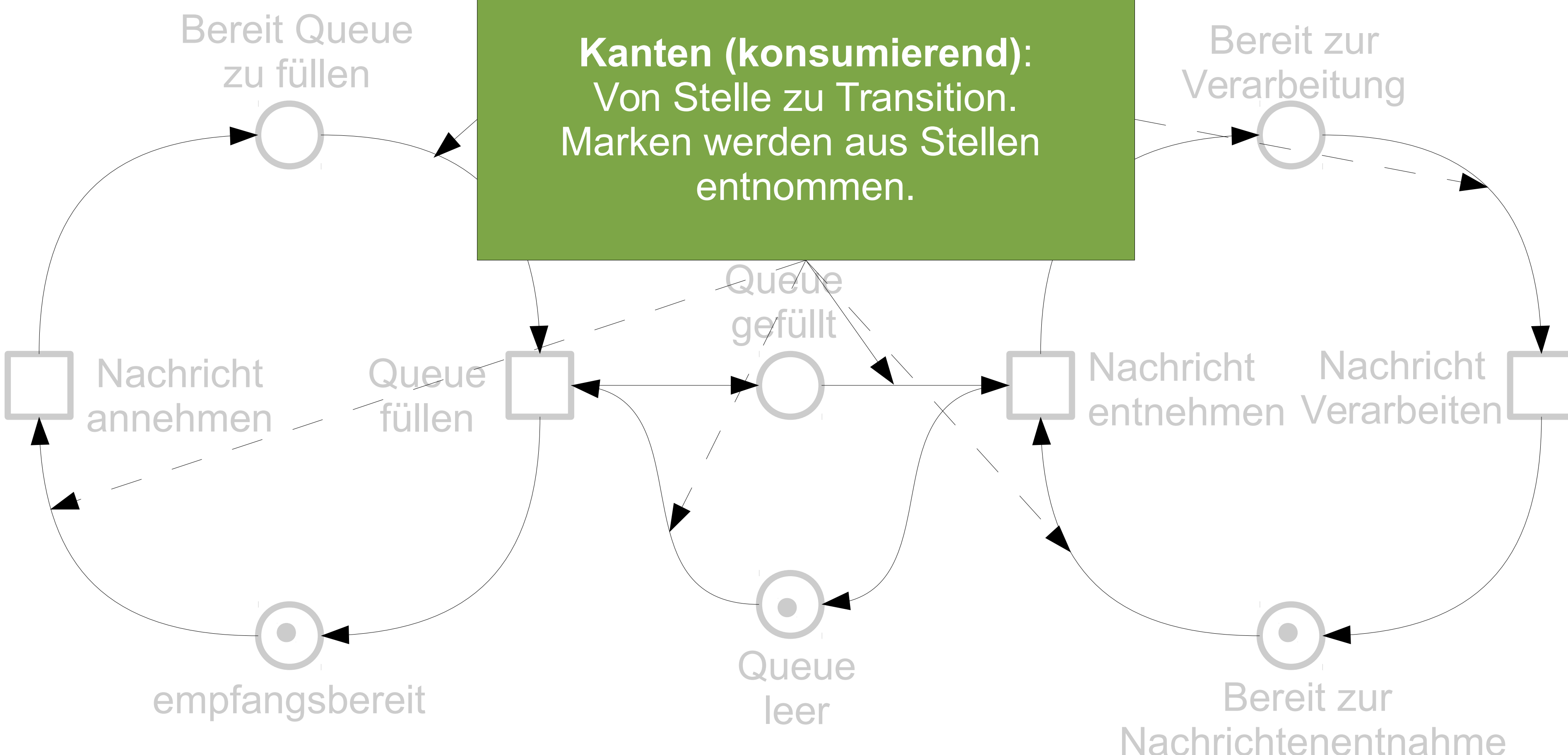
Petrinetz: Ausführung

LEHRSTUHL 14
SOFTWARE ENGINEERING

Verhaltenssimulation: evolvierende Anzahl Marken pro Stelle beobachten.

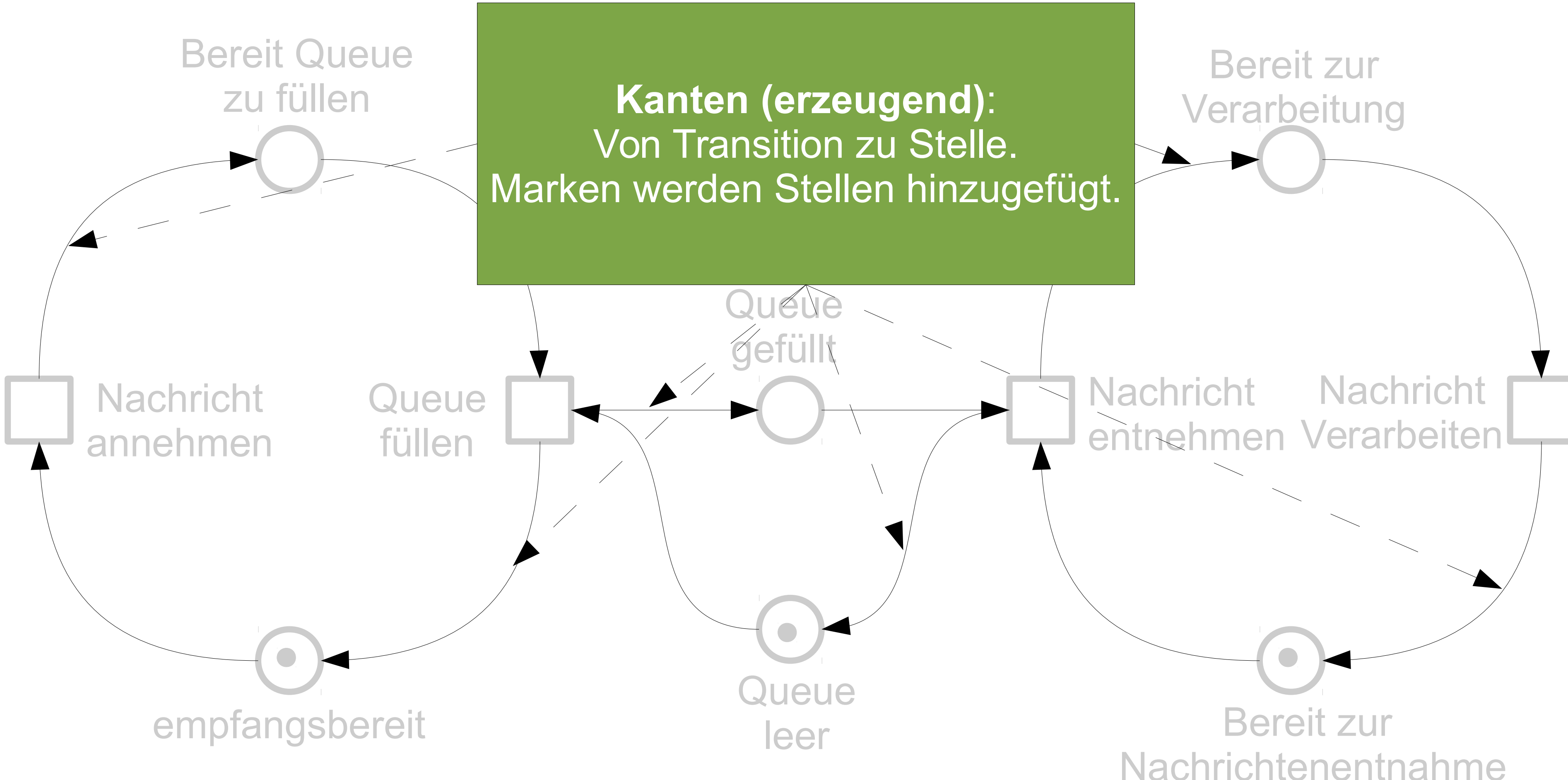
- Basierend auf aktueller Markierung: **aktivierte Transitionen** ermitteln. **Schalten** führt zu Folgemarkierung.
- Unter Folgemarkierung sind (möglicherweise) andere Transitionen aktiviert.
- Solange iterieren, bis keine Transition mehr aktiviert ist. (=> „**tote Markierung**").

Petrinetz-Syntax: Kanten (konsumierend)



Kanten (konsumierend):
Von Stelle zu Transition.
Marken werden aus Stellen entnommen.

Petrinetz-Syntax: Kanten (erzeugend)



Kanten (erzeugend):
Von Transition zu Stelle.
Marken werden Stellen hinzugefügt.

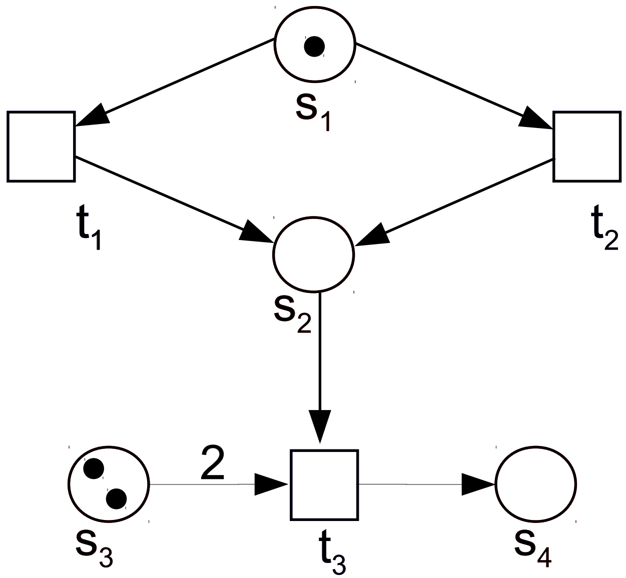
Vor- und Nachbereich einer Transition

- **Vorbereich** einer **Transition:** Menge der Stellen, die über ausgehende Kante mit Transition verbunden sind.

  Vorbereich von t: $\bullet t = \{ s \in S \mid (s,t) \in F \}$

- **Nachbereich** einer **Transition:** Menge der Stellen, die über eingehende Kante mit Transition verbunden sind.

  Nachbereich von t: $t\bullet = \{ s \in S \mid (t,s) \in F \}$



$\bullet t_1 = \bullet t_2 = \{s_1\}$, $\bullet t_3 = \{s_2, s_3\}$

$t_1\bullet = t_2\bullet = \{s_2\}$, $t_3\bullet = \{s_4\}$

Aktivierte Transition: Definition

Informell: Transition ist **aktiviert**, wenn sie

- die geforderte Anzahl Marken erhalten kann und wenn
- **die Folgemarkierung** die freigesetzten Marken aufnehmen **kann**,

d.h. wenn

- alle Stellen im Vorbereich der Transition ausreichend Marken besitzen (gemäß Kantengewicht der konsumierenden Kante) und
- Kapazitäten aller Stellen im Nachbereich der Transition groß genug sind (gemäß Kantengewicht der erzeugenden Kante)

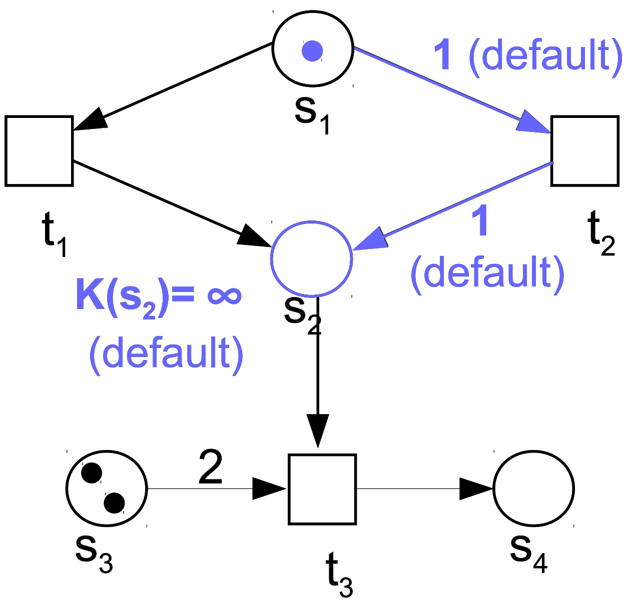
Formal: Transition t ist **aktiviert** genau dann, wenn:

$$\underline{\forall s \in \bullet t : M(s) \geq W(s,t)} \wedge \underline{\forall s' \in t \bullet : M(s') + W(t,s') \leq K(s')}$$

Aktivierte Transition: Beispiel

Transition ist **aktiviert**, wenn sie

- die geforderte Anzahl Marken erhalten kann und wenn
- **die Folgemarkierung** die freigesetzten Marken aufnehmen **kann**.



$\bullet t_2 = \{s_1\}$ und $t_2 \bullet = \{s_2\}$

Sei $M_0(s_1)=1$ und $M_0(s_3)=2$ und $M_0(s_2)=M_0(s_4)=0.$

Die Transition t_2 ist **aktiviert**, da t_2 die benötigte Marke der Kante (t_2, s_2) von s_1 erhalten kann und die Kapazität von s_2 ausreicht, um die Marke der Kante (t_2, s_2) aufzunehmen.

$$\forall s \in \bullet t_2 : M(s) \geq W(s, t_2)$$

$$\forall s' \in t_2 \bullet : M(s') + W(t_2, s') \leq K(s')$$

Ausführung eines Petrinetzes: Schalten einer Transition

Bei **Ausführung** eines Petrinetzes wird jeweils **eine** der aktivierten Transitionen von Zustand M_x nach Zustand M_{x+1} geschaltet:

- Benötigte Marken auf **Vorgänger**-Stellen werden **konsumiert.**
- Produzierte Marken auf **Nachfolger**-Stellen abgelegt.

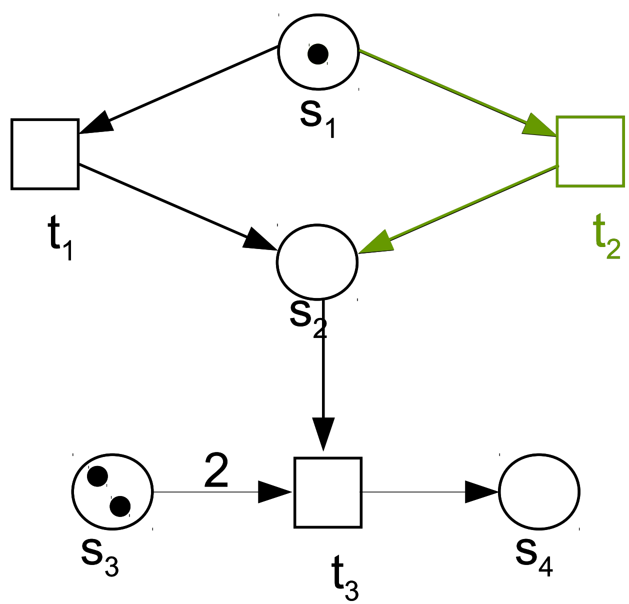
Anzahl konsumierter / produzierter Marken jeweils gemäß **Kantenvielfachheit**:
→ **Gesamtanzahl** Marken im Netz kann sich **verändern**.

Folgemarkierung (= Folgezustand): Erhältlich durch Schalten jeweils **genau einer** Transition (**nicht-deterministische Auswahl**).

Schalten einer Transition: Beispiel

Schalten einer aktivierten Transitionen:

- Benötigte Marken auf **Vorgänger**-Stellen werden **konsumiert.**
- Produzierte Marken auf **Nachfolger**-Stellen abgelegt.

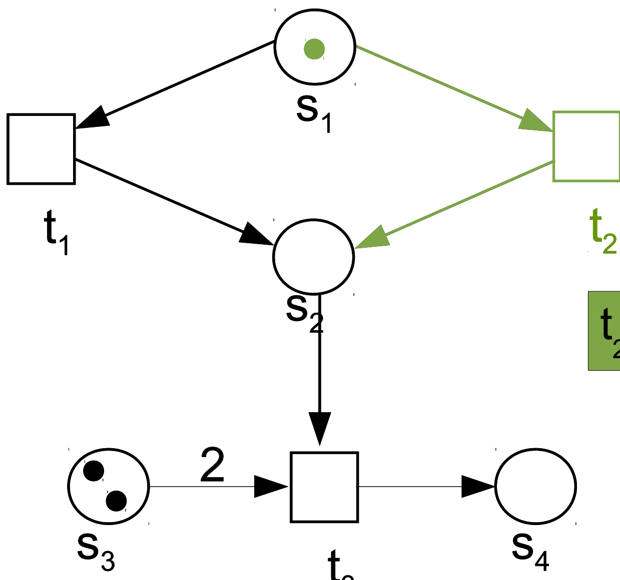


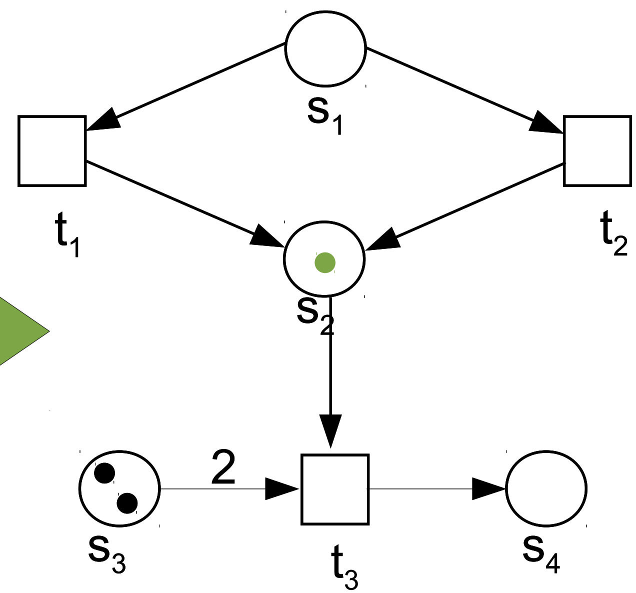
$M_0(s_1)= 1,\quad M_0(s_2)= 0,$
$M_0(s_3)= 2,\quad M_0(s_4)= 0$

Schalten einer Transition: Beispiel

Schalten einer aktivierten Transitionen:

- Benötigte Marken auf **Vorgänger**-Stellen werden **konsumiert.**
- Produzierte Marken auf **Nachfolger**-Stellen abgelegt.





$M_0(s_1)= 1$, $M_0(s_2)= 0$,
$M_0(s_3)= 2$, $M_0(s_4)= 0$

[Alternativ kann t1 schalten, mit demselben Ergebnis (in diesem Fall).]

$M_1(s_1)= 0$, $M_1(s_2)= 1$,
$M_1(s_3)= 2$, $M_1(s_4)= 0$

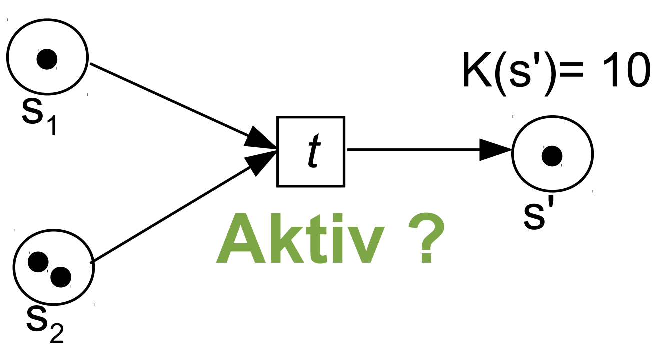
Aktiviertheit von Transitionen: Weiteres Beispiel

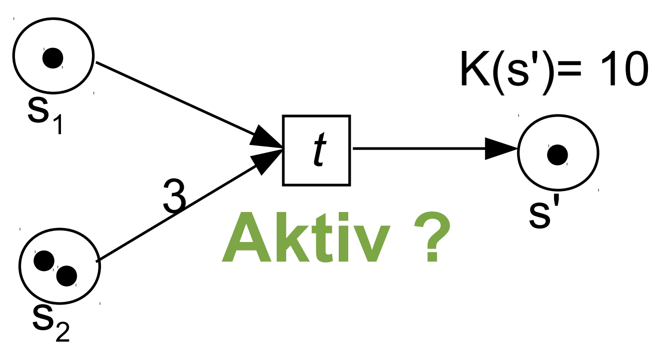
Transition t ist **aktiviert**, wenn:

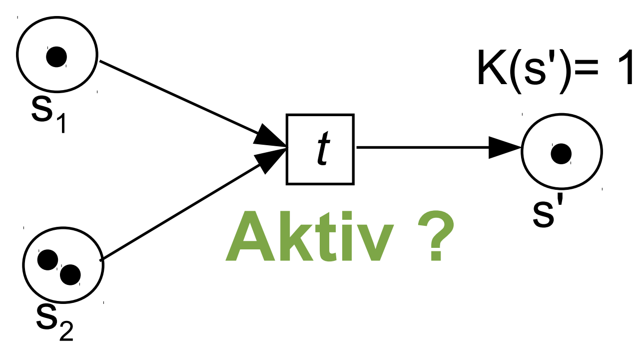
$$\forall s \in \bullet t : M(s) \geq W(s,t) \wedge \forall s' \in t \bullet : M(s') + W(t,s') \leq K(s')$$

- *W(s,t)*: Gewicht des Bogens von s nach t
- *M(s)*: Anzahl Marken in s
- *K(s)*: Kapazität von s

- W(t,s'): Gewicht des Bogens von t nach s'
- M(s'): Anzahl Marken in s'
- K(s'): Kapazität von s'







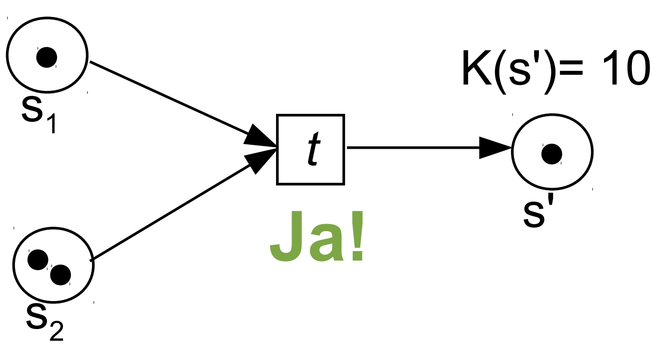
Aktiviertheit von Transitionen: Weiteres Beispiel

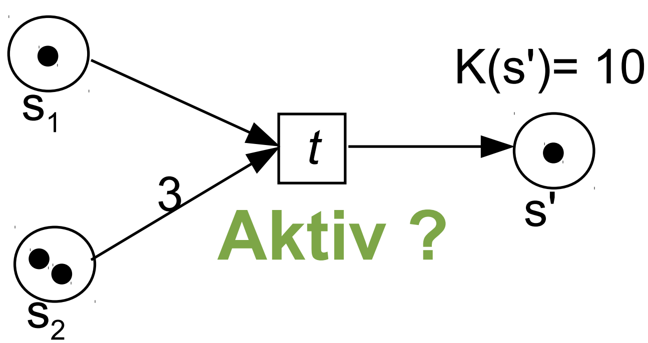
Transition t ist **aktiviert**, wenn:

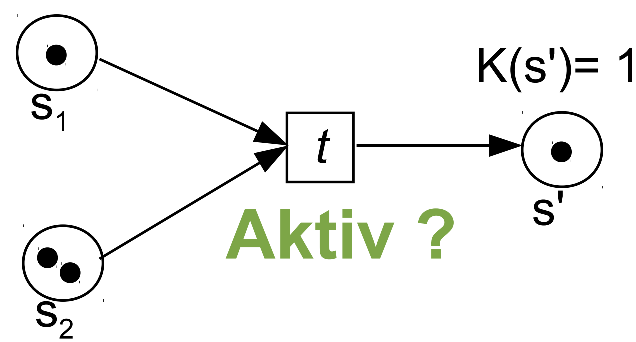
$$\forall s \in \bullet t : M(s) \geq W(s,t) \wedge \forall s' \in t \bullet : M(s') + W(t,s') \leq K(s')$$

- *W(s,t)*: Gewicht des Bogens von s nach t
- *M(s)*: Anzahl Marken in s
- *K(s)*: Kapazität von s

- W(t,s'): Gewicht des Bogens von t nach s'
- M(s'): Anzahl Marken in s'
- K(s'): Kapazität von s'





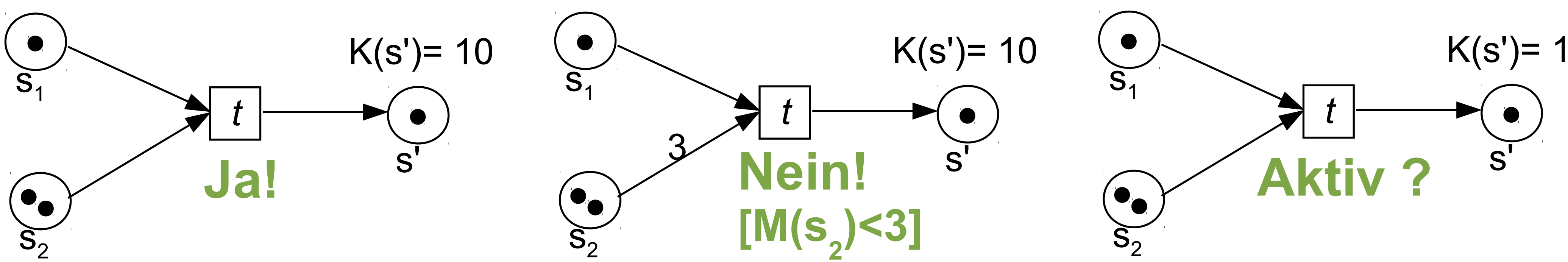


Aktiviertheit von Transitionen: Weiteres Beispiel

Transition t ist **aktiviert**, wenn:

$$\forall s \in \bullet t : M(s) \geq W(s,t) \wedge \forall s' \in t \bullet : M(s') + W(t,s') \leq K(s')$$

- *W(s,t)*: Gewicht des Bogens von s nach t
- *M(s)*: Anzahl Marken in s
- *K(s)*: Kapazität von s
- W(t,s'): Gewicht des Bogens von t nach s'
- M(s'): Anzahl Marken in s'
- K(s'): Kapazität von s'



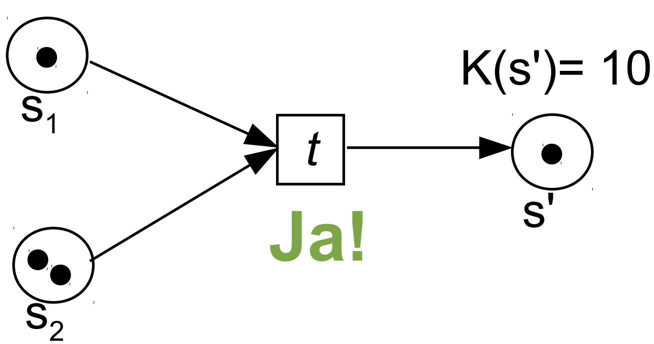
Aktiviertheit von Transitionen: Weiteres Beispiel

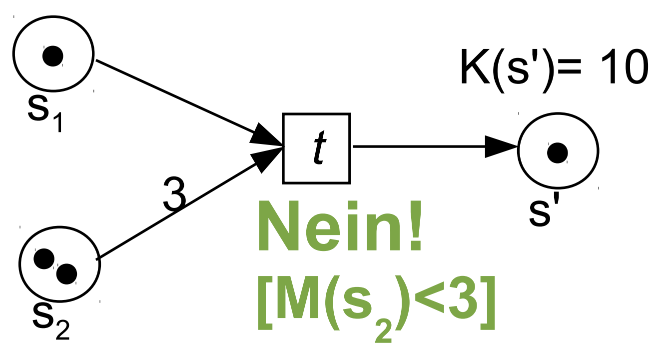
Transition t ist **aktiviert**, wenn:

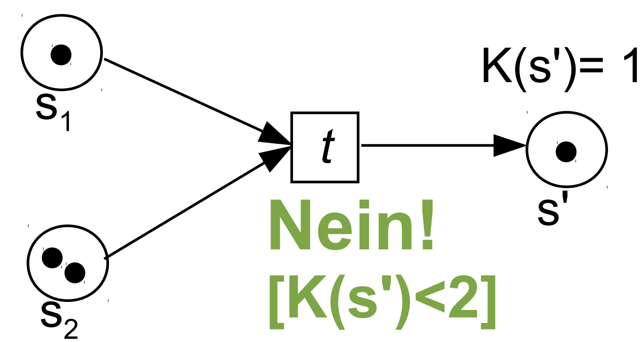
$$\forall s \in \bullet t : M(s) \geq W(s,t) \wedge \forall s' \in t \bullet : M(s') + W(t,s') \leq K(s')$$

- *W(s,t)*: Gewicht des Bogens von s nach t
- *M(s)*: Anzahl Marken in s
- *K(s)*: Kapazität von s

- W(t,s'): Gewicht des Bogens von t nach s'
- M(s'): Anzahl Marken in s'
- K(s'): Kapazität von s'







Schalten von Transition: Zur Diskussion

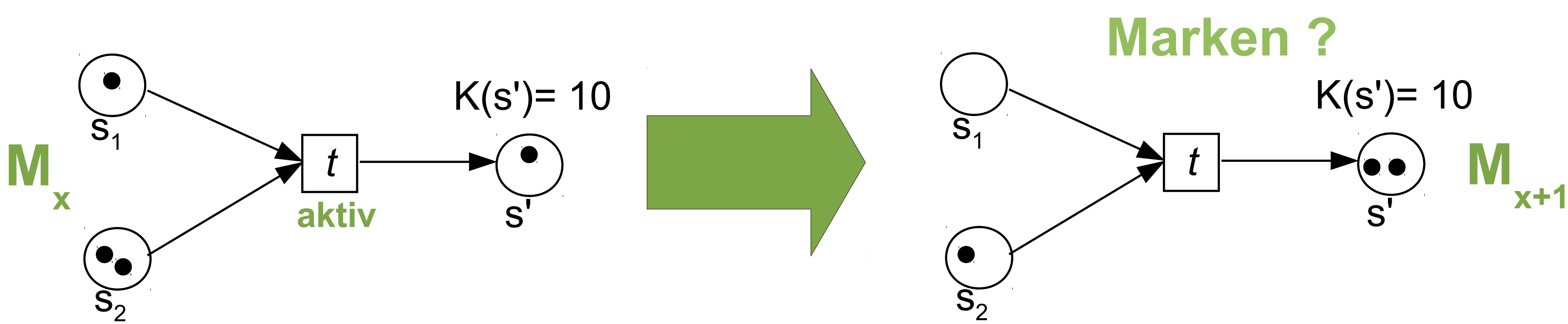
Eine der aktivierten Transitionen wird von Zustand M_x nach Zustand M_{x+1} geschaltet (**nicht-deterministische Auswahl**):

- Benötigte Marken auf **Vorgänger**-Stellen werden **konsumiert.**
- Produzierte Marken auf **Nachfolger**-Stellen abgelegt.

Anzahl konsumierter / produzierter Marken jeweils gemäß **Bogenvielfalt**:

→ **Gesamtanzahl** Marken im Netz kann sich **verändern**.

Folgemarkierung (= -zustand): Schalten jeweils **genau einer** Transition.



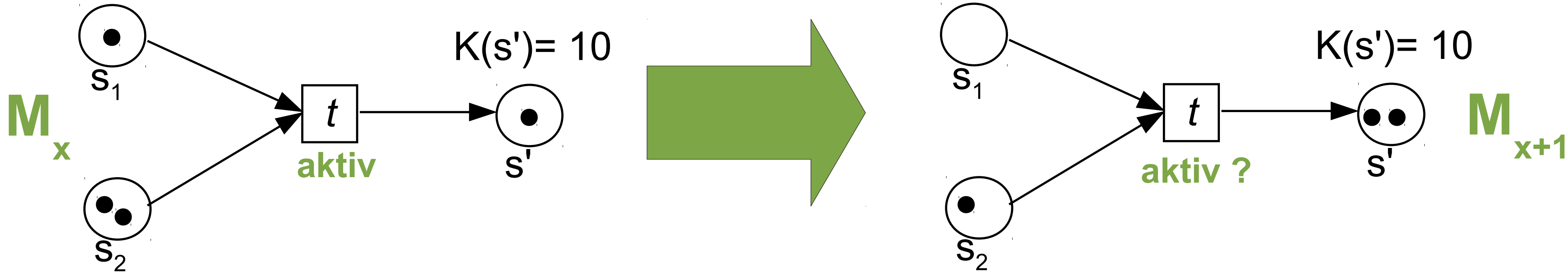
Schalten von Transition: Zur Diskussion

Eine der aktivierten Transitionen wird von Zustand M_x nach Zustand M_{x+1} geschaltet (**nicht-deterministische Auswahl**):

- Benötigte Marken auf **Vorgänger**-Stellen werden **konsumiert.**
- Produzierte Marken auf **Nachfolger**-Stellen abgelegt.

Anzahl konsumierter / produzierter Marken jeweils gemäß **Bogenvielfalt**:
→ **Gesamtanzahl** Marken im Netz kann sich **verändern**.

Folgemarkierung (= -zustand): Schalten jeweils **genau einer** Transition.



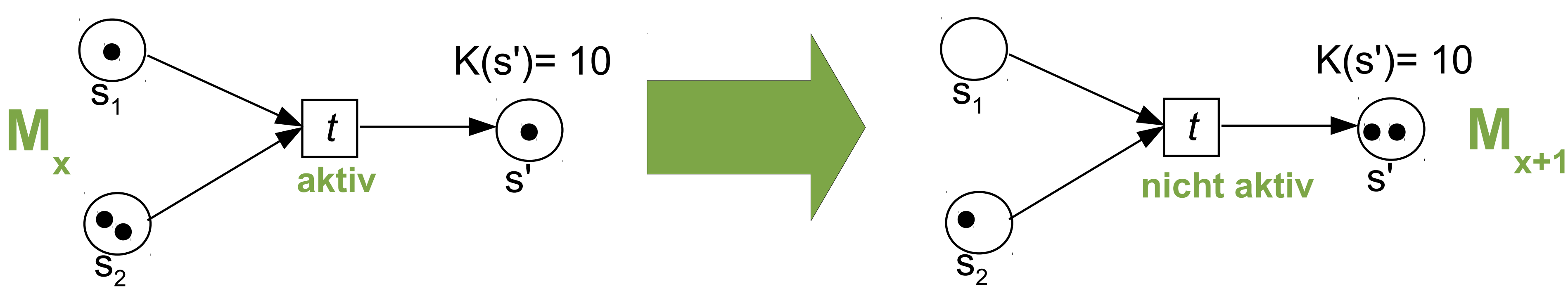
Schalten von Transition: Zur Diskussion

Eine der aktivierten Transitionen wird von Zustand M_x nach Zustand M_{x+1} geschaltet (**nicht-deterministische Auswahl**):

- Benötigte Marken auf **Vorgänger**-Stellen werden **konsumiert.**
- Produzierte Marken auf **Nachfolger**-Stellen abgelegt.

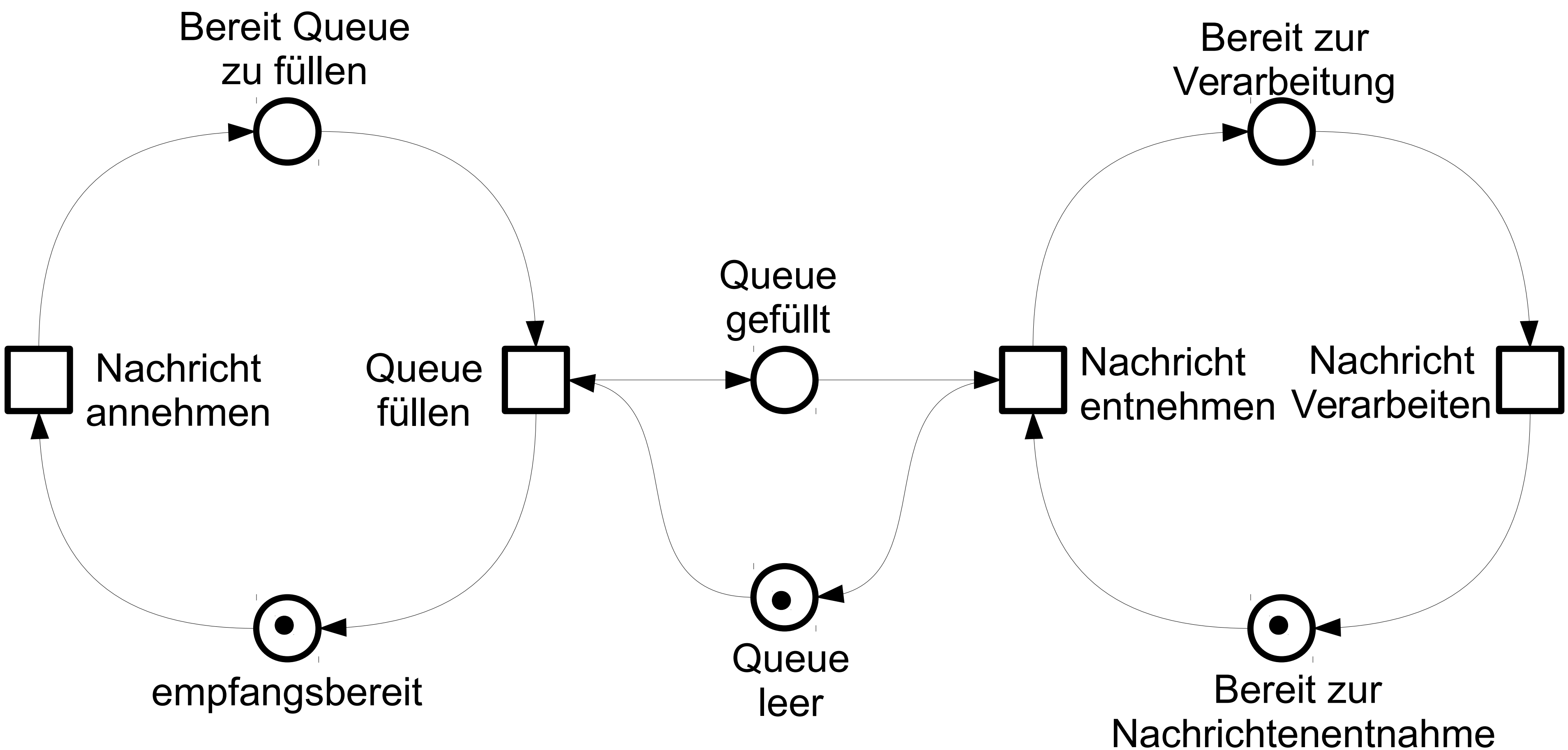
Anzahl konsumierter / produzierter Marken jeweils gemäß **Bogenvielfalt**:
→ **Gesamtanzahl** Marken im Netz kann sich **verändern**.

Folgemarkierung (= -zustand): Schalten jeweils **genau einer** Transition.

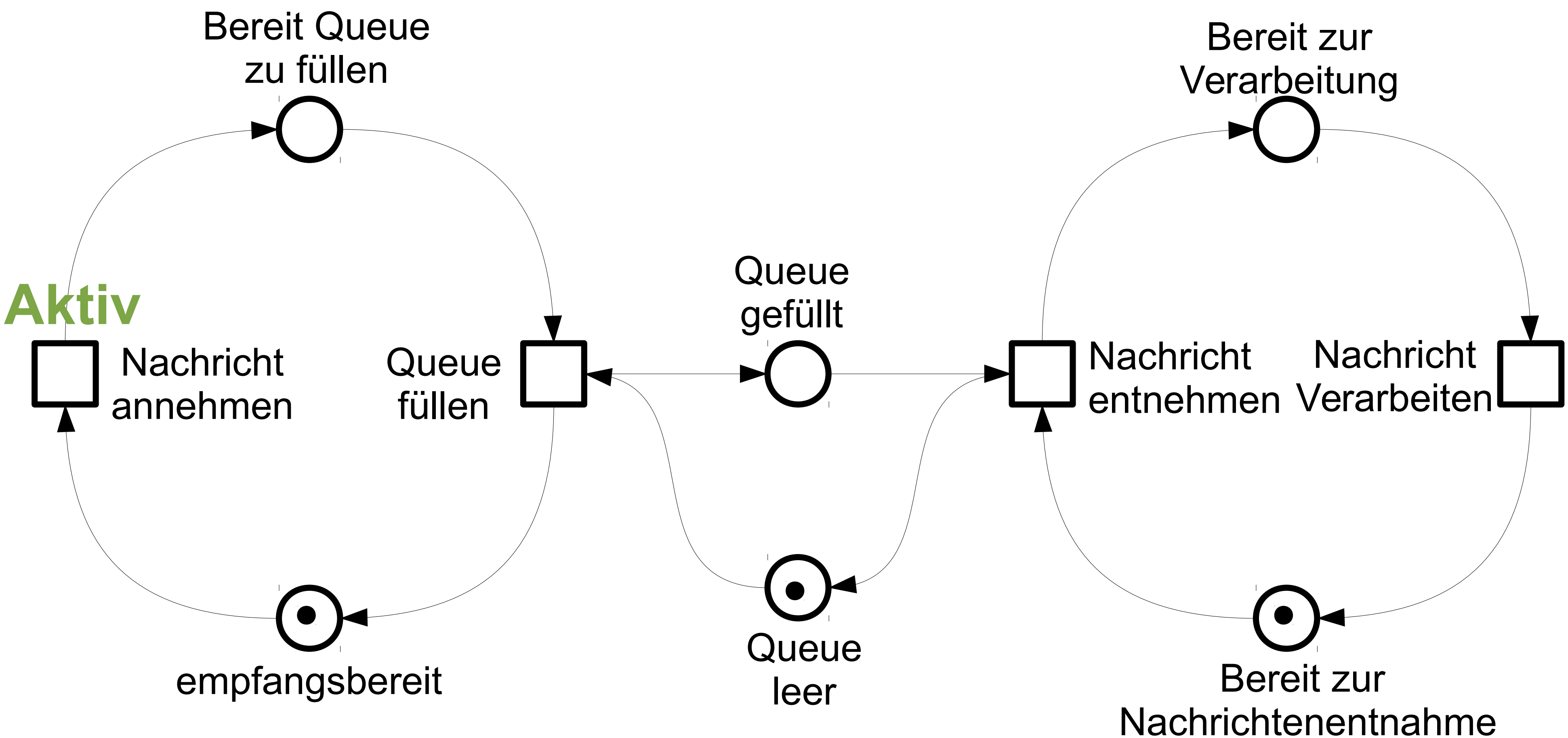


Petrinetz Ablauf: Beispiel Nachrichten-Queue (M_0)

Welche Transition(en) aktiviert ?

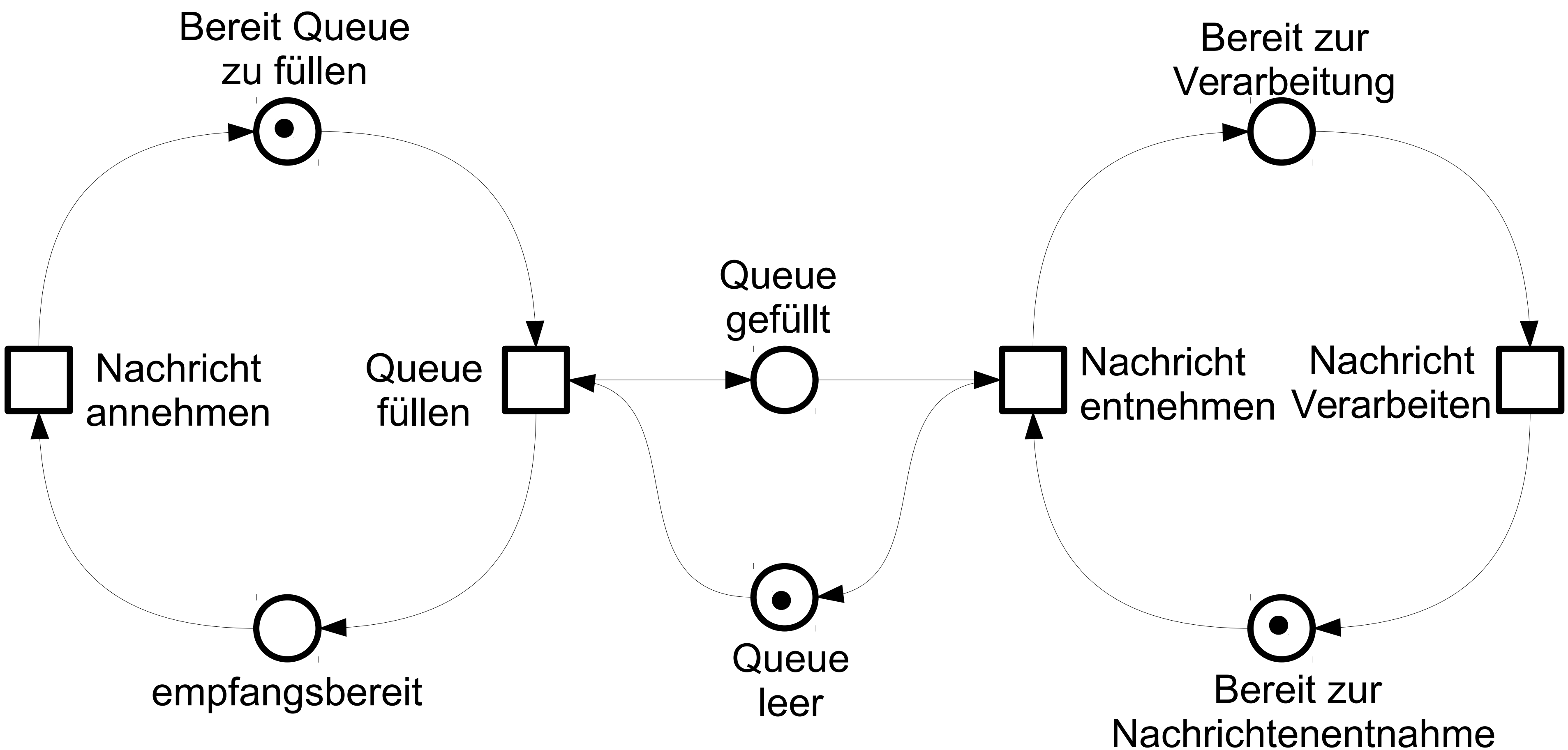


Nächster Zustand ?



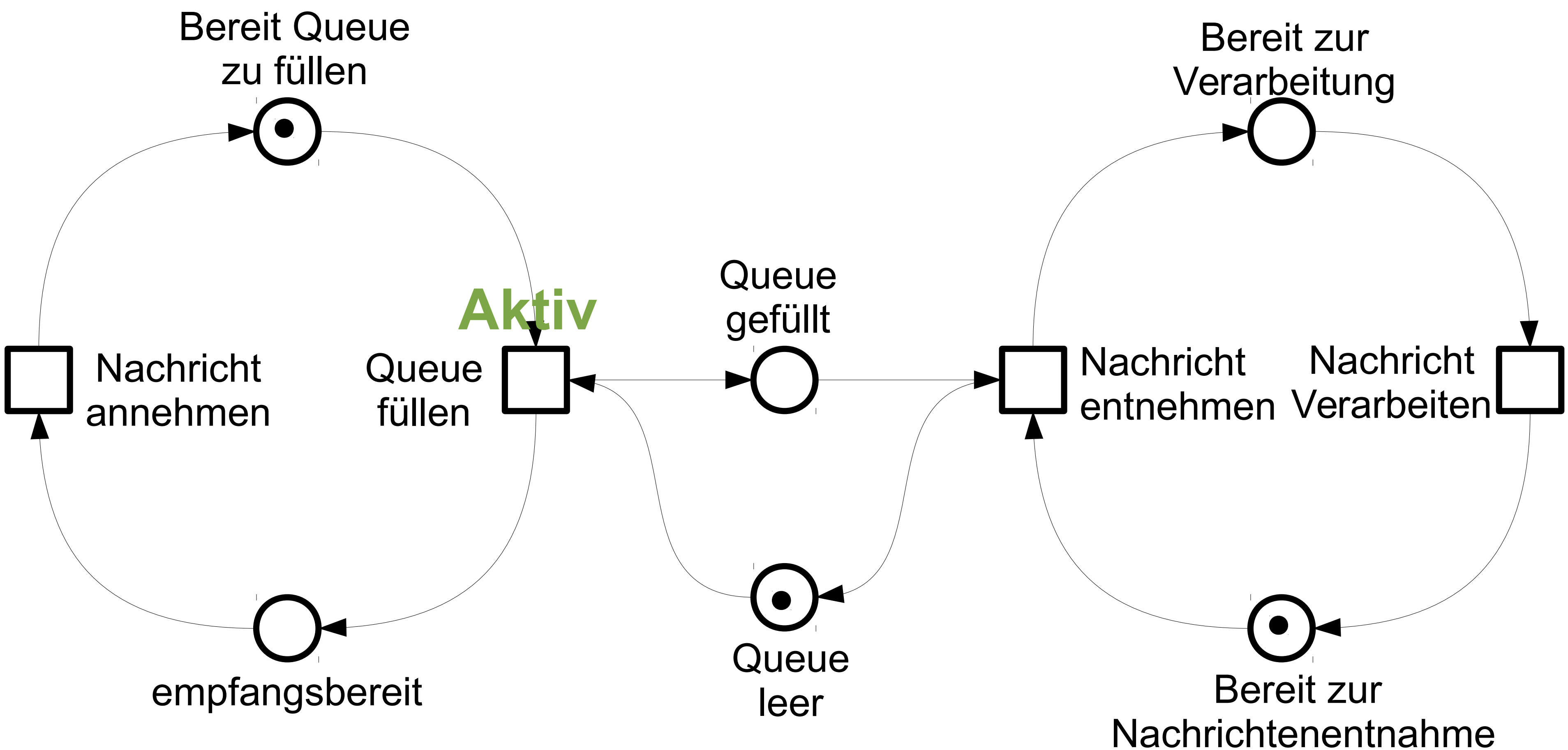
Ablauf: Beispiel (M_1)

Welche Transition(en) aktiviert ?



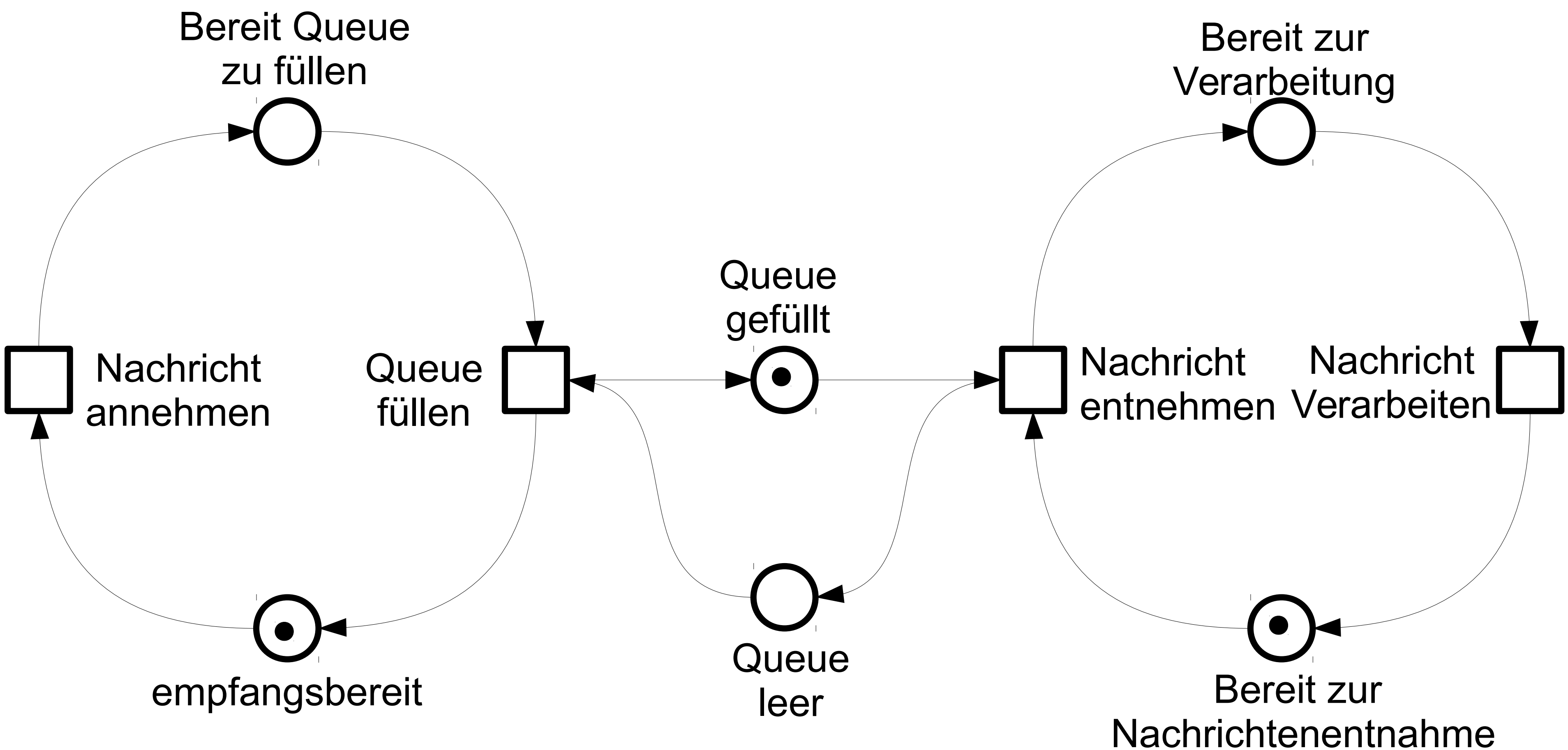
Ablauf: Beispiel (M_1 => M_2)

Nächster Zustand ?



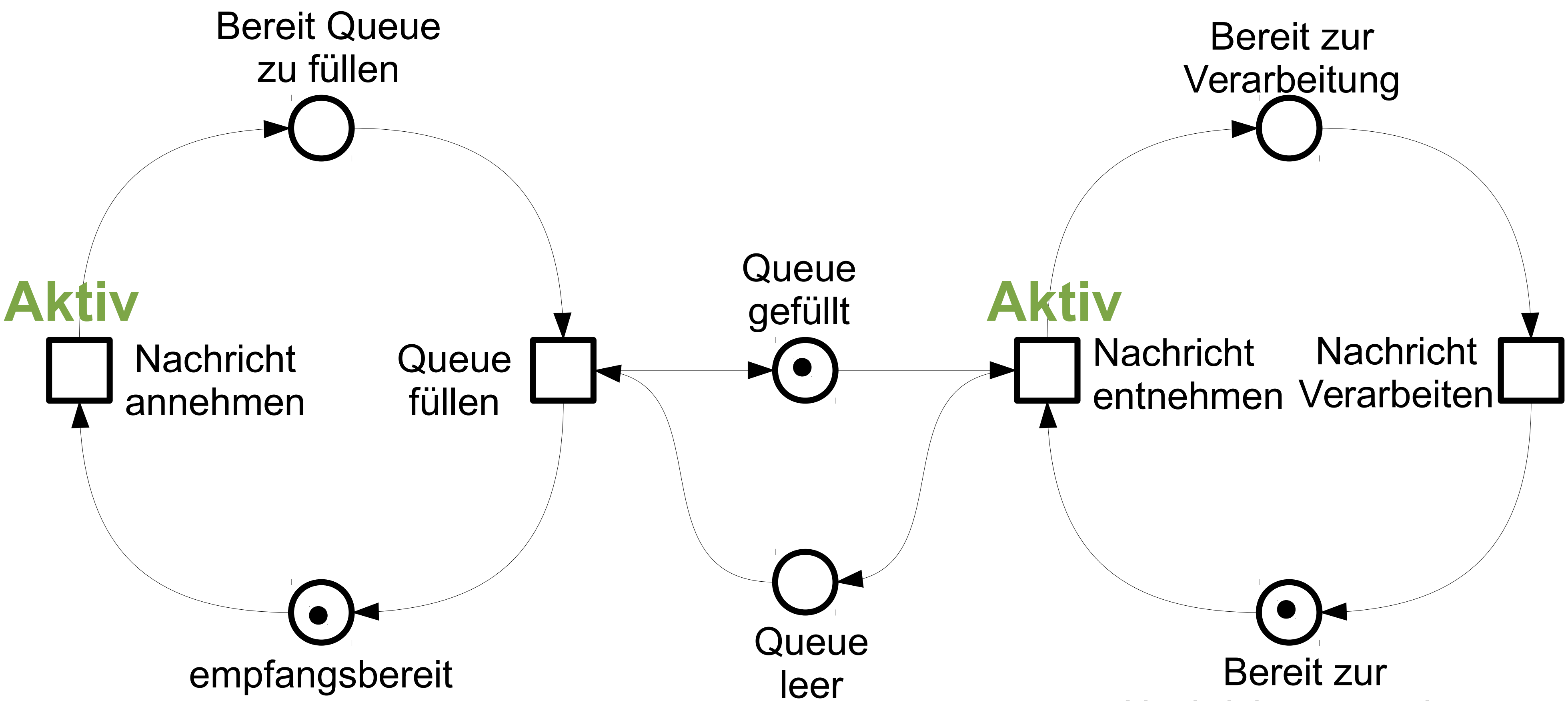
Ablauf: Beispiel (M_2)

Welche Transition(en) aktiviert ?



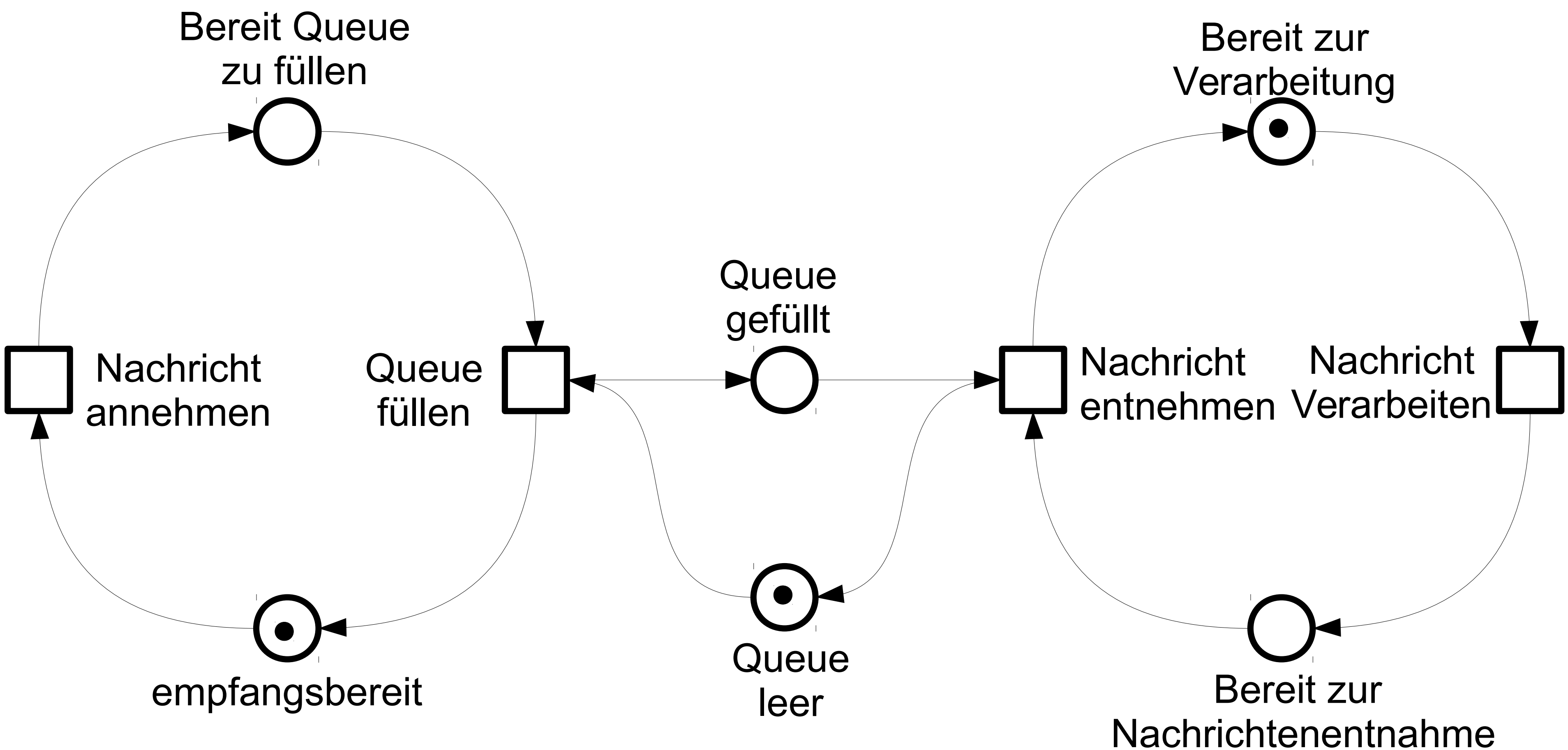
Ablauf: Beispiel (M_2 => M_3)

Nächster Zustand ?



(Transition „Nachricht entnehmen“ ausgeführt, ergibt M_3;
alternativ: „Nachricht annehmen“ ausführbar, ergibt anderen Zustand M_3'.)

Ablauf: Beispiel (M_3)



Frage: Größe der Queue

Gibt es eine obere Grenze, wie viele Nachrichten gleichzeitig in dieser Queue enthalten sein können ?

Frage: Größe der Queue

Gibt es eine obere Grenze, wie viele Nachrichten gleichzeitig in dieser Queue enthalten sein können ?

Antwort:

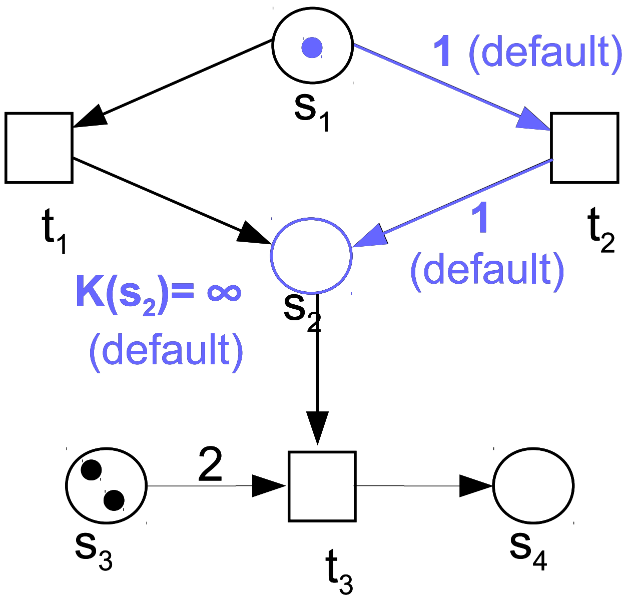
In der Queue kann höchstens **eine** Nachricht enthalten sein:

- Transition „Queue füllen“ kann nur ausgeführt werden, wenn Stelle „Queue leer“ eine Marke hat.

Erreichbarkeit: Notation und Definition (1)

- **M [t>** : bei Markierung M ist Transition t aktiviert
  ([> symbolisiert Pfeil)
  <u>Beispiel:</u>

M_0 [t_2>

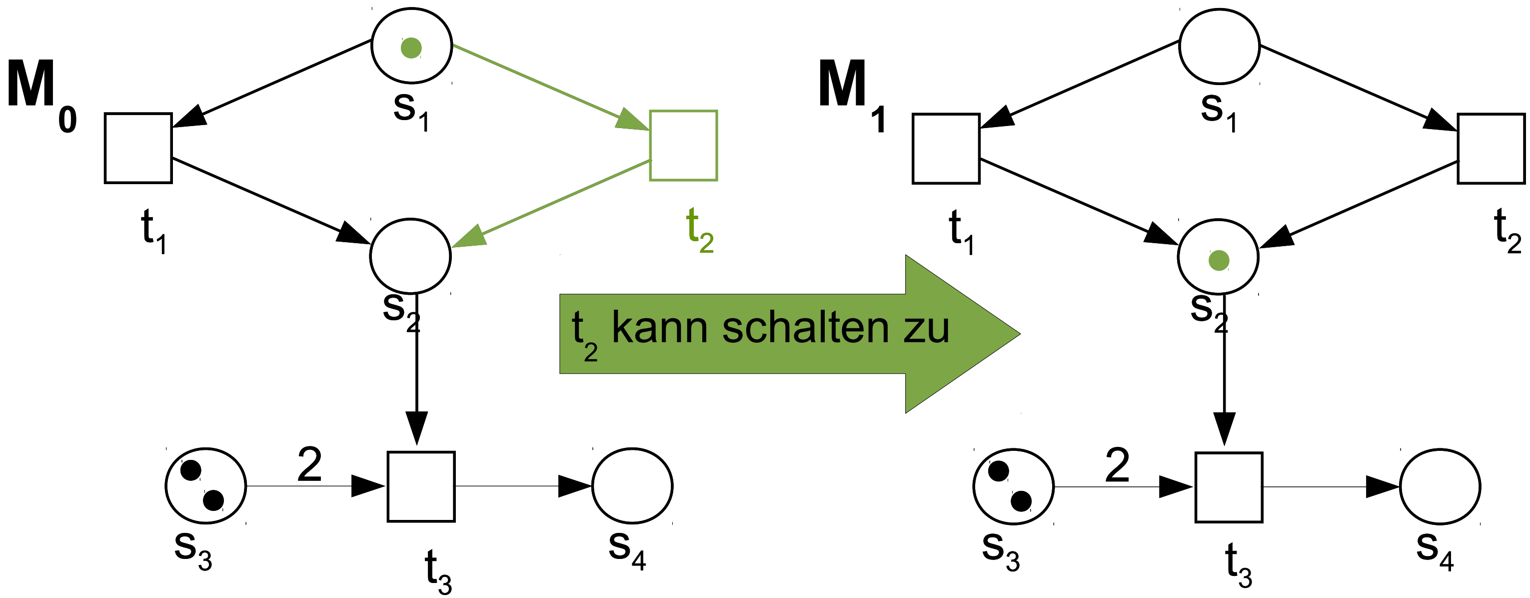


Erreichbarkeit: Notation und Definition

- **M [t>** : bei Markierung M ist Transition t aktiviert ([> symbolisiert Pfeil)
- **M [t> M' :** M' ist direkte Folgemarkierung zur Markierung M nach Schaltung von Transition t

  Beispiel:

  M_0 [t_2> M_1



Erreichbarkeit: Notation und Definition

- **M [t>** : bei Markierung M ist Transition t aktiviert ([> symbolisiert Pfeil)
- **M [t> M' :** M' ist direkte Folgemarkierung zur Markierung M nach Schaltung von Transition t
- **M [w>** : Liste von Transitionen w=[t1,t2,...,tn] ist iterativ aktiviert unter Markierung M, d.h.: M [t1> M1 [t2> M2 … [tn> Mn

  Queue-Beispiel:

  M_0 [Nachricht annehmen**> M_1 [**Queue füllen**> M_2**

  [Nachricht entnehmen**> M_3**

  w=[Nachricht annehmen, Queue füllen, Nachricht entnehmen]

  damit: **M_0 [w>**

Erreichbarkeit: Notation und Definition

- **M [t>** : bei Markierung M ist Transition t aktiviert ([> symbolisiert Pfeil)
- **M [t> M' :** M' ist direkte Folgemarkierung zur Markierung M nach Schaltung von Transition t
- **M [w>** : Liste von Transitionen w=[t1,t2,...,tn] ist iterativ aktiviert unter Markierung M, d.h.: M [t1> M1 [t2> M2 … [tn> Mn
- **M [{t1, t2, ..., tn}>** : Liste von Transitionen [t1,t2,...,tn] ist in beliebiger Schaltungsreihenfolge iterativ aktiviert unter Markierung M (= alle Permutationen als Schaltfolgen aktiviert; genannt "**nebenläufig aktiviert**")

  Queue-Beispiel:

  M_2 [{Nachricht entnehmen, Nachricht annehmen}**>**

Erreichbarkeit: Notation und Definition

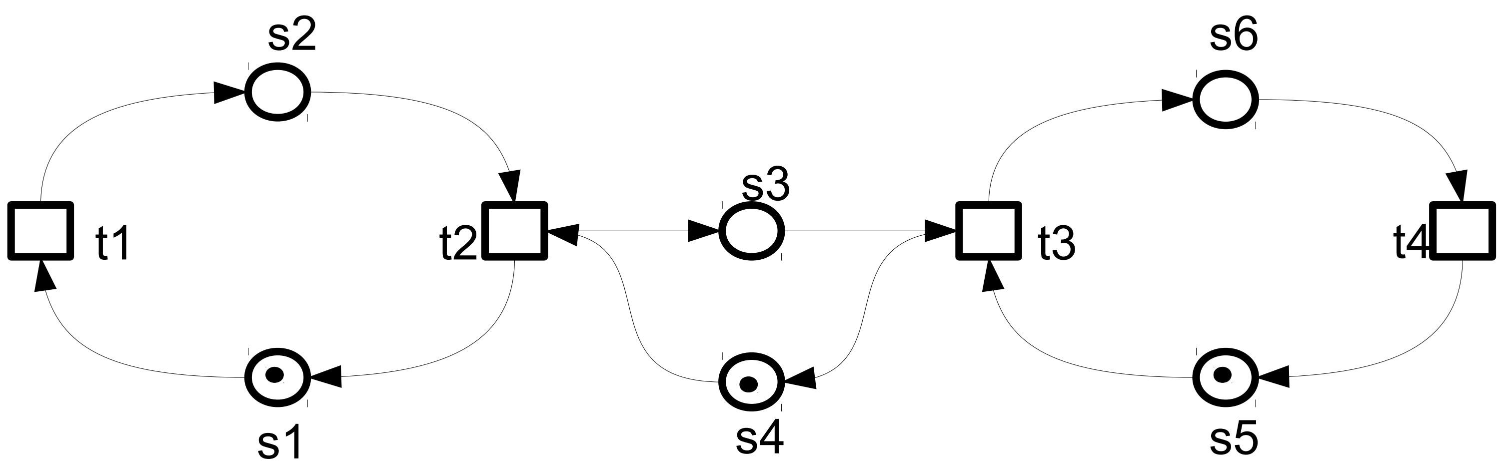
- **M [t>** : bei Markierung M ist Transition t aktiviert ([> symbolisiert Pfeil)
- **M [t> M' :** M' ist direkte Folgemarkierung zur Markierung M nach Schaltung von Transition t
- **M [w>** : Liste von Transitionen w=[t1,t2,...,tn] ist iterativ aktiviert unter Markierung M, d.h.: M [t1> M1 [t2> M2 … [tn> Mn
- **M [{t1, t2, ..., tn}>** : Liste von Transitionen [t1,t2,...,tn] ist in beliebiger Schaltungsreihenfolge iterativ aktiviert unter Markierung M (= alle Permutationen als Schaltfolgen aktiviert; genannt "**nebenläufig aktiviert**")
- **$[M_0>$** := $\{M \mid \exists\, w \in T^* \text{ mit } M_0\,[w> M\}$ (**Erreichbarkeitsmenge** des Systems; die Markierungen $M \in [M_0>$ heißen **erreichbar**)

Queue-Beispiel: Erreichbarkeitsmenge: $\mathbf{[M_0> = \{M_1, M_2, M_3, M_3', M_4, ...\}}$

Petrinetz Ablauf: Erreichbarkeitstabelle

Nr.	s1	s2	s3	s4	s5	s6	Schaltungen
M0	1	0	0	1	1	0	
M1							
M2							
M3							
M3'							

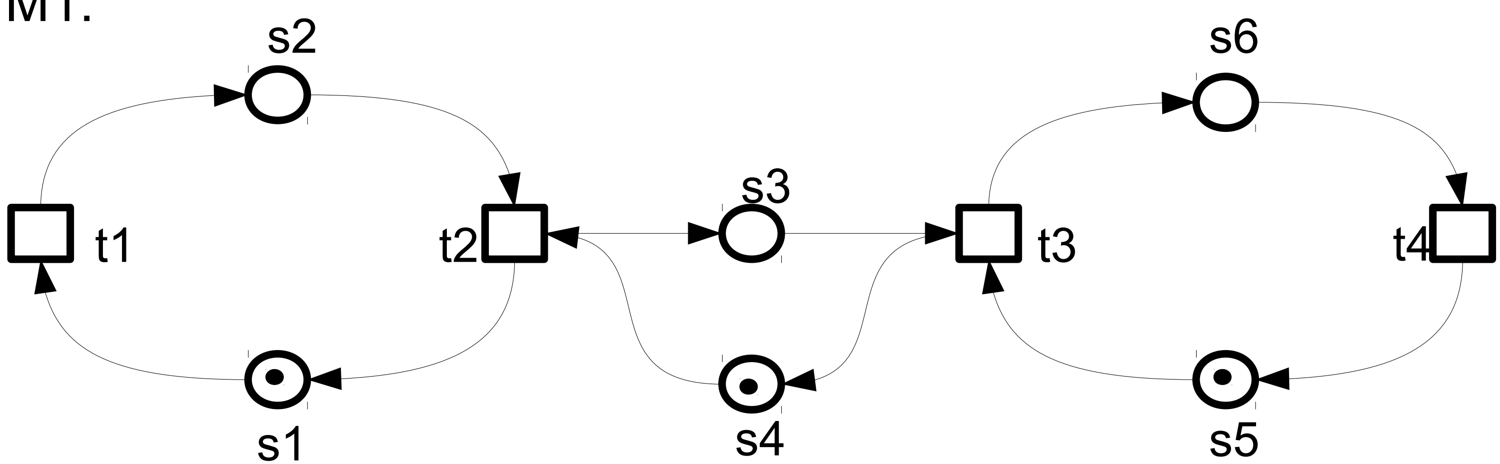
M0:



Petrinetz Ablauf: Erreichbarkeitstabelle

Nr.	s1	s2	s3	s4	s5	s6	Schaltungen
M0	1	0	0	1	1	0	t1 --> M1
M1	0	1	0	1	1	0	
M2							
M3							
M3'							

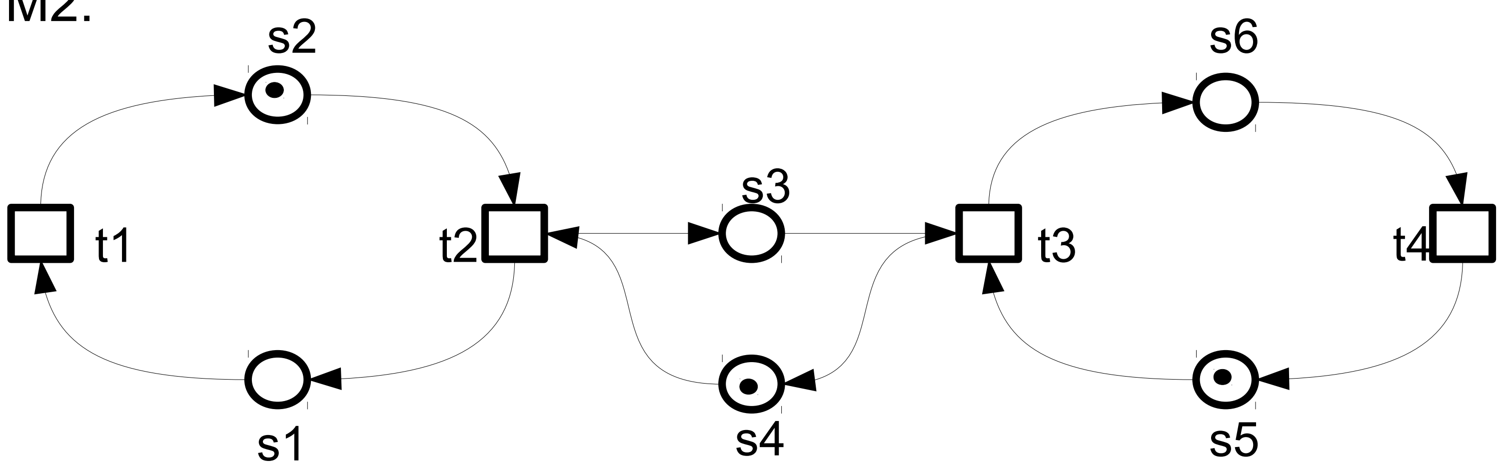
M0 [t1> M1:



Petrinetz Ablauf: Erreichbarkeitstabelle

Nr.	s1	s2	s3	s4	s5	s6	Schaltungen
M0	1	0	0	1	1	0	t1 --> M1
M1	0	1	0	1	1	0	t2 --> M2
M2	1	0	1	0	1	0	
M3							
M3'							

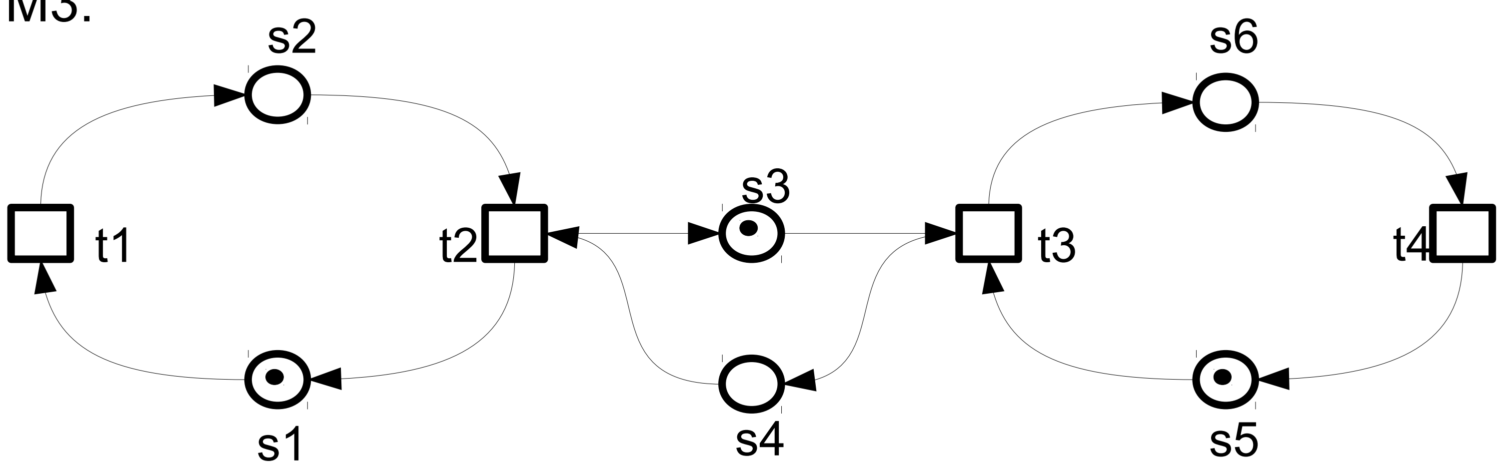
M1 [t2> M2:



Petrinetz Ablauf: Erreichbarkeitstabelle

Nr.	s1	s2	s3	s4	s5	s6	Schaltungen
M0	1	0	0	1	1	0	t1 --> M1
M1	0	1	0	1	1	0	t2 --> M2
M2	1	0	1	0	1	0	t3 --> M3
M3	1	0	0	1	0	1	
M3'							

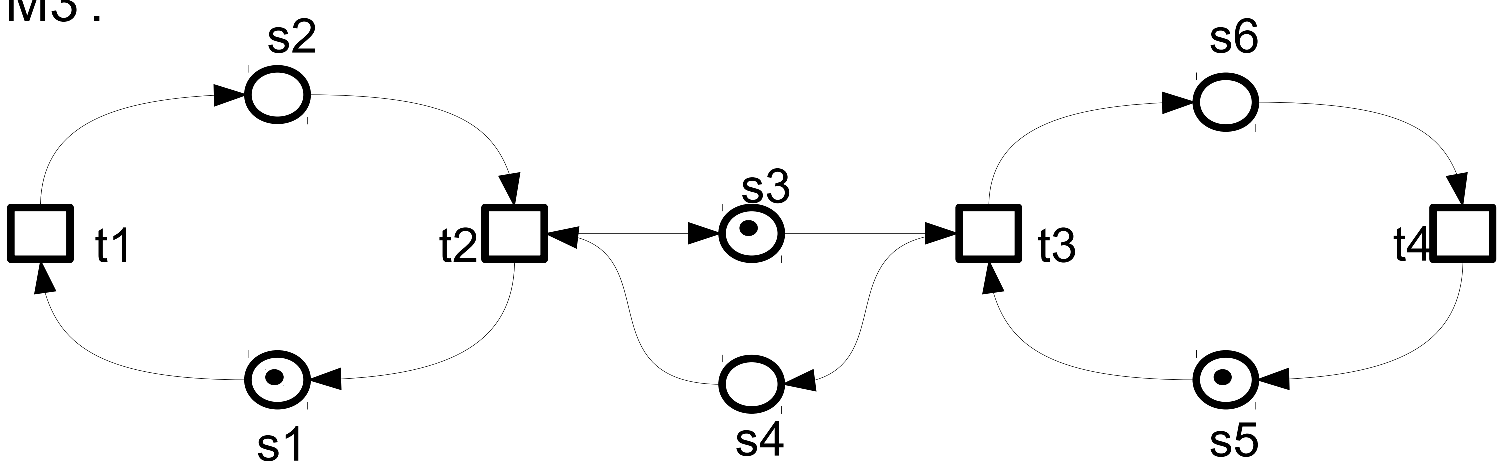
M2 [t3> M3:



Petrinetz Ablauf: Erreichbarkeitstabelle

Nr.	s1	s2	s3	s4	s5	s6	Schaltungen
M0	1	0	0	1	1	0	t1 --> M1
M1	0	1	0	1	1	0	t2 --> M2
M2	1	0	1	0	1	0	t3 --> M3 t1 --> M3'
M3	1	0	0	1	0	1	t1 --> M4 ...
M3'	0	1	1	0	1	0	t3 --> M4 ...

M2 [t1> M3':



Erreichbarkeitsalgorithmus (breadth-first; vgl. obiges Beispiel)

Eingabe: Petrinetz. Ausgabe: Erreichbarkeitstabelle (vgl. vorletzte Folie).

1. Trage in ein Schema mit Spalten „Markierungsnummer", „Markierung" und „Schaltungen" **Anfangsmarkierung M_0** ein.
2. In **aktueller Markierung M_i** für jede **Transition t**: aktiviert ?
   - Falls **t** aktiviert: Berechne Folgemarkierung.
     - Folgemarkierung bereits eine **Markierung M_j** ?
     - Wenn nicht: Benenne Folgemarkierung **M_j** (für ein neues j>i) und lege neue Zeile in der Tabelle für **M_j** an.
   - In beiden Fällen: Trage **M_i [t>M_j** in Zeile **M_i**, Spalte „Schaltungen" ein.
3. **M_i** erledigt, falls **alle** Transitionen überprüft.
4. Alle eingetragenen Markierungen erledigt ?
   - **Ja**: Erreichbarkeitsanalyse abgeschlossen
   - **Nein**: Überprüfe die nächste Markierung und fahre bei 2 fort.

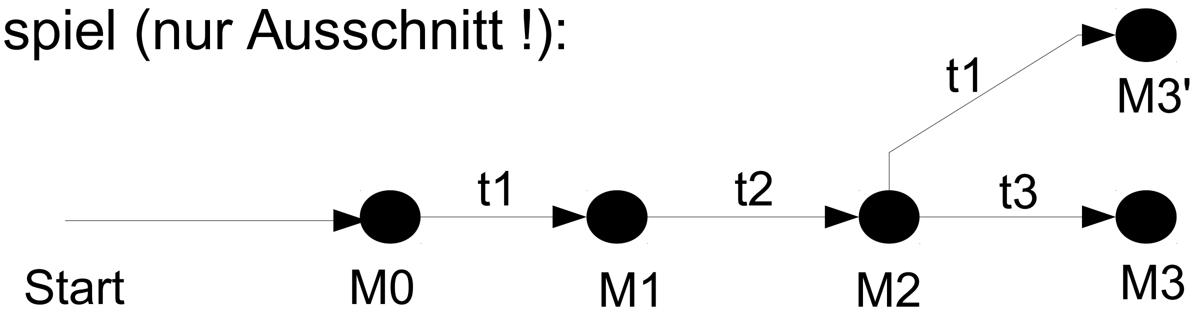
Erreichbarkeit: Tabelle als Graph

Nr.	s1	s2	s3	s4	s5	s6	Schaltungen
M0	1	0	0	1	1	0	t1 --> M1
M1	0	1	0	1	1	0	t2 --> M2
M2	1	0	1	0	1	0	t3 --> M3 t1 --> M3'
M3	1	0	0	1	0	1	
M3'	0	1	1	0	1	0	

Erreichbarkeitstabelle oft als Graph dargestellt:

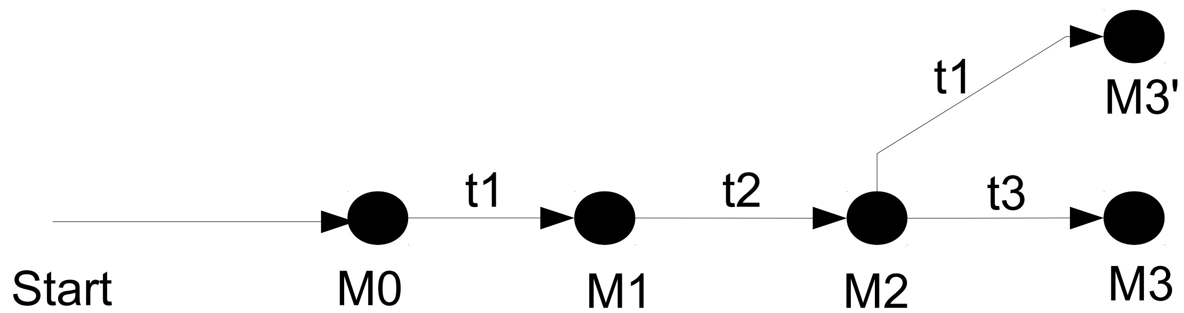
- Knoten: Zustände (linke Spalte; ggf. inkl. Markierungsbelegungen)
- Kanten: Schaltungen (rechte Spalte)

Obiges Beispiel (nur Ausschnitt !):



Erreichbarkeit: Tabelle als Graph

Nr.	s1	s2	s3	s4	s5	s6	Schaltungen
M0	1	0	0	1	1	0	t1 --> M1
M1	0	1	0	1	1	0	t2 --> M2
M2	1	0	1	0	1	0	t3 --> M3 t1 --> M3'
M3	1	0	0	1	0	1	
M3'	0	1	1	0	1	0	



Erzeugtes Event-Log

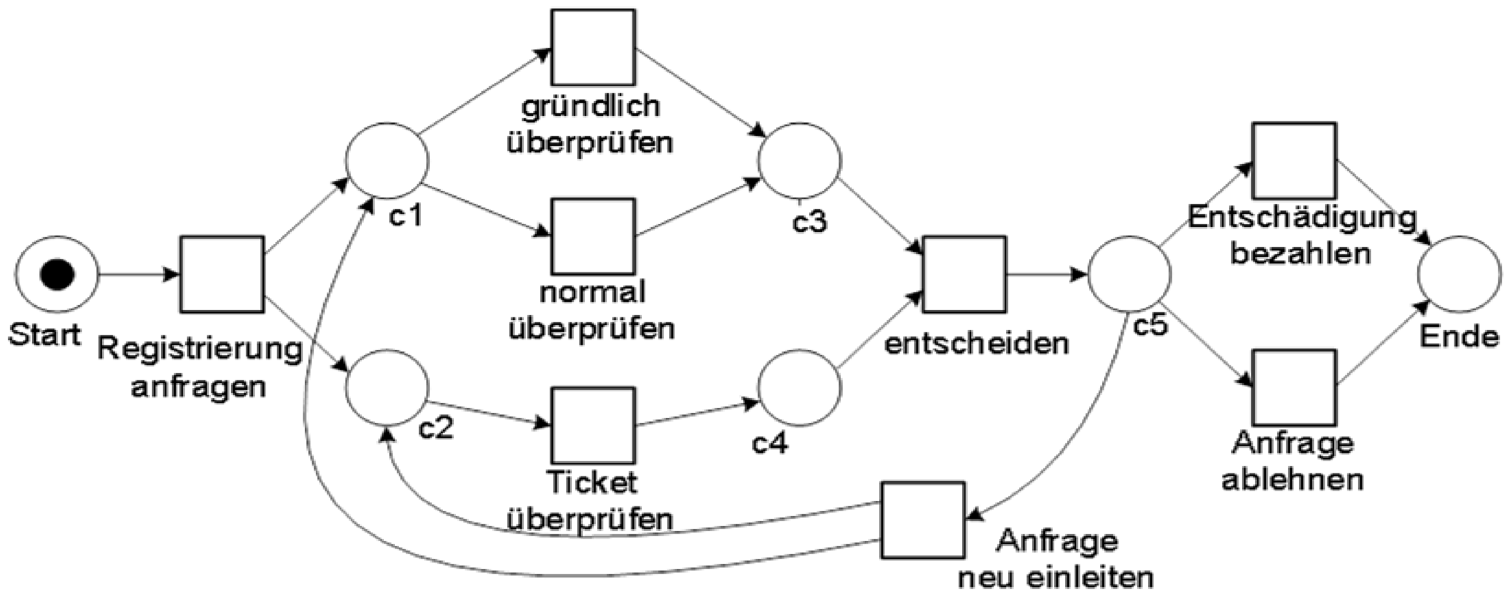
(= Menge der Folgen der ausgeführten Transitionen):

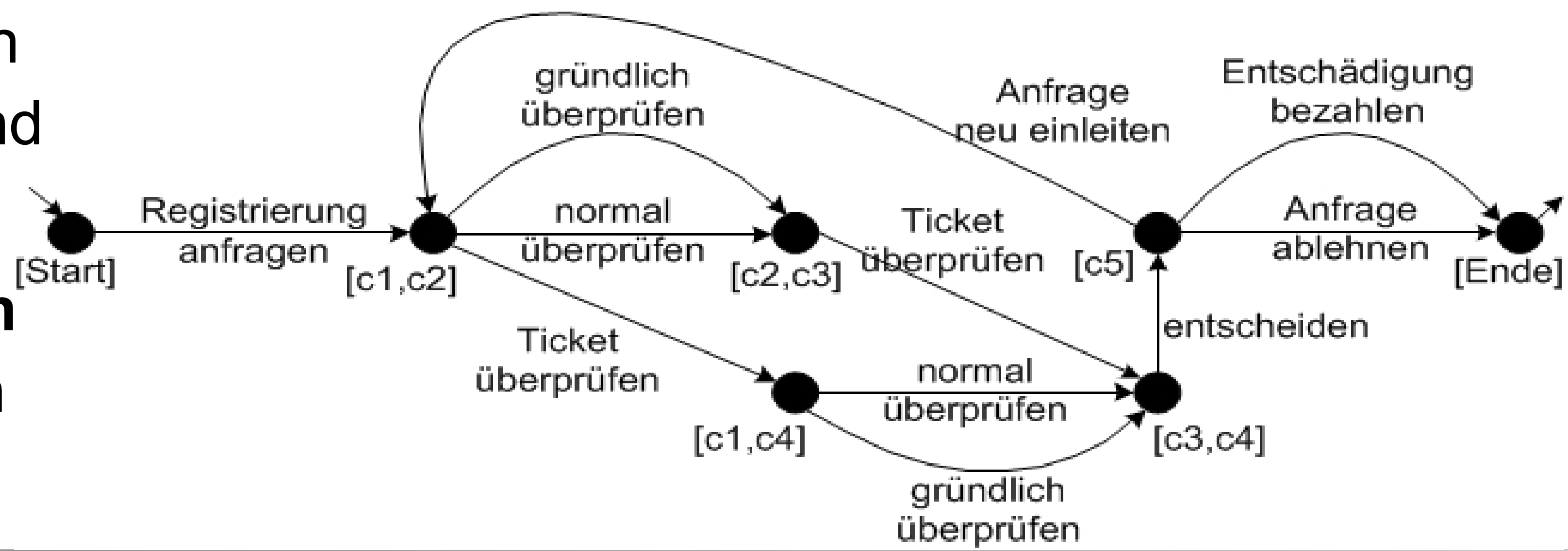
{[t1,t2,t3],[t1,t2,t1]}

Erreichbarkeitsgraph: Weiteres Beispiel

NB: **Bezeichnungen der Zustände** im Erreichbarkeits-graph (z.B. [c1,c2]) spiegeln globalen Zustand wieder

→ **mit Markern belegte Stellen** im Petrinetz.



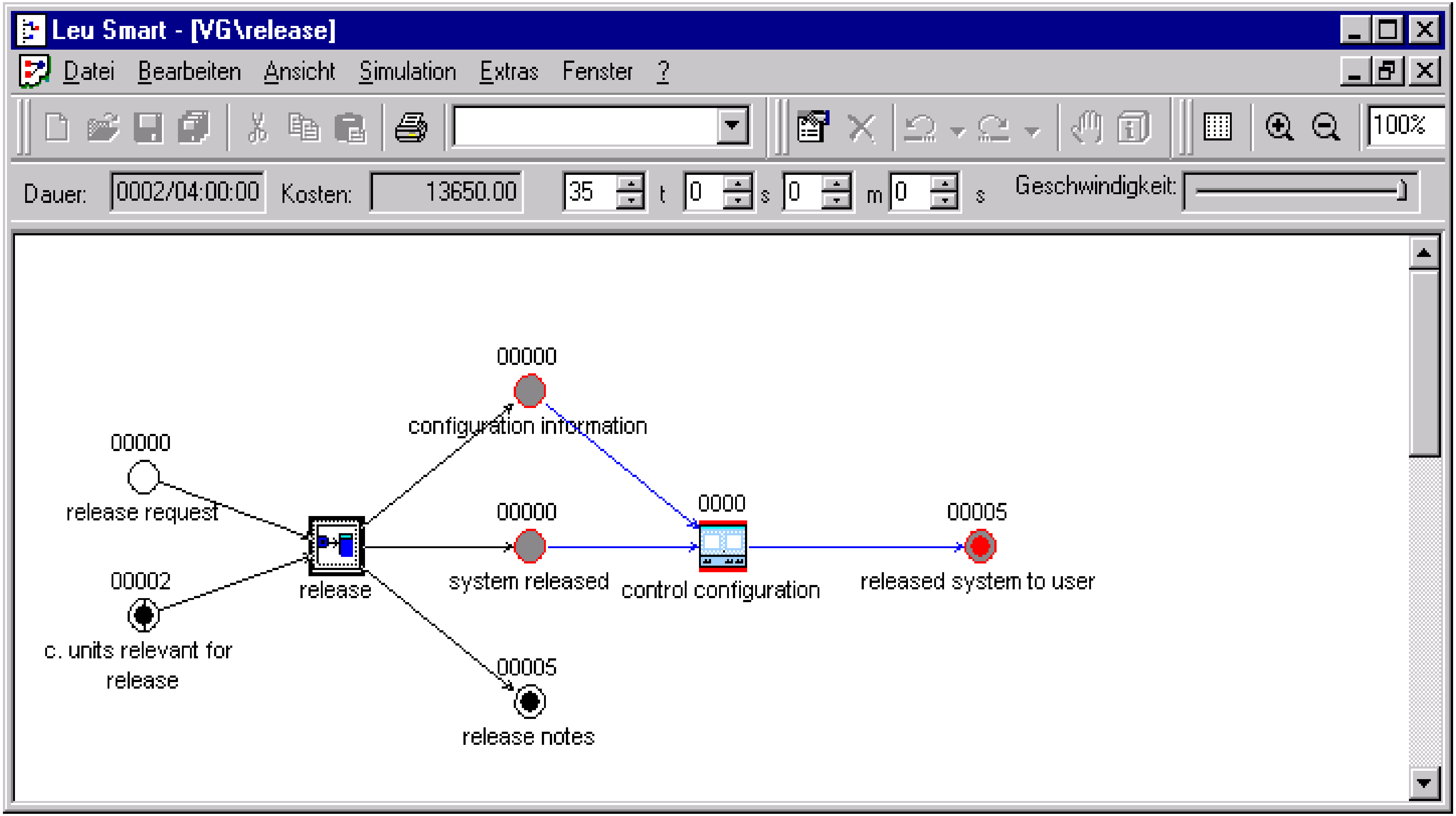


1.4 Petrinetze

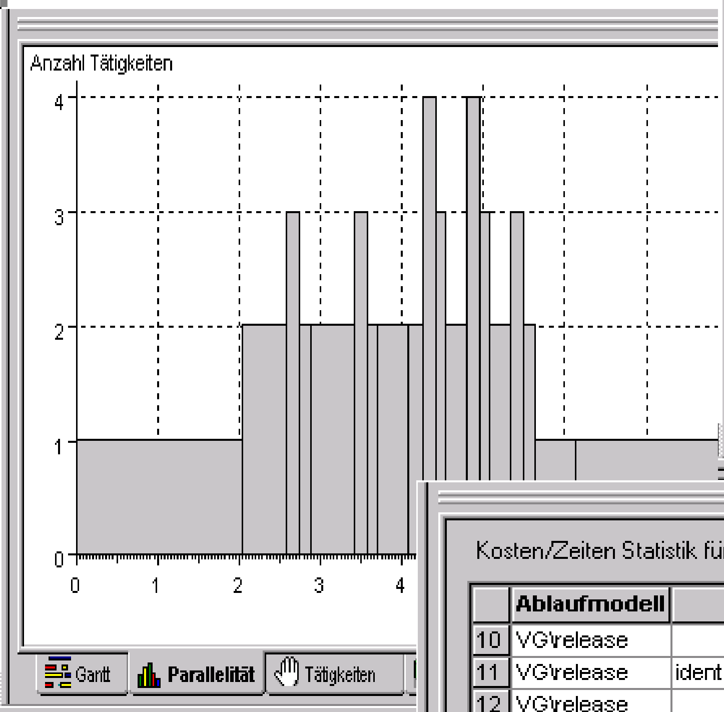
1.4 Petrinetze

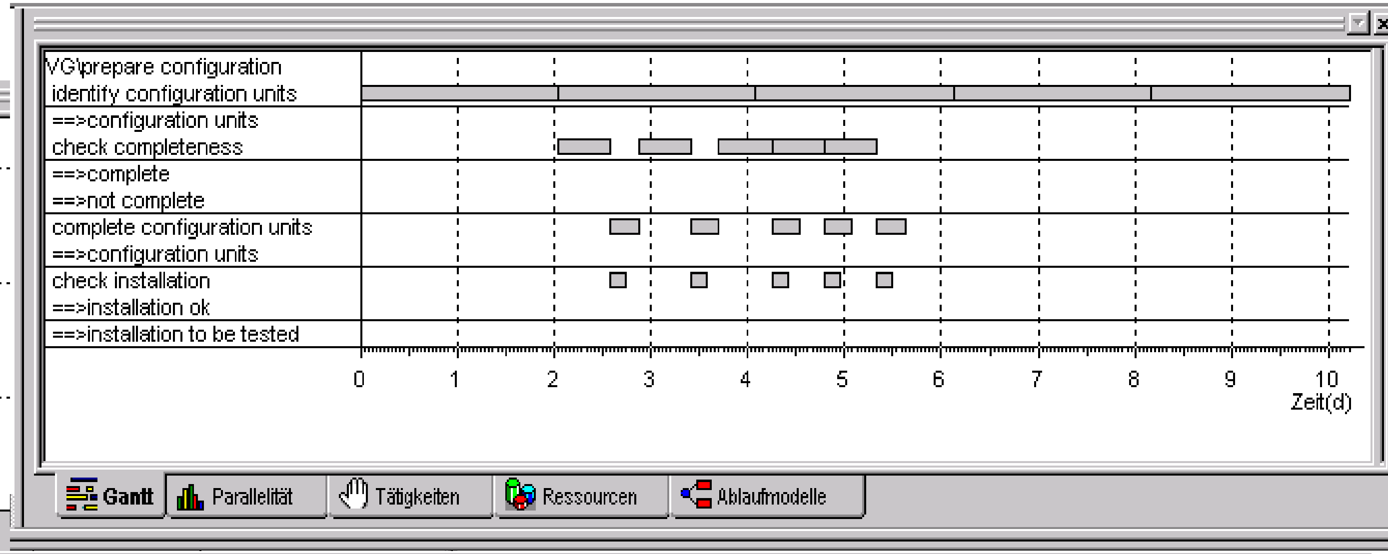
- Petrinetz Syntax
- Ausführung
- **Analyse von Systemen**

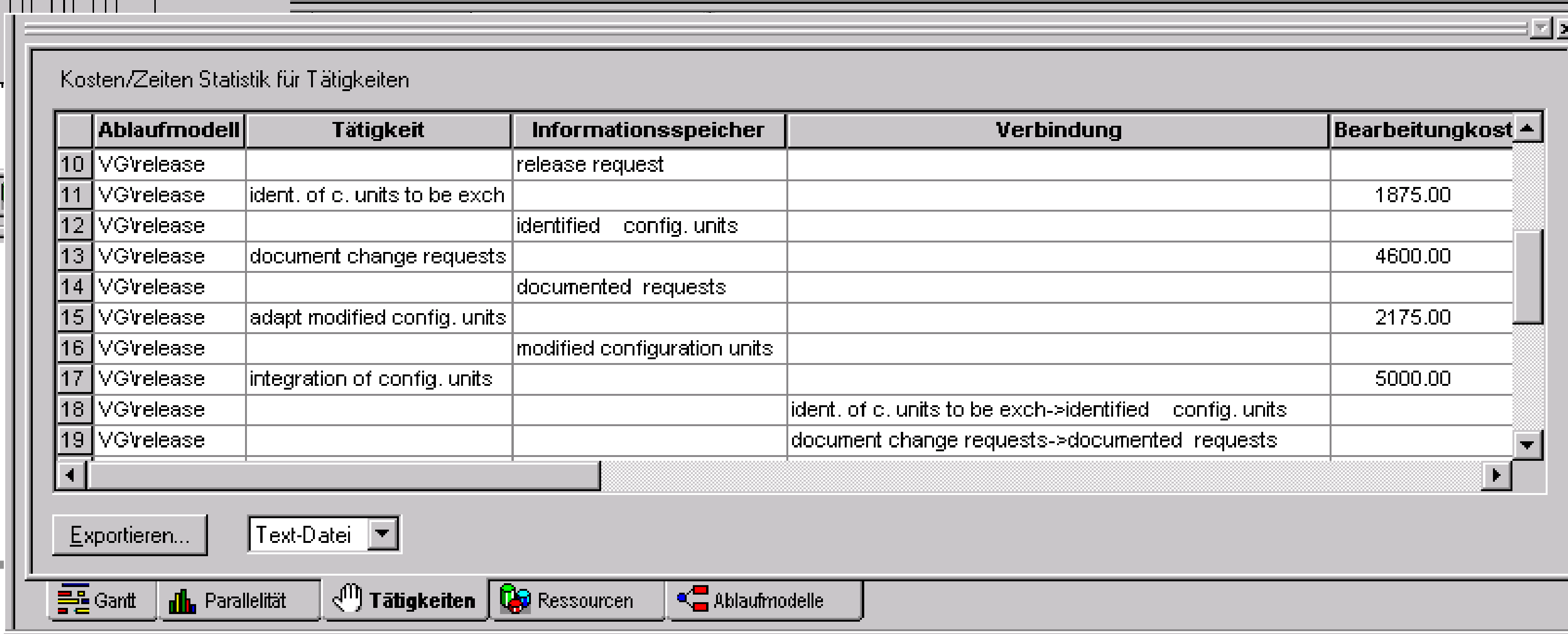
Analyse von Petrinetzen: Animation



Analyse von Systemen: Simulationsanalyse







Kosten/Zeiten Statistik für Tätigkeiten

	Ablaufmodell	Tätigkeit	Informationsspeicher	Verbindung	Bearbeitungkost
10	VG\release		release request		
11	VG\release	ident. of c. units to be exch			1875.00
12	VG\release		identified config. units		
13	VG\release	document change requests			4600.00
14	VG\release		documented requests		
15	VG\release	adapt modified config. units			2175.00
16	VG\release		modified configuration units		
17	VG\release	integration of config. units			5000.00
18	VG\release			ident. of c. units to be exch->identified config. units	
19	VG\release			document change requests->documented requests	

Simulation:

- Kann zeigen, dass bestimmte Situationen auftreten können.
- Kann **nicht** zeigen, dass bestimmte Situationen **nicht** auftreten.
- **Ausschnitt** aus Menge aller möglichen Verhalten.

Analyse von Systemen: Verifikation

Verifikation: Beweis von Eigenschaften:

Statische Eigenschaften: Unabhängig von Markierungen, nur von Netztopologie abhängig.

- z.B. Verklemmungen / Deadlocks

Dynamische Eigenschaften: Abhängig von der Menge erreichbarer Markierungen.

- Standardhilfsmittel: Erreichbarkeitsgraphen (s.o.)

Analyse von Systemen: Sicherheit

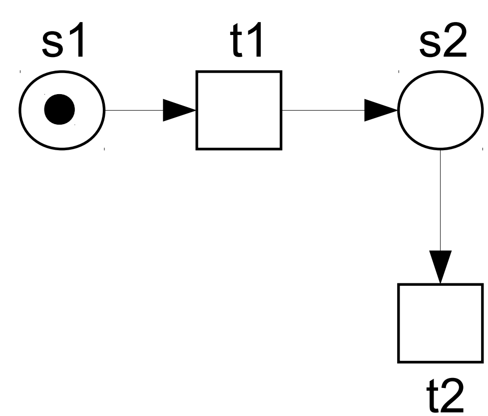
Sei P=(S,T,F,K,W,M_0) Petrinetz.

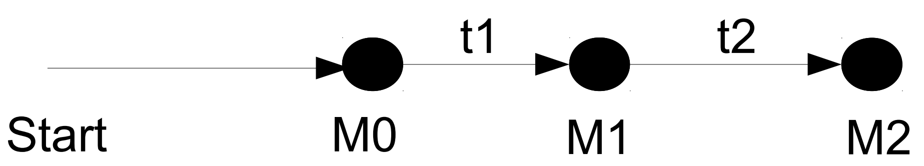
Abbildung $B: S \rightarrow \mathbb{N} \cup \{\infty\}$ ordnet jeder Stelle eine „kritische Markenzahl“ zu.

Petrinetz P heißt:

- **B-sicher** (oder **B-beschränkt**), wenn für alle erreichbaren Markierungen Anzahl der Markierungen pro Stelle durch B begrenzt, d.h.:
  für alle $M \in [M_0>$ und $s \in S$ gilt: $M(s) \leq B(s)$.

  <u>Beispiel:</u> B-sicher für jedes B mit B(s)>=1 für alle s:





Analyse von Systemen: Sicherheit

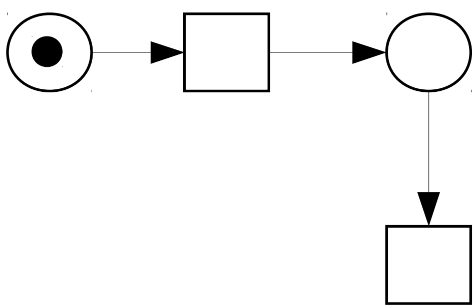
Sei $P=(S,T,F,K,W,M_0)$ Petrinetz.

Abbildung $B:S \rightarrow \mathbb{N} \cup \{\infty\}$ ordnet jeder Stelle eine „kritische Markenzahl" zu.

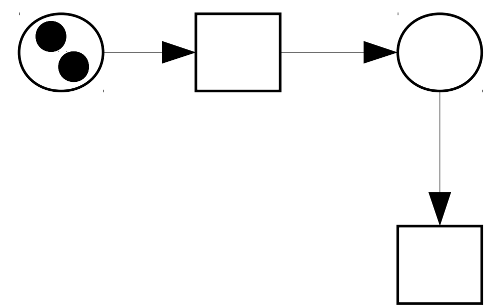
Petrinetz P heißt:

- […]
- **1-sicher**, **2-sicher** usw., wenn B=1, B=2 usw. (konstante Funktionen)

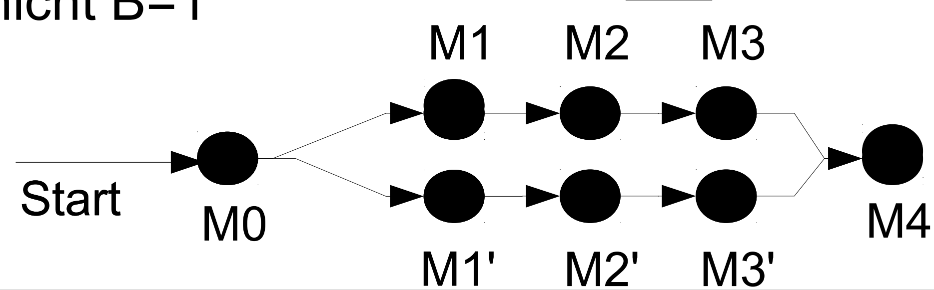
Beispiel:



B-sicher
für B>=1



B-sicher
für B>=2,
aber
nicht B=1



Analyse von Systemen: Sicherheit

Sei $P=(S,T,F,K,W,M_0)$ Petrinetz.

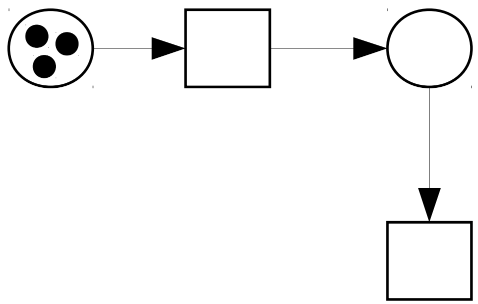
Abbildung $B : S \rightarrow \mathbb{N} \cup \{\infty\}$ ordnet jeder Stelle eine „kritische Markenzahl“ zu.

Petrinetz P heißt:

- **B-sicher** (oder **B-beschränkt**), wenn für alle erreichbaren Markierungen Anzahl der Markierungen pro Stelle durch B begrenzt, d.h.: für alle $M \in [M_0>$ und $s \in S$ gilt: $M(s) \leq B(s)$.
- **1-sicher**, **2-sicher** usw., wenn B=1, B=2 usw.
- **beschränkt**, wenn es natürliche Zahl b gibt, für die P b-sicher.

<u>Beispiel:</u>

Beschränkt,
weil B-sicher
für B>=3



Analyse von Systemen: Sicherheit

Sei $P=(S,T,F,K,W,M_0)$ Petrinetz.

Abbildung $B: S \rightarrow \mathbb{N} \cup \{\infty\}$ ordnet jeder Stelle eine „kritische Markenzahl“ zu.

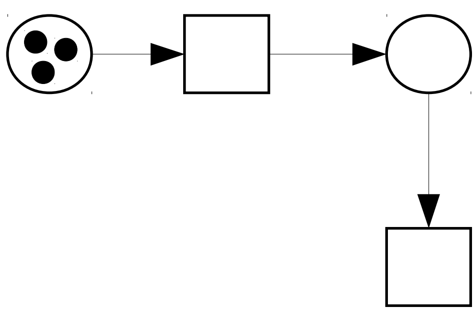
Petrinetz P heißt:

- **B-sicher** (oder **B-beschränkt**), wenn für alle erreichbaren Markierungen Anzahl der Markierungen pro Stelle durch B begrenzt, d.h.: für alle $M \in [M_0>$ und $s \in S$ gilt: $M(s) \leq B(s)$.
- **1-sicher**, **2-sicher** usw., wenn B=1, B=2 usw.
- **beschränkt**, wenn es natürliche Zahl b gibt, für die P b-sicher.

Stelle s heißt **b-sicher**, wenn P B-sicher mit B(s)=b, und B(s')= ∞ für $s' \neq s$.

Beispiel:

Beide Stellen sind b-sicher für b>=3



Analyse von Systemen: Sicherheit

Sei $P=(S,T,F,K,W,M_0)$ Petrinetz.

Abbildung $B: S \to \mathbb{N} \cup \{\infty\}$ ordnet jeder Stelle eine „kritische Markenzahl“ zu.

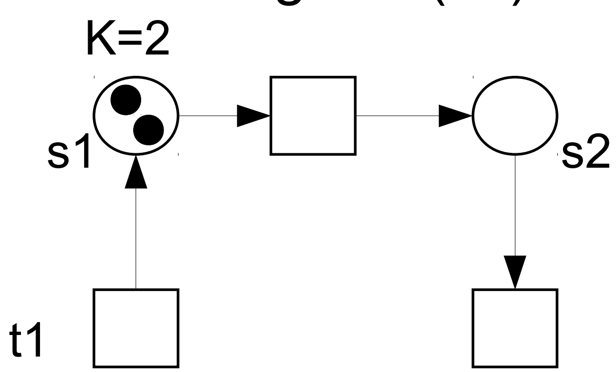
Stelle s heißt **b-sicher**, wenn P B-sicher mit B(s)=b, und B(s‘)= ∞ für s'≠s.

Unterschied zwischen Kapazität und Sicherheit:

- **Kapazität begrenzt** Stellenmarkierung (a priori-Begrenzung).

  Beispiel:

  t1 ist *nicht aktiviert* wegen K(s1)=M(s1)=2.



Analyse von Systemen: Sicherheit

Sei $P=(S,T,F,K,W,M_0)$ Petrinetz.

Abbildung $B: S \rightarrow \mathbb{N} \cup \{\infty\}$ ordnet jeder Stelle eine „kritische Markenzahl“ zu.

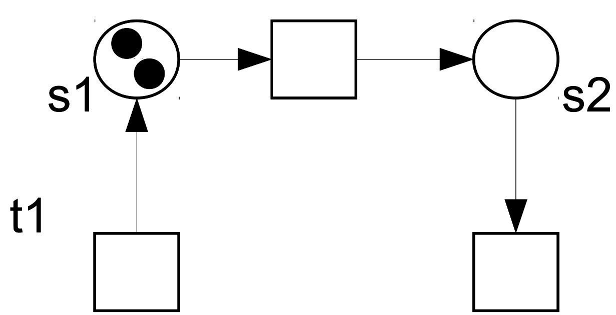
Stelle s heißt **b-sicher**, wenn P B-sicher mit B(s)=b, und B(s‘)= ∞ für s'≠s.

Unterschied zwischen Kapazität und Sicherheit:

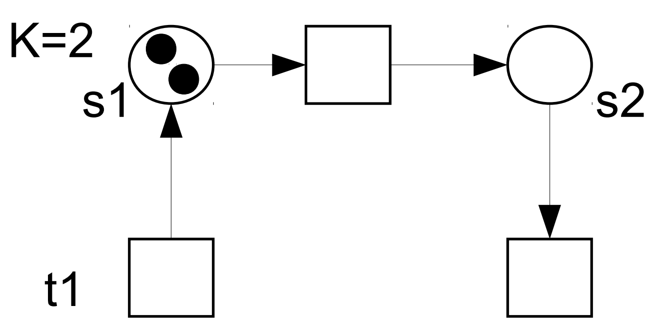
- **Kapazität begrenzt** Stellenmarkierung (a priori-Begrenzung).
- **Sicherheit beobachtet** Stellenmarkierung (a posteriori-Begrenzung).

  Beispiel:

s1 ist *nicht 2-sicher*.

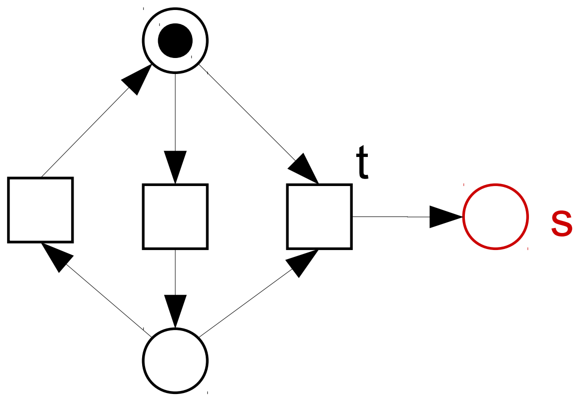


s1 *ist 2-sicher*.



Sicherheit: Beispiele

- *Beispiel:* Verkehrsplaner modelliert **Ampelsystem**.
  - An bestimmter Stelle s darf sich nur ein Auto aufhalten.
  - K(s)=1: Keine Aussage über Korrektheit der Modellierung.
  - K(s)=∞: In Analyse prüfbar, ob B-sicher für B(s)=1.
- *Beispiel:* Transition t soll nie schalten dürfen.
  - Sicherheitseigenschaft: Beobachtungsstelle s und Bedingung B(s)=0 hinzufügen.

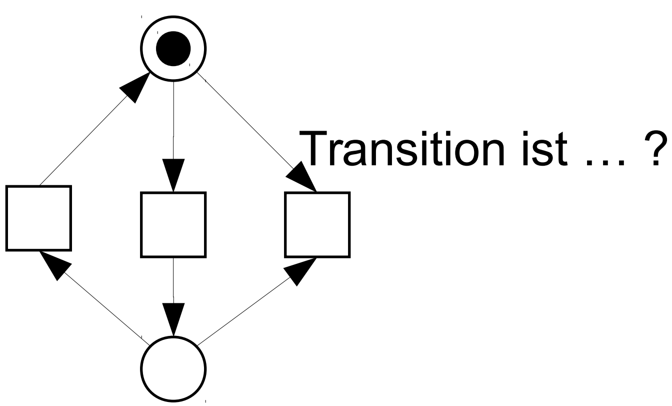


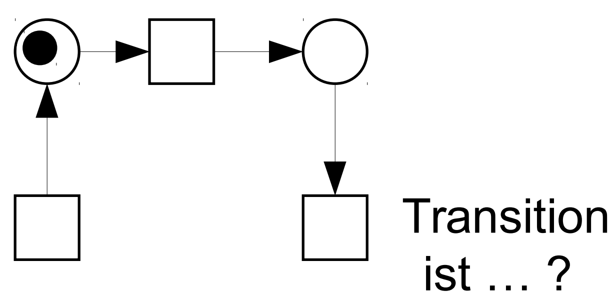
Analyse von Systemen: Lebendigkeit von Transitionen

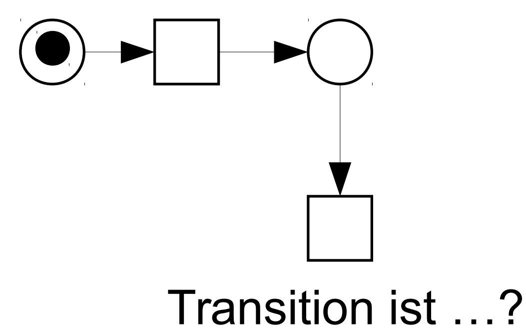
Transition t eines Petrinetz $P=(S,T,F,K,W,M_0)$ heißt:

- **aktivierbar**: In mindestens einer erreichbaren Markierung aktiviert: existiert $M_1 \in [M_0\rangle$ mit: $M_1[t\rangle$
- **lebendig**: In allen erreichbaren Markierung **aktivierbar**: für alle $M_1 \in [M_0\rangle$ gilt: existiert $M_2 \in [M_1\rangle$ mit: $M_2[t\rangle$
- **tot:** In keiner erreichbaren Markierung aktiviert: für alle $M \in [M_0\rangle$ gilt: $\neg M[t\rangle$

Tot ist nicht logische Negation von lebendig sondern von aktivierbar !





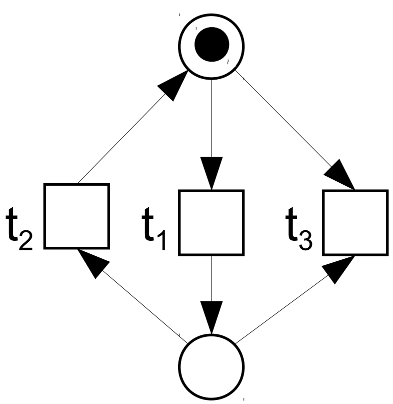


Analyse von Systemen: Lebendigkeit von Transitionen

Transition t eines Petrinetz $P=(S,T,F,K,W,M_0)$ heißt:

- **aktivierbar**: In mindestens einer erreichbaren Markierung aktiviert: existiert $M_1 \in [M_0>$ mit: $M_1[t>$
- **lebendig**: In allen erreichbaren Markierung **aktivierbar**: für alle $M_1 \in [M_0>$ gilt: existiert $M_2 \in [M_1>$ mit: $M_2[t>$
- **tot:** In keiner erreichbaren Markierung aktiviert: für alle $M \in [M_0>$ gilt: $\neg M[t>$

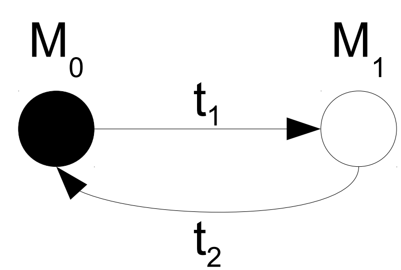
Tot ist nicht logische Negation von lebendig sondern von aktivierbar !



Tote Transition

Gibt in jedem Zustand nur eine Marke, Transition t_3 braucht aber zwei !

Erreichbarkeitsgraph:

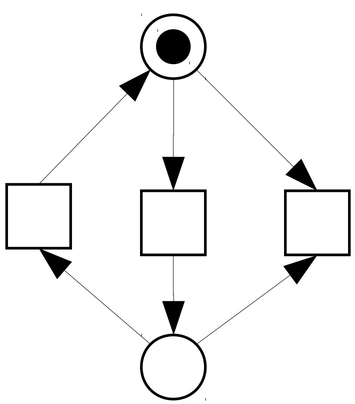


Analyse von Systemen: Lebendigkeit von Transitionen

Transition t eines Petrinetz $P=(S,T,F,K,W,M_0)$ heißt:

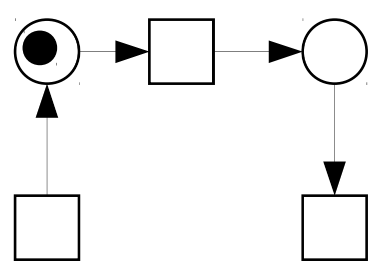
- **aktivierbar**: In mindestens einer erreichbaren Markierung aktiviert:
  existiert $M_1 \in [M_0\rangle$ mit: $M_1[t\rangle$
- **lebendig**: In allen erreichbaren Markierung **aktivierbar**:
  für alle $M_1 \in [M_0\rangle$ gilt: existiert $M_2 \in [M_1\rangle$ mit: $M_2[t\rangle$
- **tot:** In keiner erreichbaren Markierung aktiviert:
  für alle $M \in [M_0\rangle$ gilt: $\neg M[t\rangle$

Tot ist nicht logische Negation von lebendig sondern von aktivierbar !

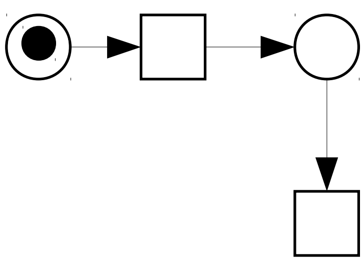


Tote Transition

Gibt in jedem Zustand nur eine Marke, Transition t_3 braucht aber zwei !



Transition ist … ?



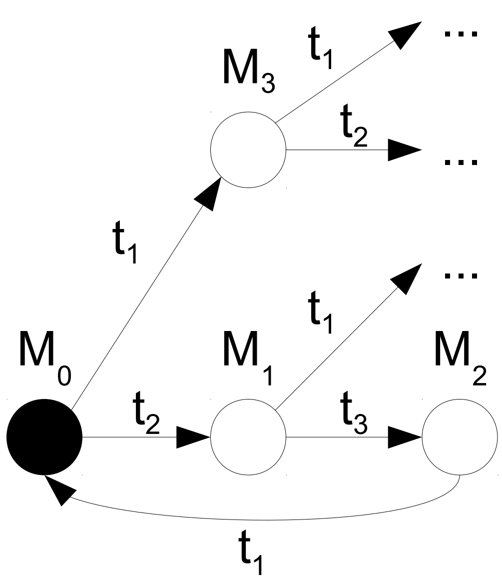
Transition ist …?

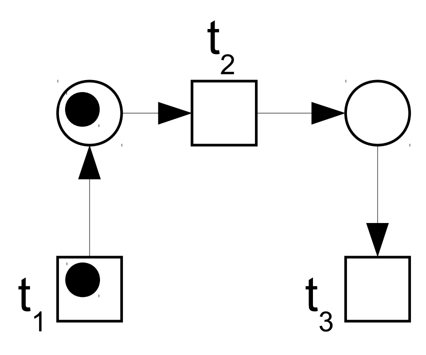
Analyse von Systemen: Lebendigkeit von Transitionen

Transition t eines Petrinetz $P=(S,T,F,K,W,M_0)$ heißt:

- **aktivierbar**: In mindestens einer erreichbaren Markierung aktiviert: existiert $M_1 \in [M_0>$ mit: $M_1[t>$
- **lebendig**: In allen erreichbaren Markierung **aktivierbar**: für alle $M_1 \in [M_0>$ gilt: existiert $M_2 \in [M_1>$ mit: $M_2[t>$

Erreichbarkeitsgraph:



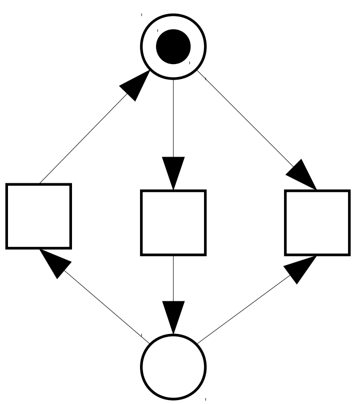


Analyse von Systemen: Lebendigkeit von Transitionen

Transition t eines Petrinetz $P=(S,T,F,K,W,M_0)$ heißt:

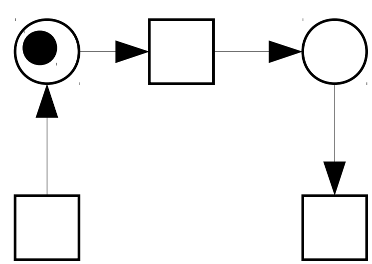
- **aktivierbar**: In mindestens einer erreichbaren Markierung aktiviert:
  existiert $M_1 \in [M_0\rangle$ mit: $M_1[t\rangle$
- **lebendig**: In allen erreichbaren Markierung **aktivierbar**:
  für alle $M_1 \in [M_0\rangle$ gilt: existiert $M_2 \in [M_1\rangle$ mit: $M_2[t\rangle$
- **tot:** In keiner erreichbaren Markierung aktiviert:
  für alle $M \in [M_0\rangle$ gilt: $\neg M[t\rangle$

Tot ist nicht logische Negation von lebendig sondern von aktivierbar !

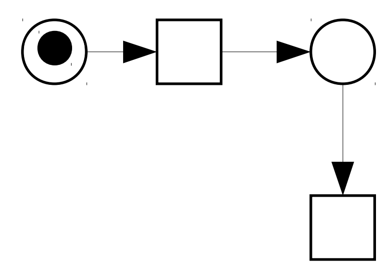


Tote Transition

Gibt in jedem Zustand nur eine Marke, Transition t_3 braucht aber zwei !



Lebendige Transition



Transition ist …?

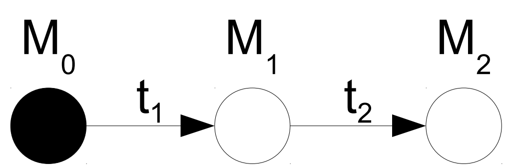
Analyse von Systemen: Lebendigkeit von Transitionen

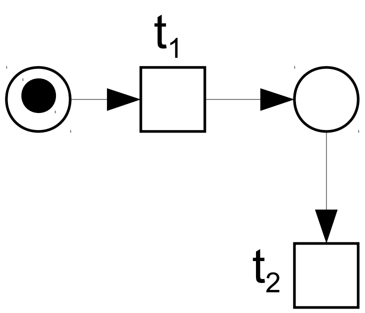
Transition t eines Petrinetz $P=(S,T,F,K,W,M_0)$ heißt:

- **aktivierbar**: In mindestens einer erreichbaren Markierung aktiviert: existiert $M_1 \in [M_0>$ mit: $M_1[t>$
- **lebendig**: In allen erreichbaren Markierung **aktivierbar**: für alle $M_1 \in [M_0>$ gilt: existiert $M_2 \in [M_1>$ mit: $M_2[t>$
- **tot:** In keiner erreichbaren Markierung aktiviert: für alle $M \in [M_0>$ gilt: $\neg M[t>$

Tot ist nicht logische Negation von lebendig sondern von aktivierbar !

Erreichbarkeitsgraph:





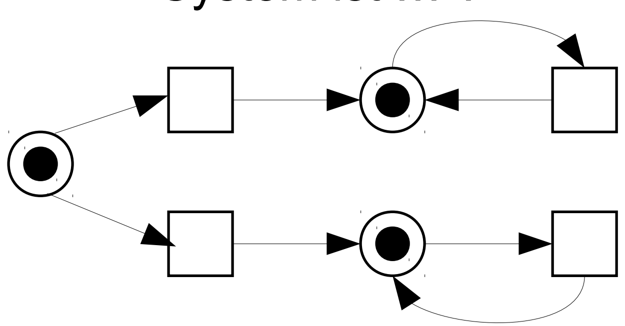
Aktivierbare Transition

Analyse von Systemen: Lebendigkeit von Petrinetzen

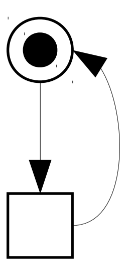
Petrinetz $P=(S,T,F,K,W,M_0)$ heißt:

- **lebendig**: In jeder erreichbaren Markierung ist jede Transition **aktivierbar**:
  $\forall M_1 \in [M_0\rangle$ und $t \in T$ gilt: $\exists M_2 \in [M_1\rangle$ mit: $M_2[t\rangle$
- **deadlockfrei**: In jeder erreichbaren Markierung ist mindestens eine Transition aktiviert:
  $\forall M_1 \in [M_0\rangle$ gilt: $\exists\, t \in T$ mit: $M_1[t\rangle$
- **tot:** Keine Transition aktiviert:
  $\forall\, t \in T: \neg M_0\, [t\rangle$

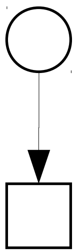
System ist … ?



System ist … ?



System ist … ?

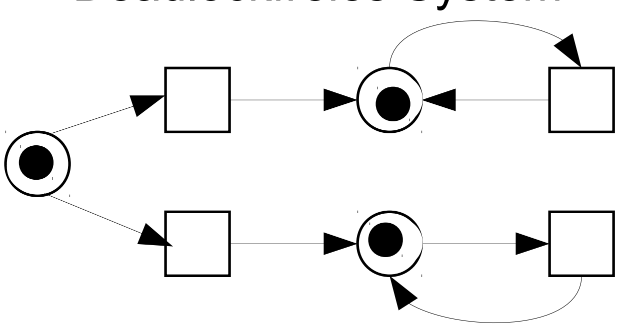


Analyse von Systemen: Lebendigkeit von Petrinetzen

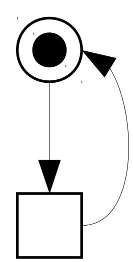
Petrinetz $P=(S,T,F,K,W,M_0)$ heißt:

- **lebendig**: In jeder erreichbaren Markierung ist jede Transition **aktivierbar**:
  $\forall M_1 \in [M_0>$ und $t \in T$ gilt: $\exists M_2 \in [M_1>$ mit: $M_2[t>$
- **deadlockfrei**: In jeder erreichbaren Markierung ist mindestens eine Transition aktiviert:
  $\forall M_1 \in [M_0>$ gilt: $\exists\, t \in T$ mit: $M_1[t>$
- **tot:** Keine Transition aktiviert:
  $\forall t \in T: \neg M_0\,[t>$

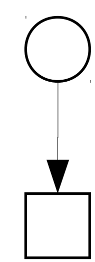
Deadlockfreies System



System ist … ?



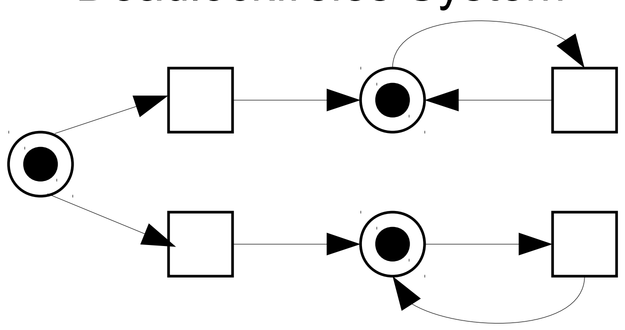
System ist … ?



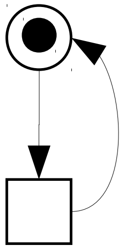
Petrinetz $P=(S,T,F,K,W,M_0)$ heißt:

- **lebendig**: In jeder erreichbaren Markierung ist jede Transition **aktivierbar**:
  $\forall M_1 \in [M_0\rangle$ und $t \in T$ gilt: $\exists M_2 \in [M_1\rangle$ mit: $M_2[t\rangle$
- **deadlockfrei**: In jeder erreichbaren Markierung ist mindestens eine Transition aktiviert:
  $\forall M_1 \in [M_0\rangle$ gilt: $\exists\, t \in T$ mit: $M_1[t\rangle$
- **tot:** Keine Transition aktiviert:
  $\forall t \in T: \neg M_0\, [t\rangle$

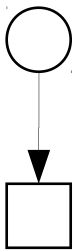
Deadlockfreies System



Lebendiges System



System ist … ?

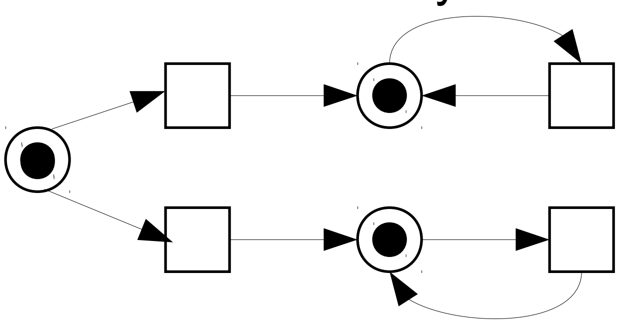


Analyse von Systemen: Lebendigkeit von Petrinetzen

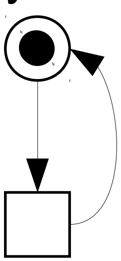
Petrinetz $P=(S,T,F,K,W,M_0)$ heißt:

- **lebendig**: In jeder erreichbaren Markierung ist jede Transition **aktivierbar**:
  $\forall M_1 \in [M_0>$ und $t \in T$ gilt: $\exists M_2 \in [M_1>$ mit: $M_2[t>$
- **deadlockfrei**: In jeder erreichbaren Markierung ist mindestens eine Transition aktiviert:
  $\forall M_1 \in [M_0>$ gilt: $\exists\, t \in T$ mit: $M_1[t>$
- **tot:** Keine Transition aktiviert:
  $\forall t \in T: \neg M_0\, [t>$

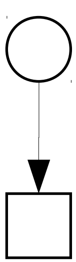
Deadlockfreies System



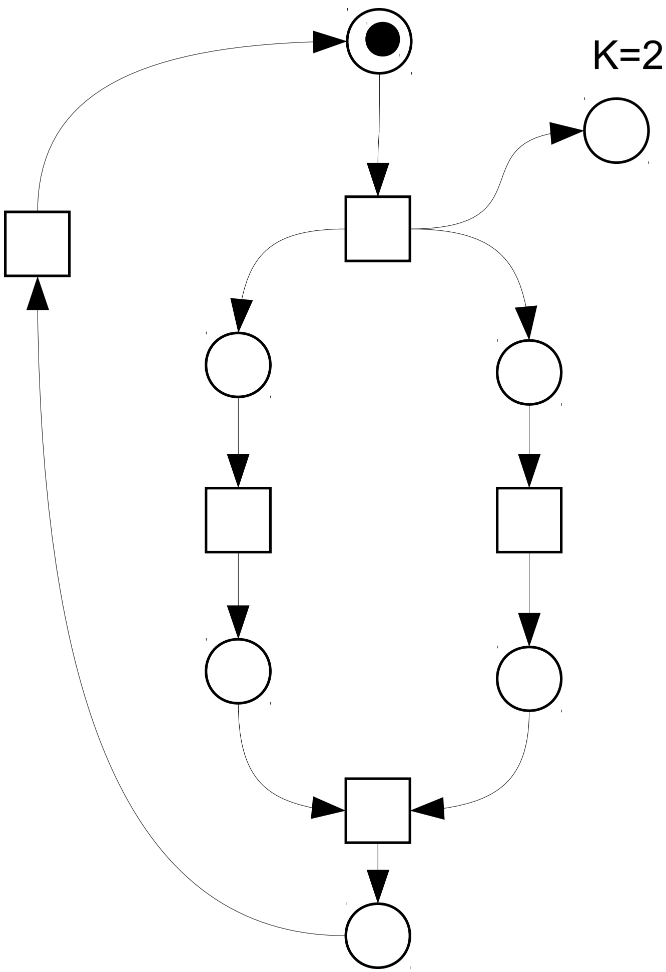
Lebendiges System



Totes System

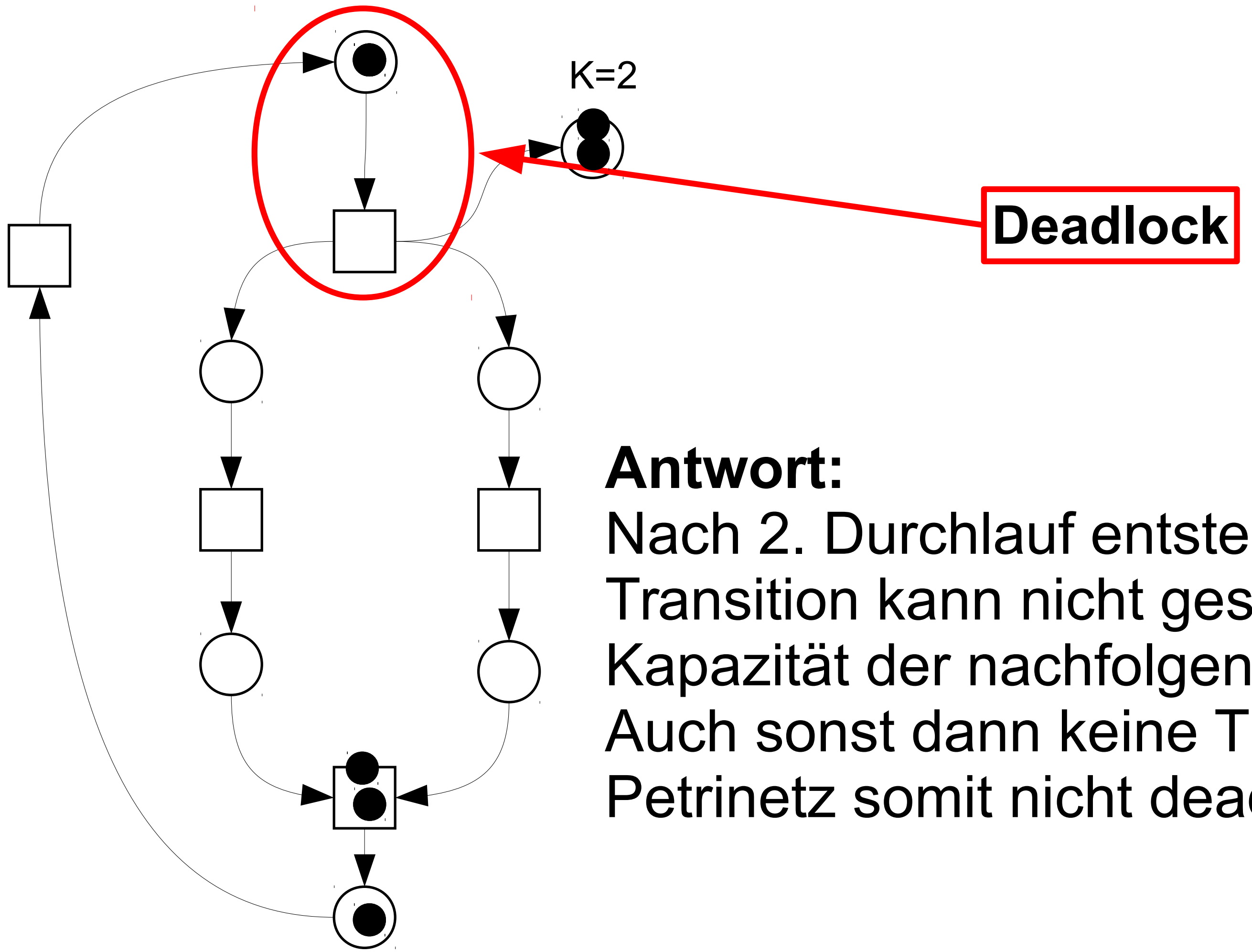


Deadlockfrei / lebendig / tot ?



Analyse von Systemen: Beispiel

Deadlockfrei / lebendig / tot ?



Antwort:
Nach 2. Durchlauf entsteht Deadlock:
Transition kann nicht geschaltet werden, da Kapazität der nachfolgenden Stelle erschöpft.
Auch sonst dann keine Transition aktiviert.
Petrinetz somit nicht deadlockfrei.

Lebendigkeit von Petrinetzen: Anmerkungen

tot: keine Transition aktivierbar, d.h. alle Transitionen tot.

(**Achtung**: Nicht **aktivierte** Transition kann trotzdem **aktivierbar** sein, aber wenn keine Transition im Petrinetz **aktiviert** ist, ist auch keine **aktivierbar** !)

Lebendigkeit von Petrinetzen: Anmerkungen

tot: keine Transition aktivierbar, d.h. alle Transitionen tot.

(**Achtung**: Nicht **aktivierte** Transition kann trotzdem **aktivierbar** sein, aber wenn keine Transition im Petrinetz **aktiviert** ist, ist auch keine **aktivierbar** !)

D.h. im Erreichbarkeitsgraph geht von der initialen Markierung keine Kante aus.

- Bedeutet häufig einen Deadlock

Lebendigkeit von Petrinetzen: Anmerkungen

tot: keine Transition aktivierbar, d.h. alle Transitionen tot.

(**Achtung**: Nicht **aktivierte** Transition kann trotzdem **aktivierbar** sein, aber wenn keine Transition im Petrinetz **aktiviert** ist, ist auch keine **aktivierbar** !)

D.h. im Erreichbarkeitsgraph geht von der initialen Markierung keine Kante aus.

- Bedeutet häufig einen Deadlock

deadlockfrei: keine erreichbare Markierung tot.

- Berücksichtigung partieller Ausfälle („graceful degradation“).

Lebendigkeit von Petrinetzen: Anmerkungen

tot: keine Transition aktivierbar, d.h. alle Transitionen tot.

(**Achtung**: Nicht **aktivierte** Transition kann trotzdem **aktivierbar** sein, aber wenn keine Transition im Petrinetz **aktiviert** ist, ist auch keine **aktivierbar** !)

D.h. im Erreichbarkeitsgraph geht von der initialen Markierung keine Kante aus.

- Bedeutet häufig einen Deadlock

deadlockfrei: keine erreichbare Markierung tot.

- Berücksichtigung partieller Ausfälle („graceful degradation“).

lebendig: alle Transitionen lebendig.

D.h.: in Erreichbarkeitsgraph existiert von jedem Knoten ausgehend für jedes t aus T einen Pfad, in dem t als Label vorkommt.

Lebendigkeit von Petrinetzen: Anmerkungen

tot: keine Transition aktivierbar, d.h. alle Transitionen tot.

(**Achtung**: Nicht **aktivierte** Transition kann trotzdem **aktivierbar** sein, aber wenn keine Transition im Petrinetz **aktiviert** ist, ist auch keine **aktivierbar** !)

D.h. im Erreichbarkeitsgraph geht von der initialen Markierung keine Kante aus.

- Bedeutet häufig einen Deadlock

deadlockfrei: keine erreichbare Markierung tot.

- Berücksichtigung partieller Ausfälle („graceful degradation“).

lebendig: alle Transitionen lebendig.

D.h.: in Erreichbarkeitsgraph existiert von jedem Knoten ausgehend für jedes t aus T einen Pfad, in dem t als Label vorkommt.

Lebendig ist nicht logische Negation von tot (sondern stärker).

Gibt viele Varianten dieser Definitionen.

Bewertung: Petrinetze

- \+ **Einfache** und **wenige** Sprachelemente.
- \+ **Graphisch** gut darstellbar.
- \+ Marken: übersichtliche **Visualisierung** des **Systemzustands**.
- \+ Syntax und Semantik **formal** definiert.
- \+ **Werkzeuge** zur Erstellung, Analyse, Simulation, Code-Generierung vorhanden (z.B. Process Mining).
- \+ Gut geeignet für **kooperierende** Prozesse.
- \- Zunächst keine **Datenmodellierung** (kann aber dahin erweitern).

Zusammenfassung

In diesem Abschnitt:

Petrinetze als Grundlage für Geschäftsprozessmodellierung und für Process-Mining.

- Petrinetz-Syntax
- Ausführung
- Analyse von Systemen
- Workflow-Netze

Im nächsten Abschnitt:

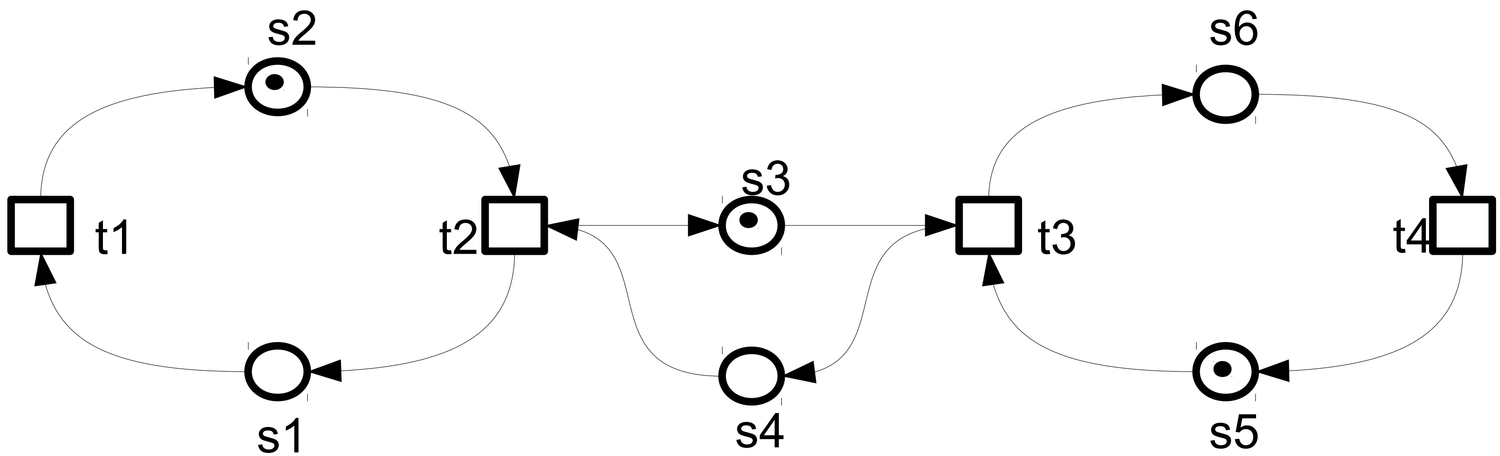
- Eclipse Modeling Framework (EMF): Standard zur Modellierung.

Anhang (zum selbständigen Nacharbeiten)

Petrinetz Ablauf: Erreichbarkeitstabelle

Nr.	s1	s2	s3	s4	s5	s6	Schaltungen
M0	1	0	0	1	1	0	t1 --> M1
M1	0	1	0	1	1	0	t2 --> M2
M2	1	0	1	0	1	0	t3 --> M3 t1 --> M3'
M3	1	0	0	1	0	1	t1 --> M4 t4 --> M0
M3'	0	1	1	0	1	0	t2 --> M5 t3 --> M6

M3':

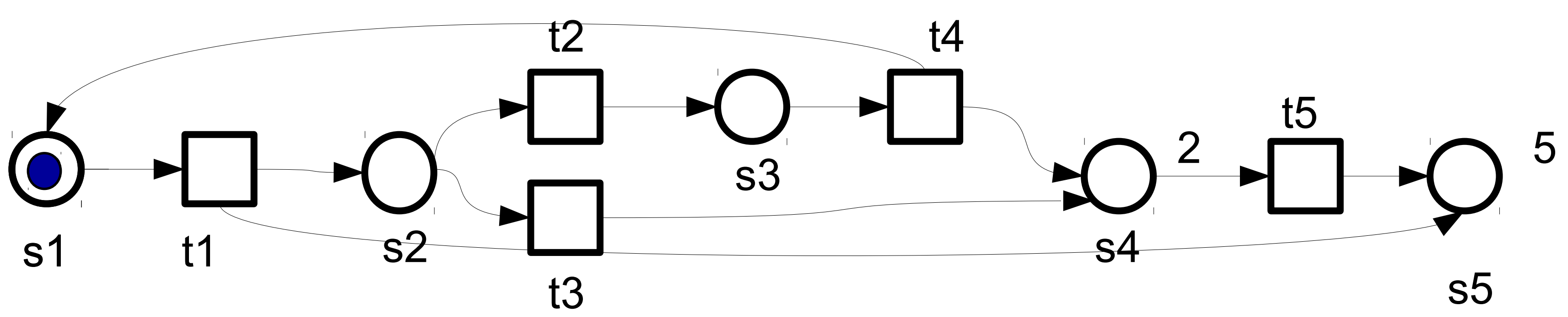


Nichtdeterministische Auswahl von Transition: Beispiel

Nr.	s1	s2	s3	s4	s5	Schaltungen
M0	1	0	0	0	0	
M1						
M2						
M3						
M4						

M0:

Nächster Zustand ?

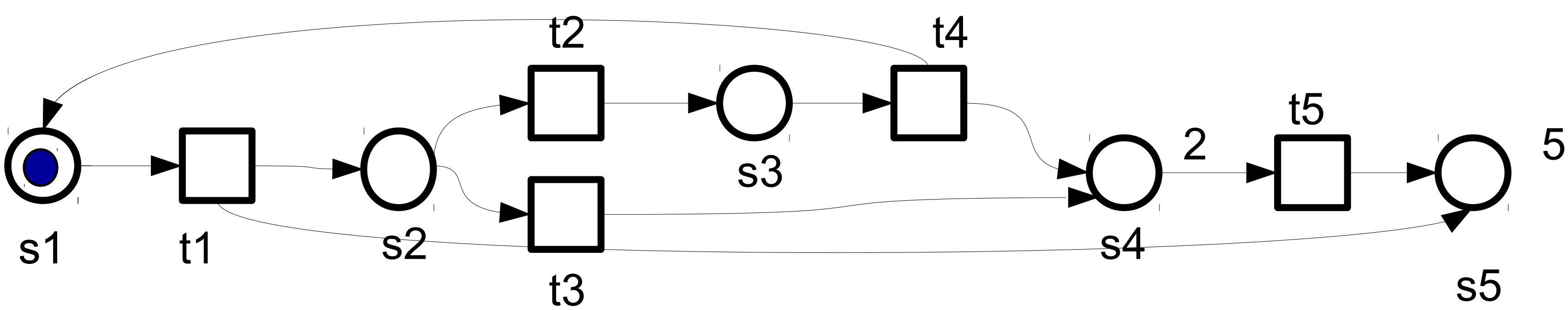


Nichtdeterministische Auswahl von Transition: Beispiel

Nr.	s1	s2	s3	s4	s5	Schaltungen
M0	1	0	0	0	0	t1->M1
M1	0	1	0	0	1	
M2						
M3						
M4						

M1:

Nächster Zustand ?

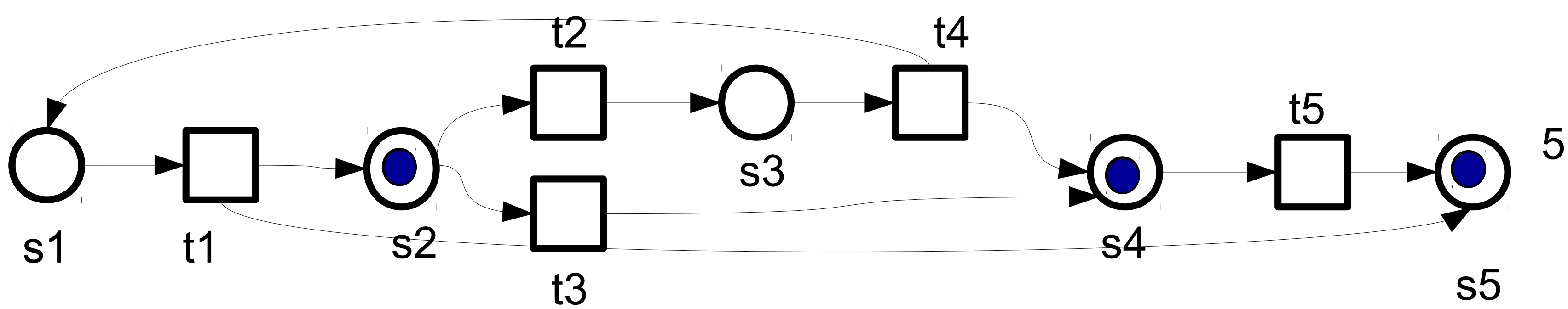


Nichtdeterministische Auswahl von Transition: Beispiel

Nr.	s1	s2	s3	s4	s5	Schaltungen
M0	1	0	0	0	0	t1->M1
M1	0	1	0	0	1	t3->M2
M2	0	0	0	1	1	
M3						
M4						

M2:

Nächster Zustand ?

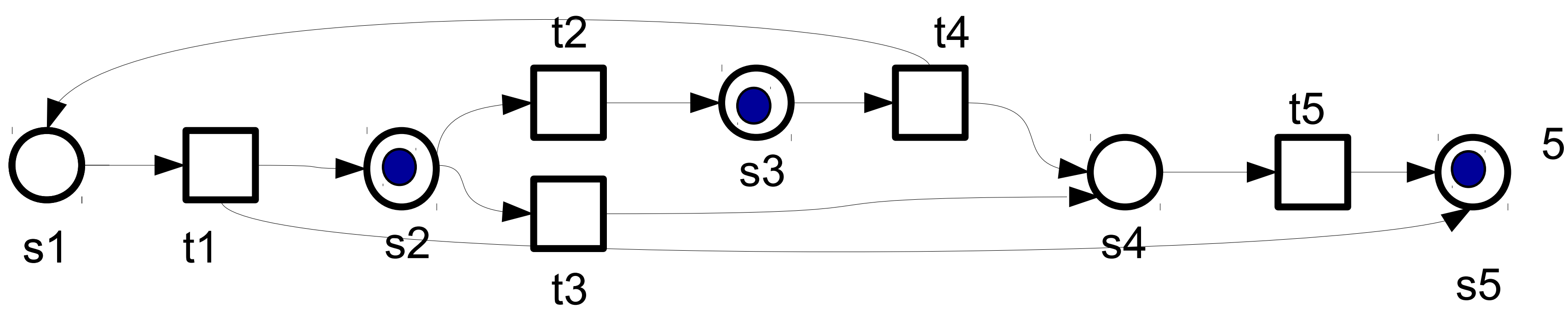


Nichtdeterministische Auswahl von Transition: Beispiel

Nr.	s1	s2	s3	s4	s5	Schaltungen
M0	1	0	0	0	0	t1->M1
M1	0	1	0	0	1	t3->M2
M2	0	0	0	1	1	t2->M3
M3	0	0	1	0	1	
M4						

M3:

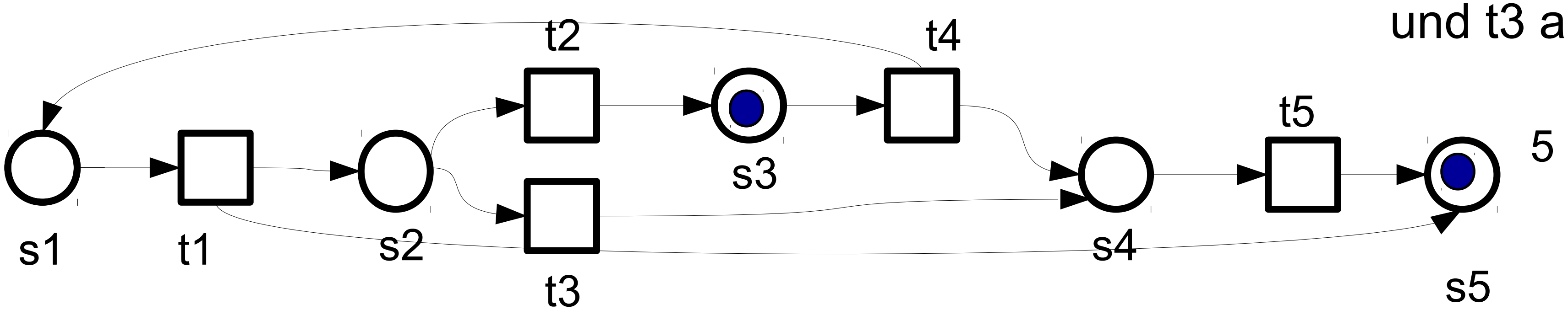
Nächster Zustand ?



Nichtdeterministische Auswahl von Transition: Beispiel

Nr.	s1	s2	s3	s4	s5	Schaltungen
M0	**1**	**0**	**0**	**0**	**0**	**t1->M1**
M1	**0**	**1**	**0**	**0**	**0**	**t2->M2**
M2	**0**	**0**	**1**	**0**	**1**	**t3->M3**
M3	**0**	**0**	**0**	**1**	**1**	**t4->M4**
M4	**1**	**0**	**0**	**1**	**1**	t5->M5 t1->M1

M4:



Was fällt bzgl. der Ausführung von t2 und t3 auf ?

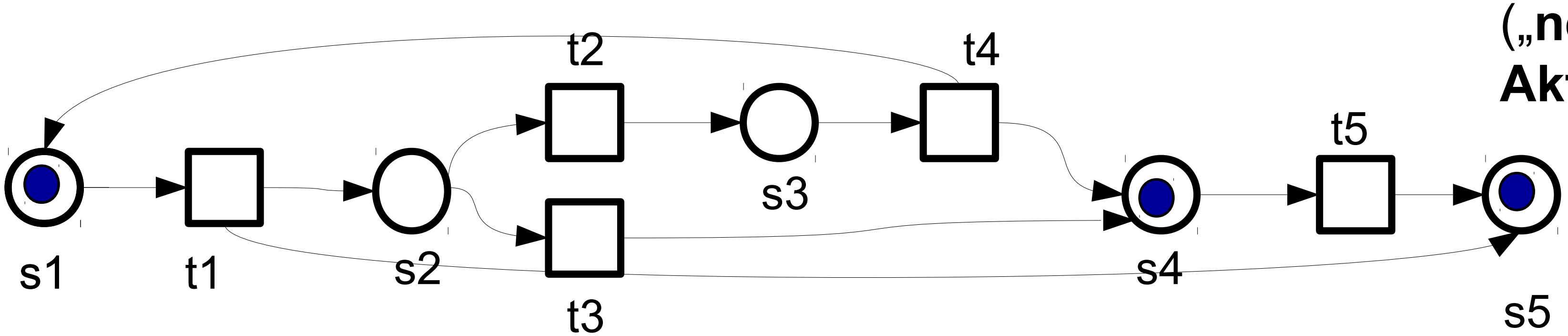
Nichtdeterministische Auswahl von Transition: Beispiel

Nr.	s1	s2	s3	s4	s5	Schaltungen
M0	1	0	0	0	0	t1->M1
M1	0	1	0	0	1	t2->M2
M2	0	0	1	0	1	t3->M3
M3	0	0	0	1	1	t4->M4
M4	1	0	0	1	1	t5->M5 t1->M1

t2 und t4 können in beliebiger Reihenfolge ausgeführt werden und resultieren in denselben Zustand !

(„**nebenläufige Aktivierung**“)

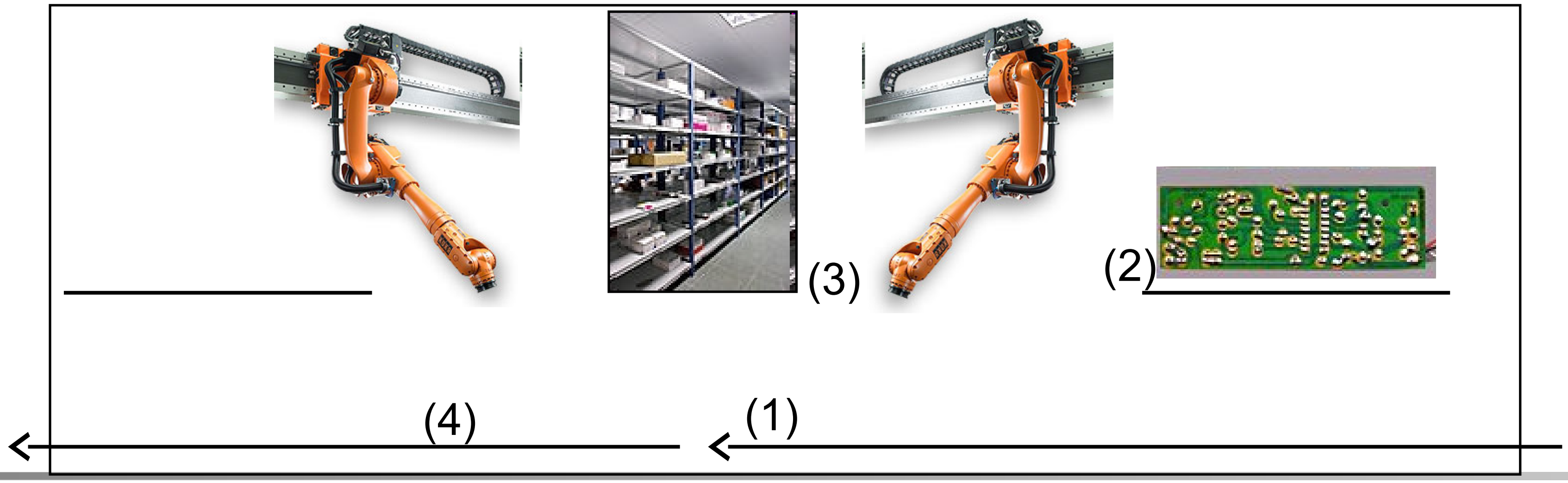
M4:



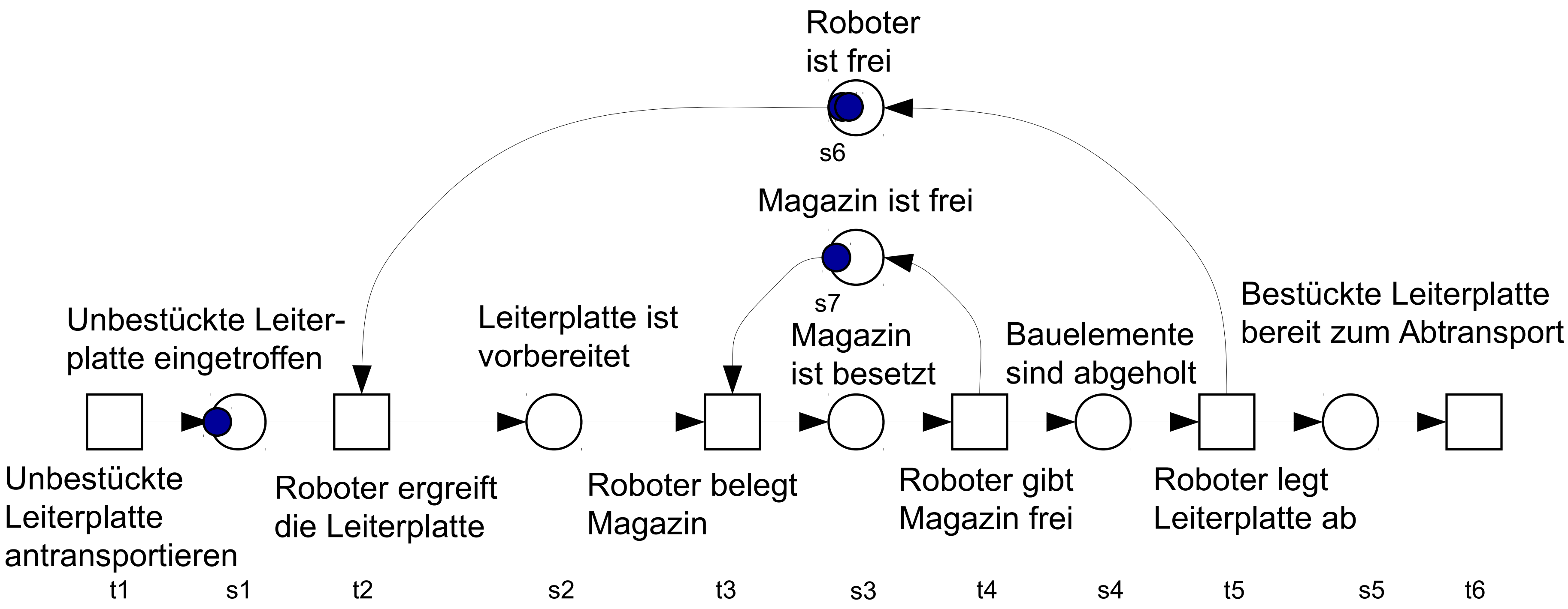
Beispiel: Bestückungsroboter

Zwei Roboter bestücken **Leiterplatten** mit elektronischen **Bauelementen**:

- Leiterplatten auf Fließband antransportiert (1).
- Freier Roboter nimmt Leiterplatte vom Fließband (2).
- Beide Roboter frei: nichtdeterministisch entschieden, wer Leiterplatte nimmt.
- Jeweils ein Roboter darf auf Bauelemente-Magazin zugreifen (3), um Leiterplatte mit Bauelementen zu bestücken.
- Jeweils eine Leiterplatte zu einem Zeitpunkt abtransportierbar (4).

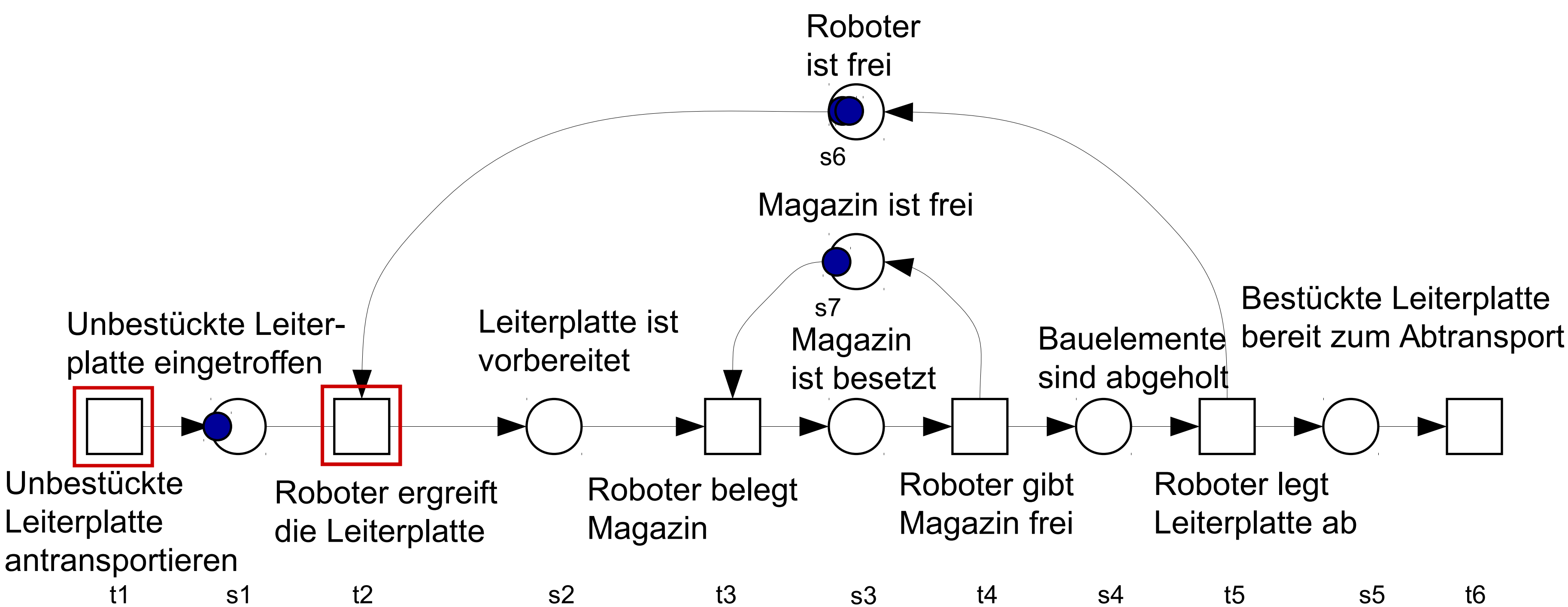


Beispiel: Bestückungsroboter (M0)



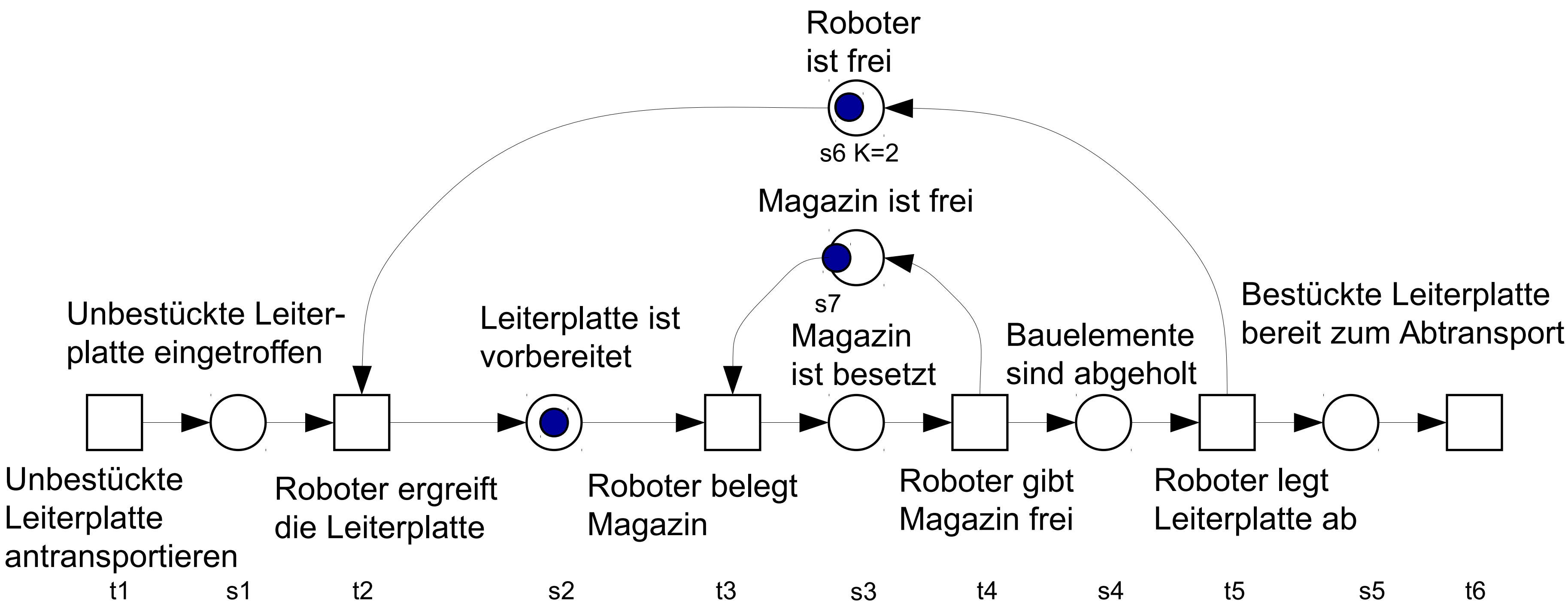
Welche Transition(en) aktiviert ?

Beispiel: Bestückungsroboter (M0)



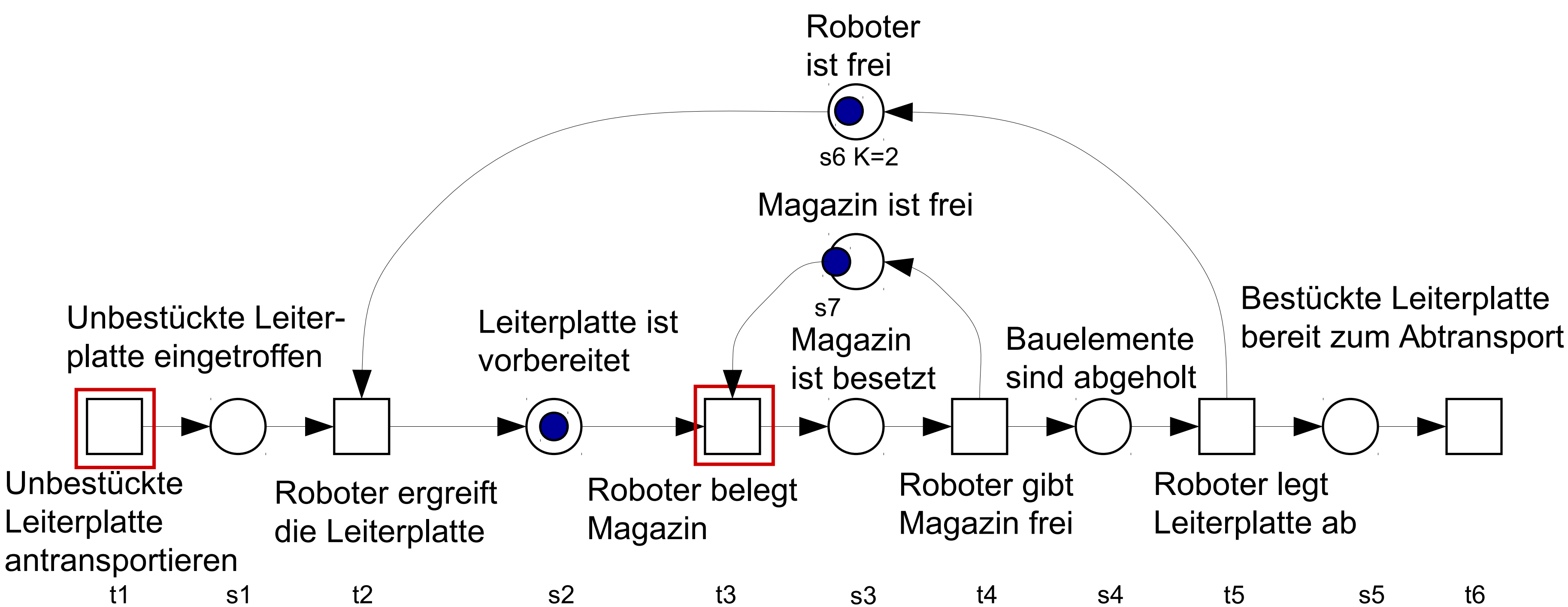
☐ bezeichnet aktivierte Transitionen. Möglicher nächster Zustand ?

Beispiel: Bestückungsroboter (M1)



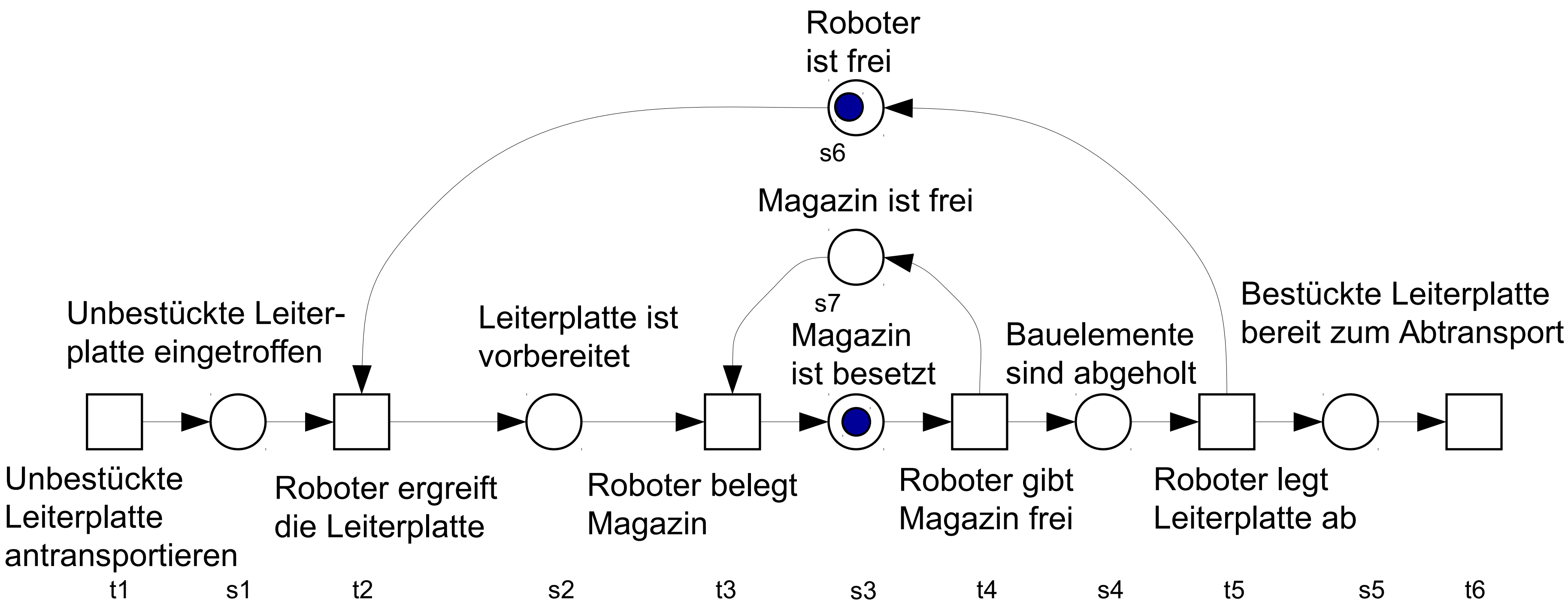
Welche Transition(en) aktiviert ?

Beispiel: Bestückungsroboter (M1)



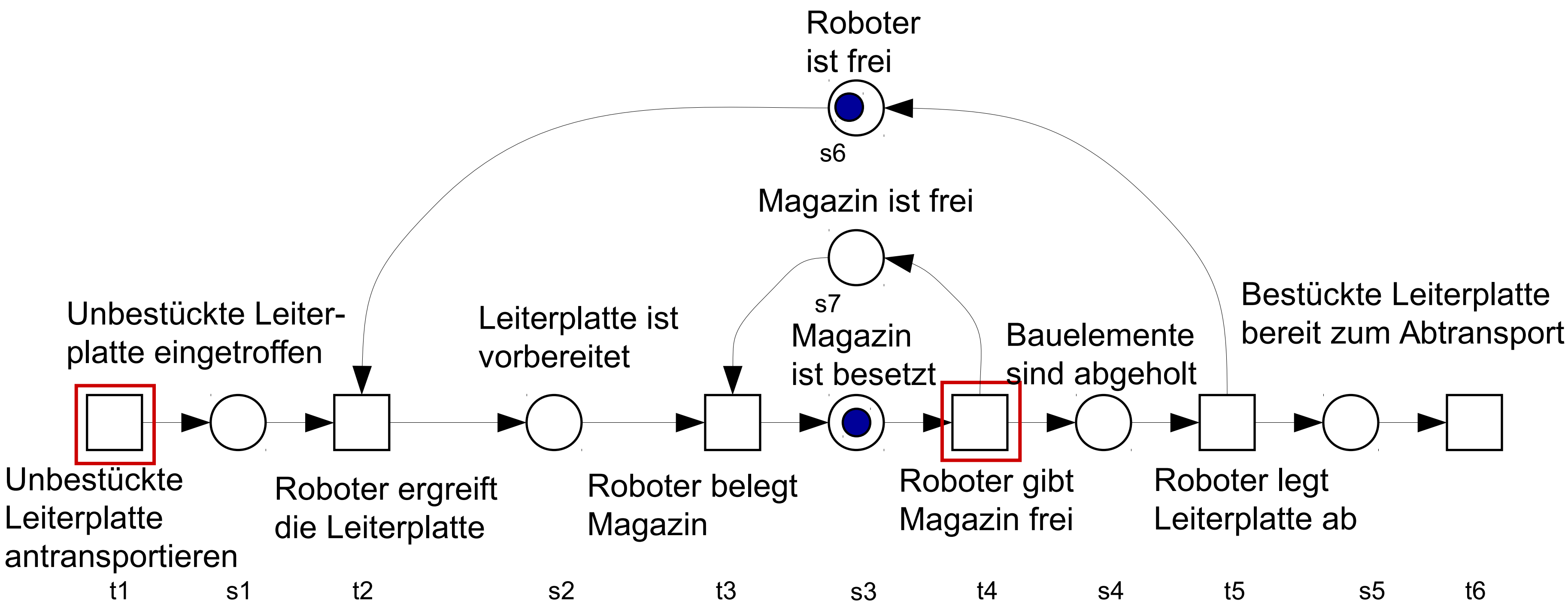
Möglicher nächster Zustand ?

Beispiel: Bestückungsroboter (M2)



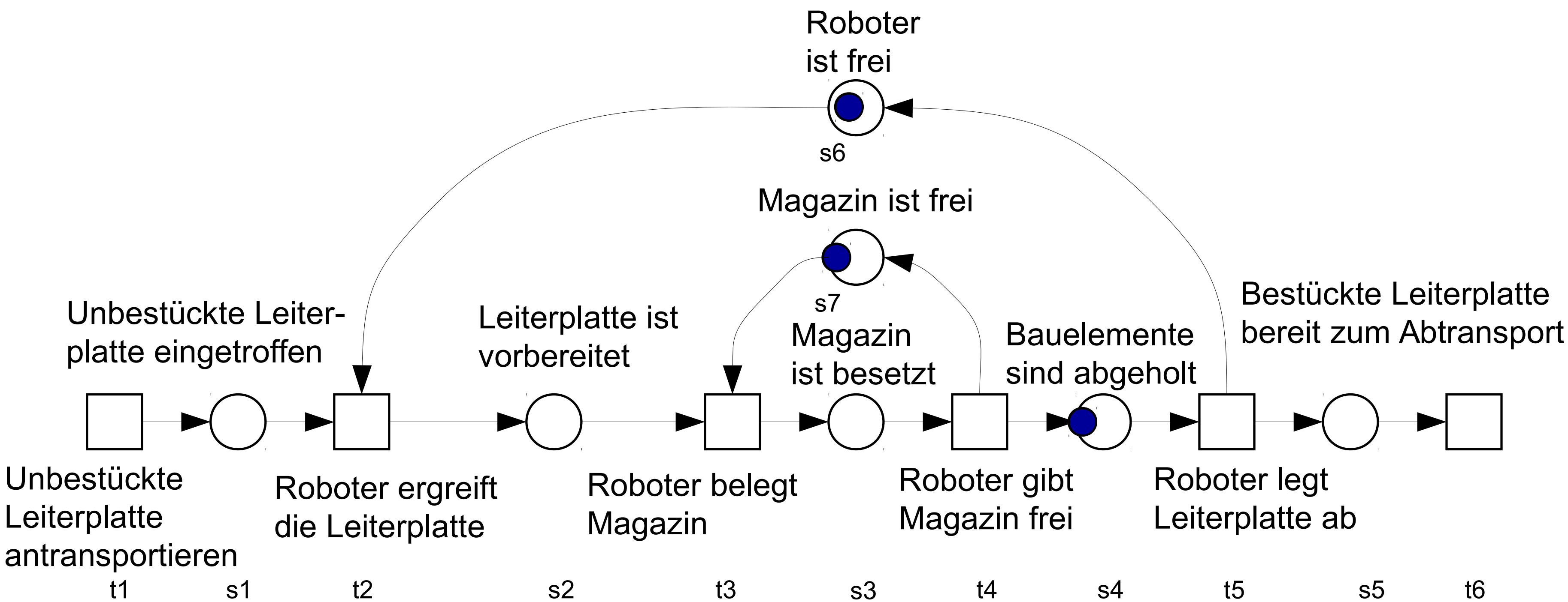
Welche Transition(en) aktiviert ?

Beispiel: Bestückungsroboter (M2)



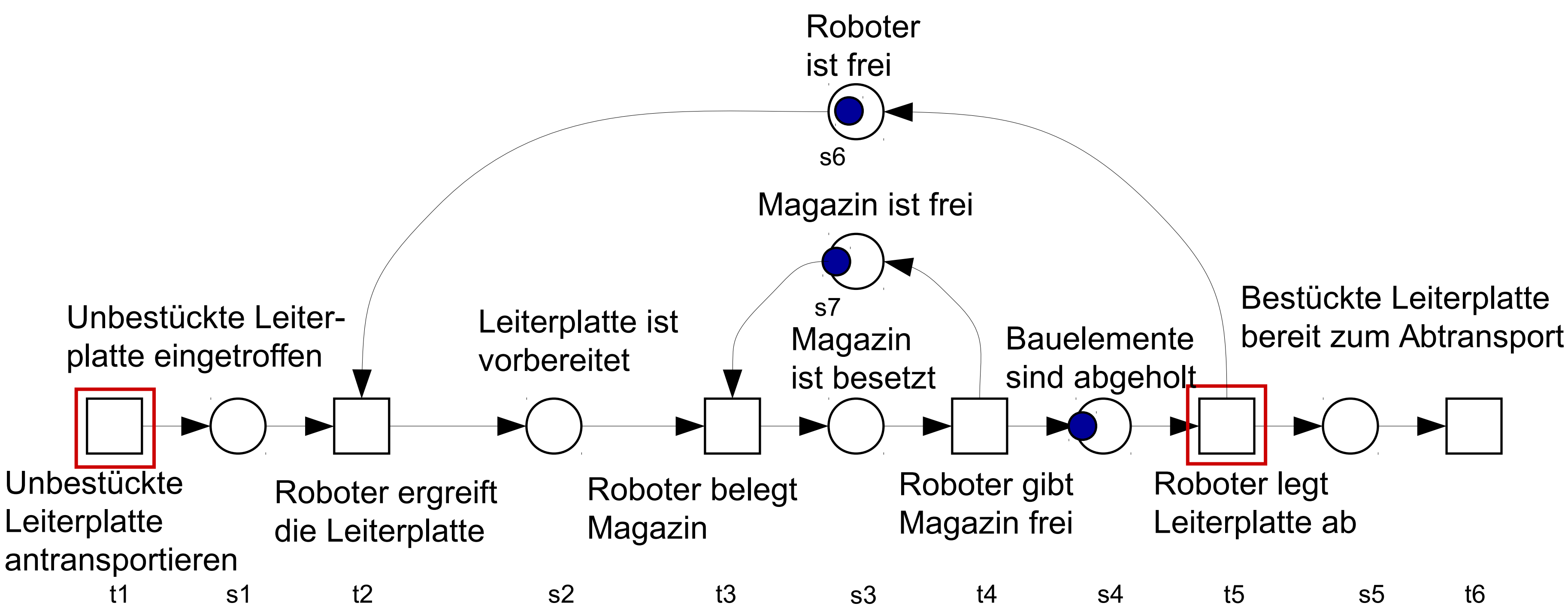
Möglicher nächster Zustand ?

Beispiel: Bestückungsroboter (M3)



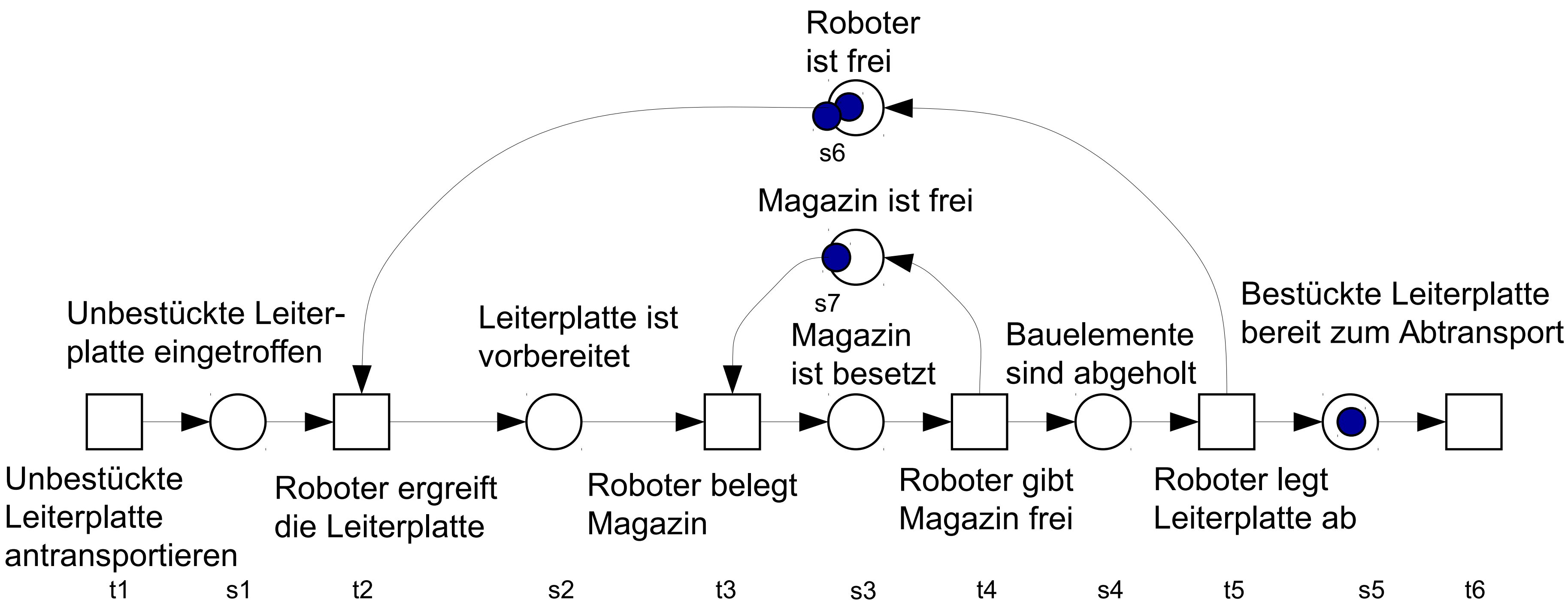
Welche Transition(en) aktiviert ?

Beispiel: Bestückungsroboter (M3)



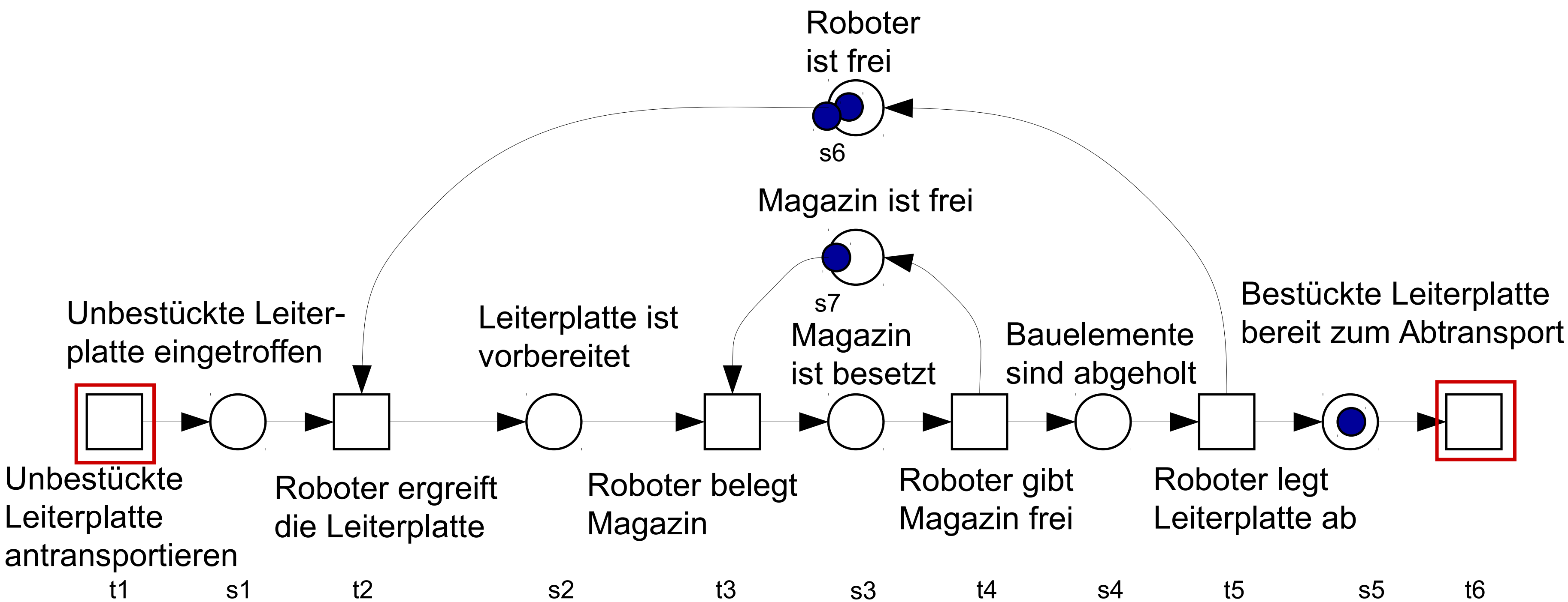
Möglicher nächster Zustand ?

Beispiel: Bestückungsroboter (M4)



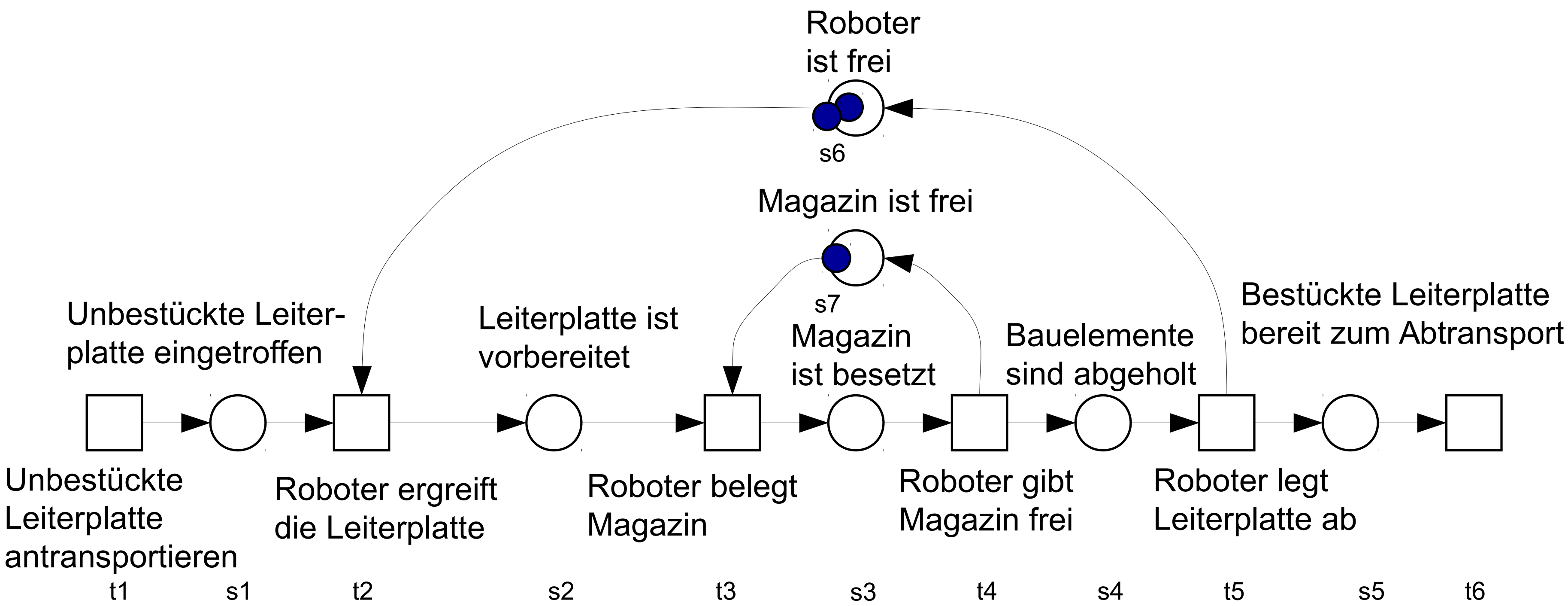
Welche Transition(en) aktiviert ?

Beispiel: Bestückungsroboter (M4)



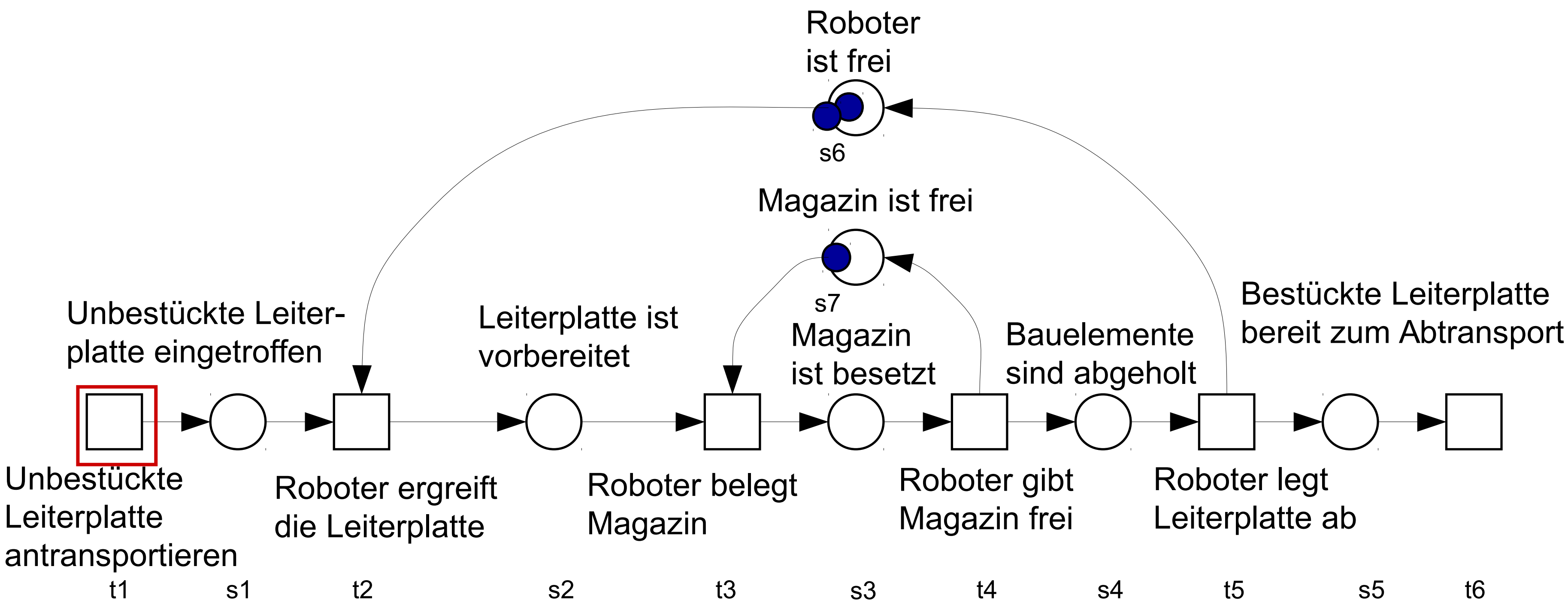
Möglicher nächster Zustand ?

Beispiel: Bestückungsroboter (M5)



Welche Transition(en) aktiviert ?

Beispiel: Bestückungsroboter (M5)



Usw. ...

Kapazitäten

- K: S $\rightarrow \mathrm{N} \cup \{\infty\}$ erklärt eine (möglicherweise unbeschränkte) **Kapazität** für jede Stelle.
- **Markierungen** M: S $\rightarrow \mathrm{N}_0$ müssen Kapazitäten respektieren, d.h. für jede Stelle $s \in S$ gilt: $M(s) \leq K(s)$.
- Transitionen sind bei Verwendung von Kapazitäten **nur dann** aktiviert, wenn **Folgemarkierung** Kapazitäten **respektiert.**

Definition Netz: Beispiel

Nachrichten-Queue: **Netz (S,T,F)** mit

- **S = {**empfangsbereit, Bereit Queue zu füllen, Queue gefüllt, Queue leer, Bereit zur Verarbeitung, Bereit zur Nachrichtenentnahme**}**
- **T = {**Nachricht annehmen, Queue füllen, Nachricht entnehmen, Nachricht verarbeiten**}**
- **F = {**(empfangsbereit, Nachricht annehmen), (Nachricht annehmen, Bereit Queue zu füllen), (Bereit Queue zu füllen, Queue füllen), (Queue füllen, empfangsbereit), (Queue füllen, Queue gefüllt), (Queue gefüllt, Nachricht entnehmen), (Nachricht entnehmen, Queue leer), (Queue leer, Queue füllen), (Bereit zur Nachrichtenentnahme, Nachricht entnehmen), (Nachricht entnehmen, Bereit zur Verarbeitung), (Bereit zur Verarbeitung, Nachricht verarbeiten), (Nachricht verarbeiten, Bereit zur Nachrichtenentnahme)**}**

Beispiel: Ampelschaltung

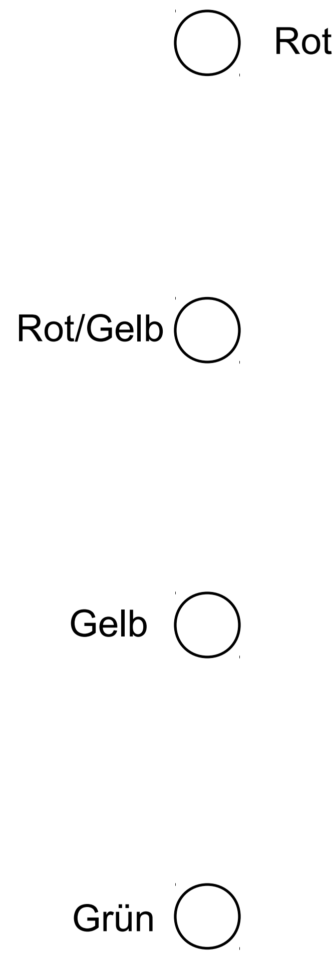
Wir wollen eine Ampelschaltung modellieren.

- Zwei Ampeln an einer Kreuzung
- Nord-Süd-Ampel und Ost-West-Ampel
- Abbieger-Ampel vernachlässigen wir
- Folgende Stellen gibt es:
    - Rot
    - Rot/Gelb
    - Gelb
    - Grün

Beispiel: Ampelschaltung

Wir wollen eine Ampelschaltung modellieren.

- Zwei Ampeln an einer Kreuzung
- Nord-Süd-Ampel und Ost-West-Ampel
- Abbieger-Ampel vernachlässigen wir
- Folgende Stellen gibt es:
  - Rot
  - Rot/Gelb
  - Gelb
  - Grün

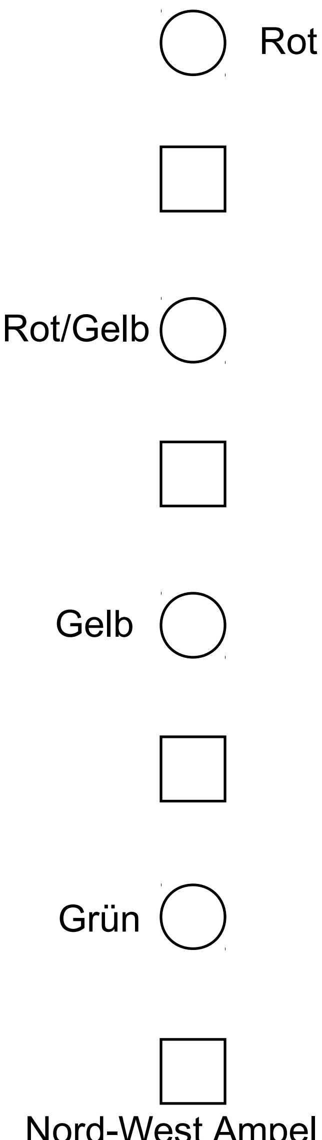


Nord-West Ampel

Beispiel: Ampelschaltung

Wir wollen eine Ampelschaltung modellieren.

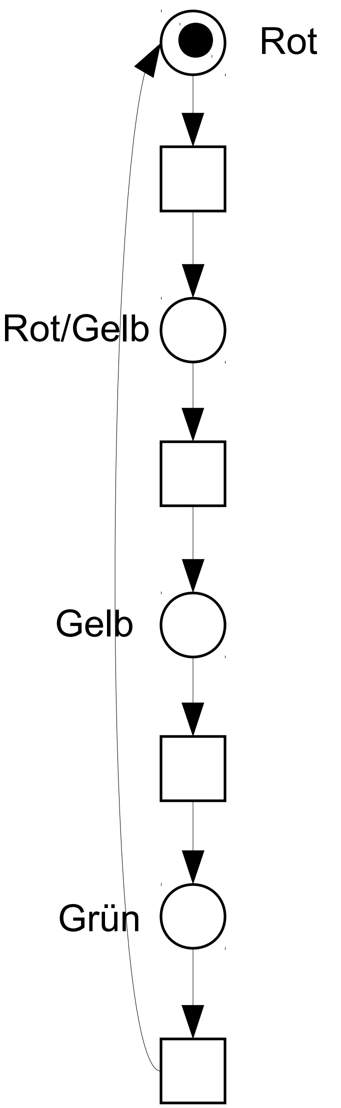
- Zwei Ampeln an einer Kreuzung
- Nord-Süd-Ampel und Ost-West-Ampel
- Abbieger-Ampel vernachlässigen wir
- Folgende Stellen gibt es:
  - Rot
  - Rot/Gelb
  - Gelb
  - Grün



Beispiel: Ampelschaltung

Wir wollen eine Ampelschaltung modellieren.

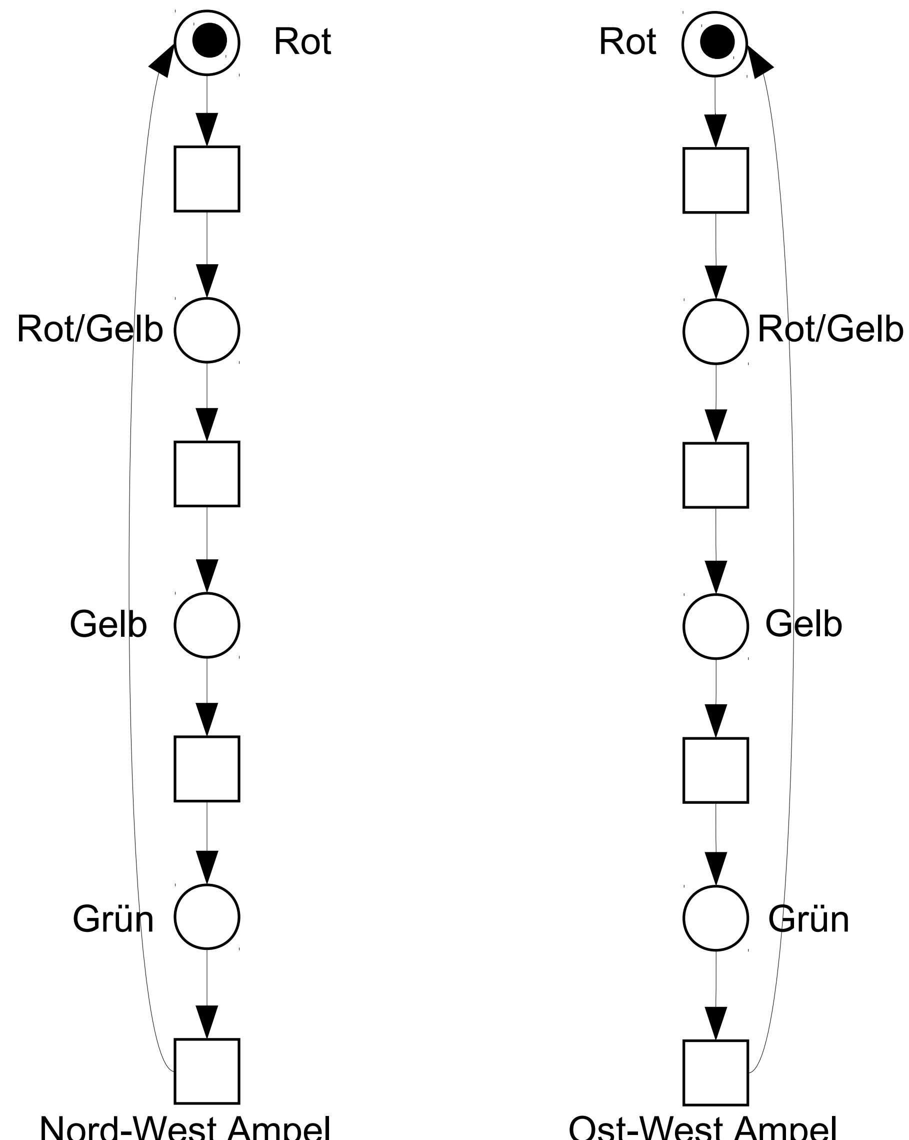
- Zwei Ampeln an einer Kreuzung
- Nord-Süd-Ampel und Ost-West-Ampel
- Abbieger-Ampel vernachlässigen wir
- Folgende Stellen gibt es:
  - Rot
  - Rot/Gelb
  - Gelb
  - Grün



Beispiel: Ampelschaltung

Wir wollen eine Ampelschaltung modellieren.

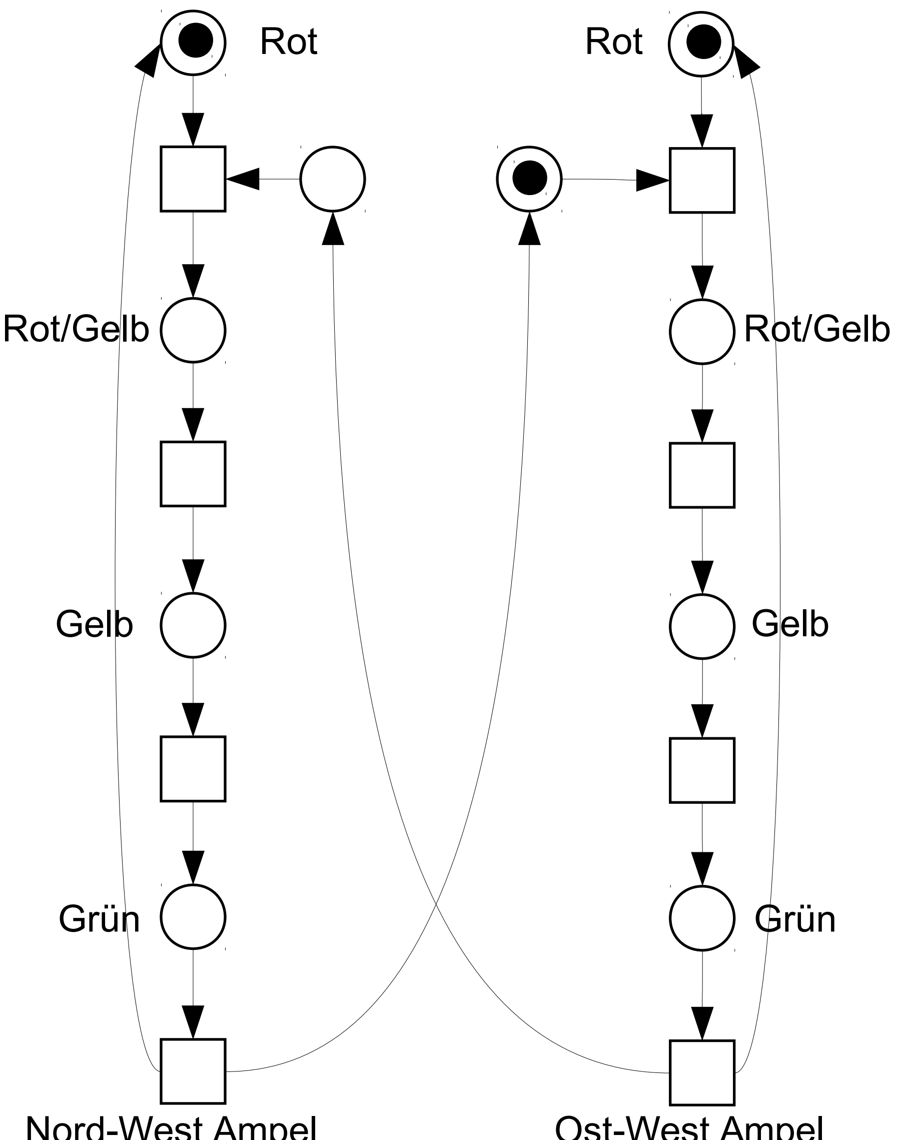
- Zwei Ampeln an einer Kreuzung
- Nord-Süd-Ampel und Ost-West-Ampel
- Abbieger-Ampel vernachlässigen wir
- Folgende Stellen gibt es:
  - Rot
  - Rot/Gelb
  - Gelb
  - Grün



Beispiel: Ampelschaltung

Wir wollen eine Ampelschaltung modellieren.

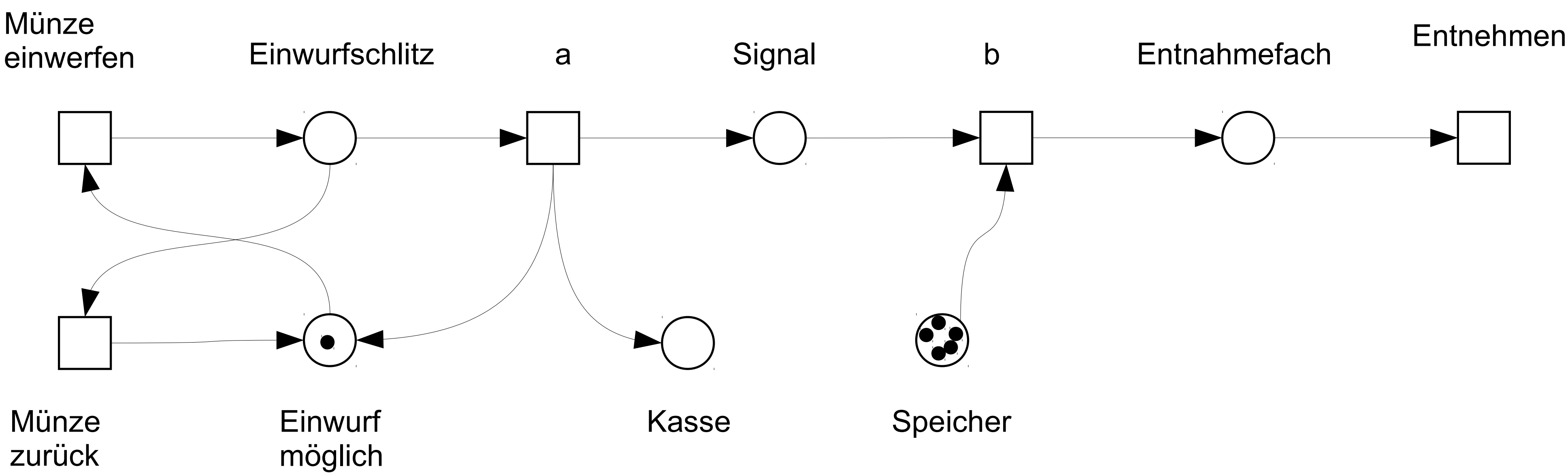
- Zwei Ampeln an einer Kreuzung
- Nord-Süd-Ampel und Ost-West-Ampel
- Abbieger-Ampel vernachlässigen wir
- Folgende Stellen gibt es:
  - Rot
  - Rot/Gelb
  - Gelb
  - Grün



Zweites Beispiel

Was könnte folgendes Petri-Netz modellieren?

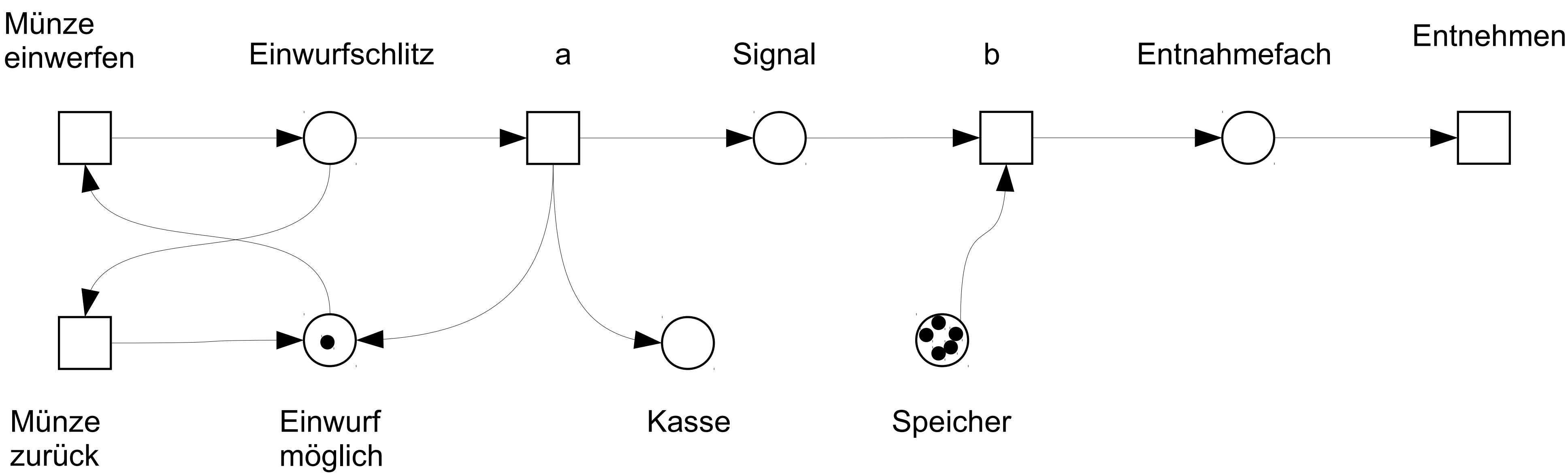
Was könnten die einzelnen Token repräsentieren?



Zweites Beispiel

Was könnte folgendes Petri-Netz modellieren?

Was könnten die einzelnen Token repräsentieren?



Antwort:
Es könnte sich z.B. um einen Getränkeautomaten handeln. Dabei können die Token einerseits „Münzen“ oder auch „Getränkeflaschen“ repräsentieren.

Methodik

Kein Standard zu Erstellung von Petri-Netzen vorhanden.

Vorgeschlagenes Vorgehen (aus [Bal00]):

1. Stellen und Transitionen auf hohem Abstraktionsniveau identifizieren.
2. Beziehungen ermitteln.
3. Verfeinerung und Ergänzung.
4. Festlegung der Objekte.
5. Schaltregeln identifizieren.
6. Netztyp festlegen.
7. Anfangsmarkierung festlegen.
8. Analyse, Simulation.

Performanzanalyse, z.B.: Simulation in BPM|one