

# 4. Zustandsorientierte Modellierung

4.1 Zustandsautomat

4.2 Petrinetz

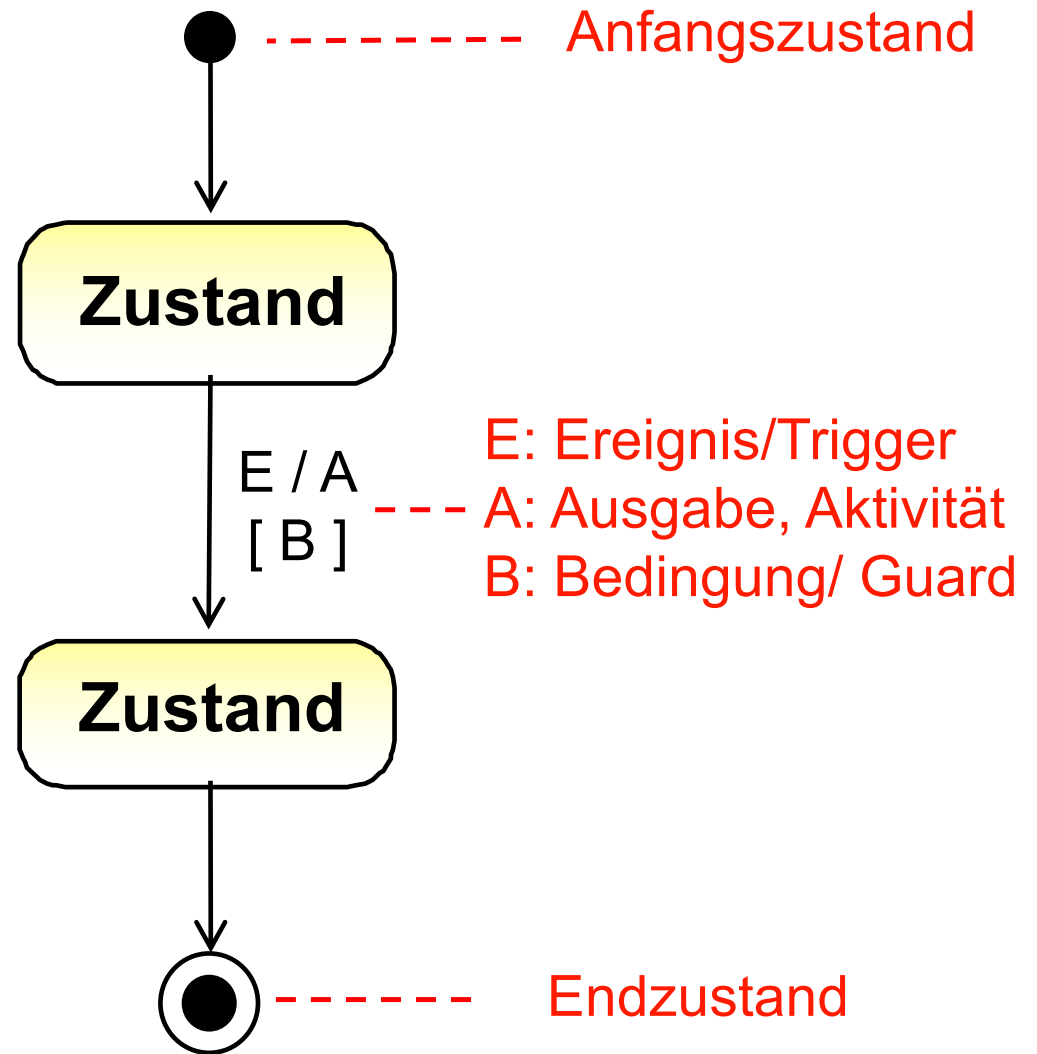
# Zustandsautomaten

- Bei vielen Systemen hängt Ergebnis oder Ausgabe nicht nur von Eingabe oder Ereignis ab, sondern auch von Historie, die das System bisher durchlaufen hat
- Automat = Technisches Gerät, das zu einer Eingabe ein bestimmtes Ergebnis ausgibt
- Zustandsautomat ("endlicher Automat") besteht aus einer endlichen Anzahl von internen Konfigurationen (= Zuständen)
- Entwickelt von Huffman, Moore und Mealy in den Jahren von 1954 bis 1956
- Idee entstand als "Abfallprodukt" der Forschung auf dem Gebiet formaler Sprachen (→ "Automatentheorie")

# Zustandsautomaten

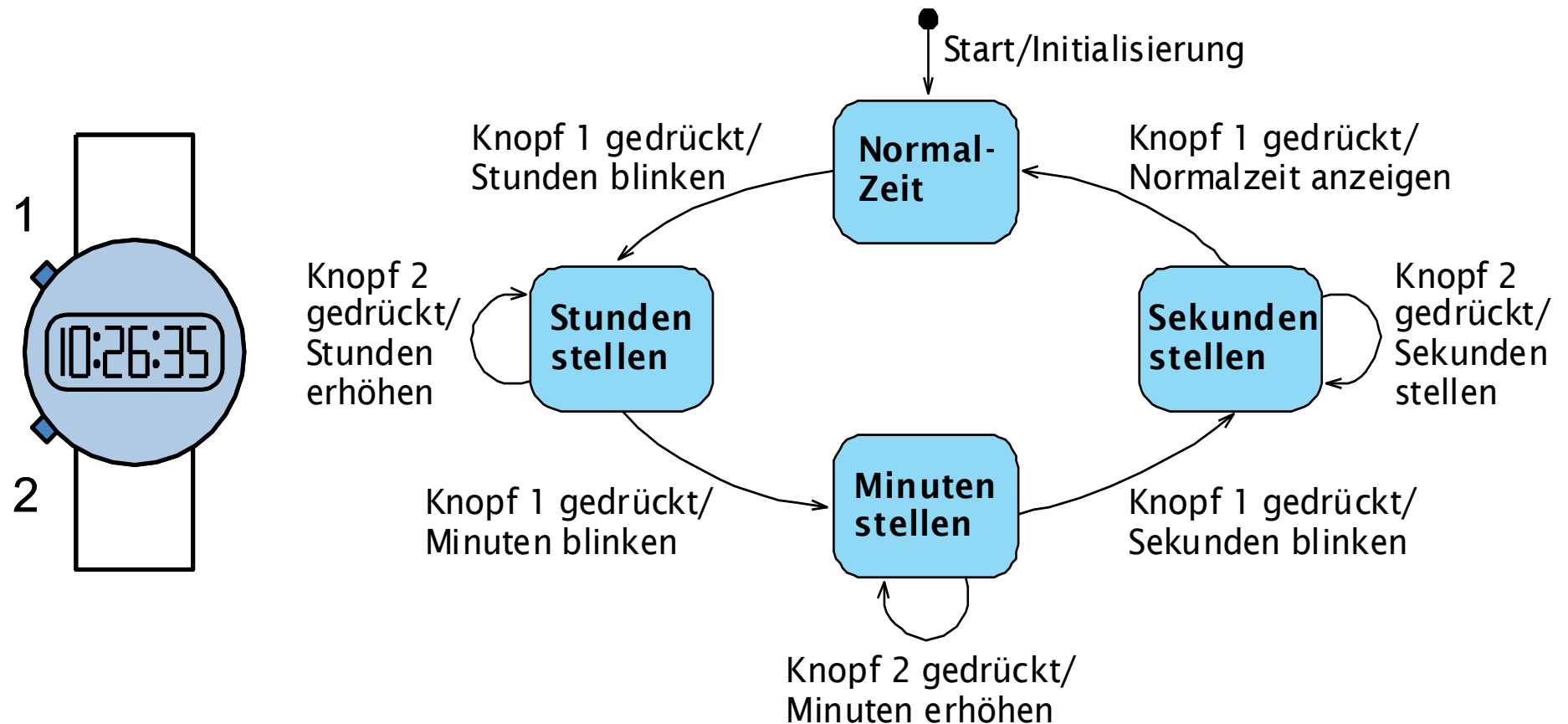
## Notation

- Zustandsautomat:  
Endlicher Automat
  - Endliche, nicht-leere Menge von Zustände
  - Endliche, nicht-leere Menge von Ereignissen
  - Zustandsübergänge
  - Anfangszustand
  - Menge von Endzuständen



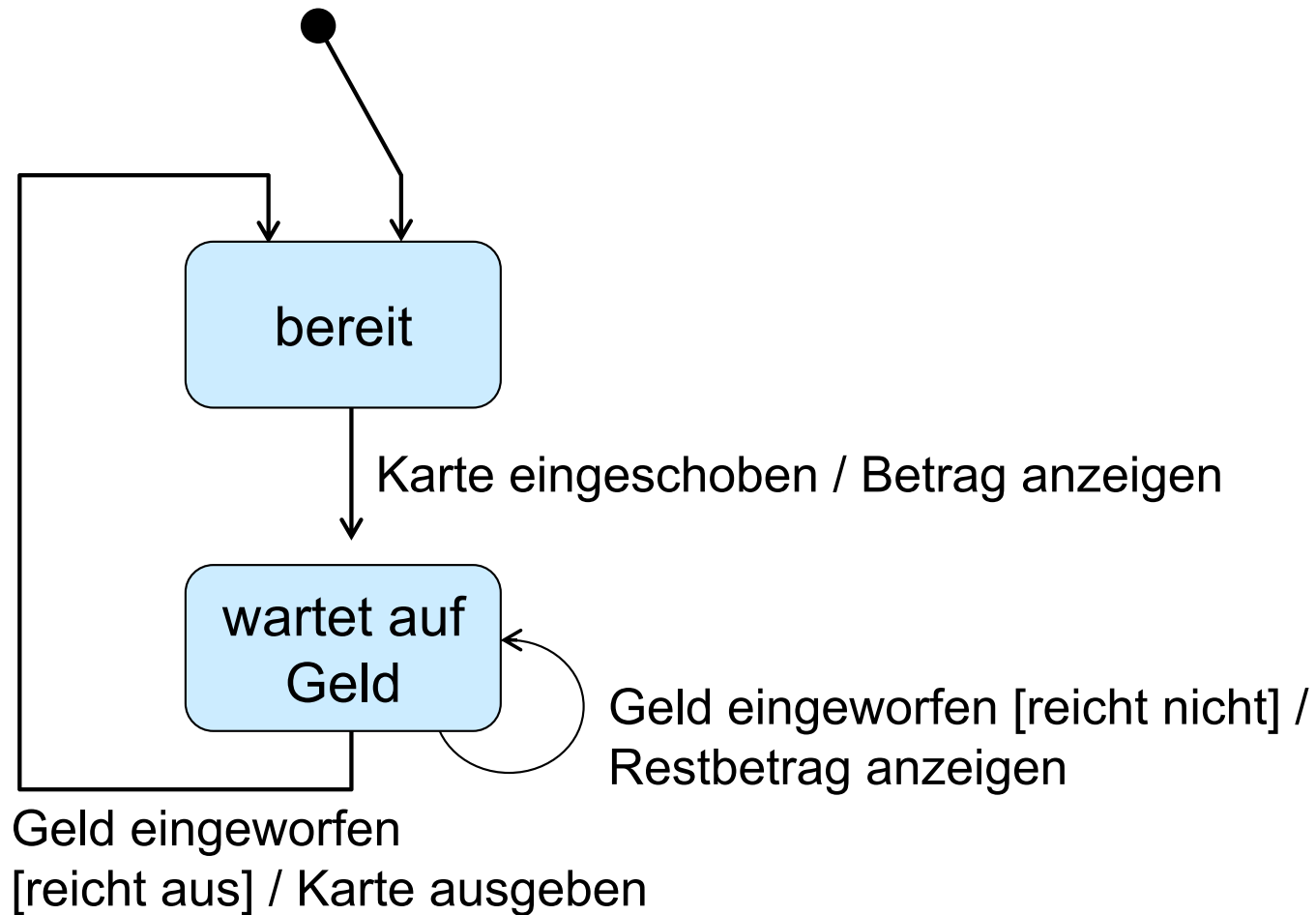
# Zustandsautomat

## Beispiel



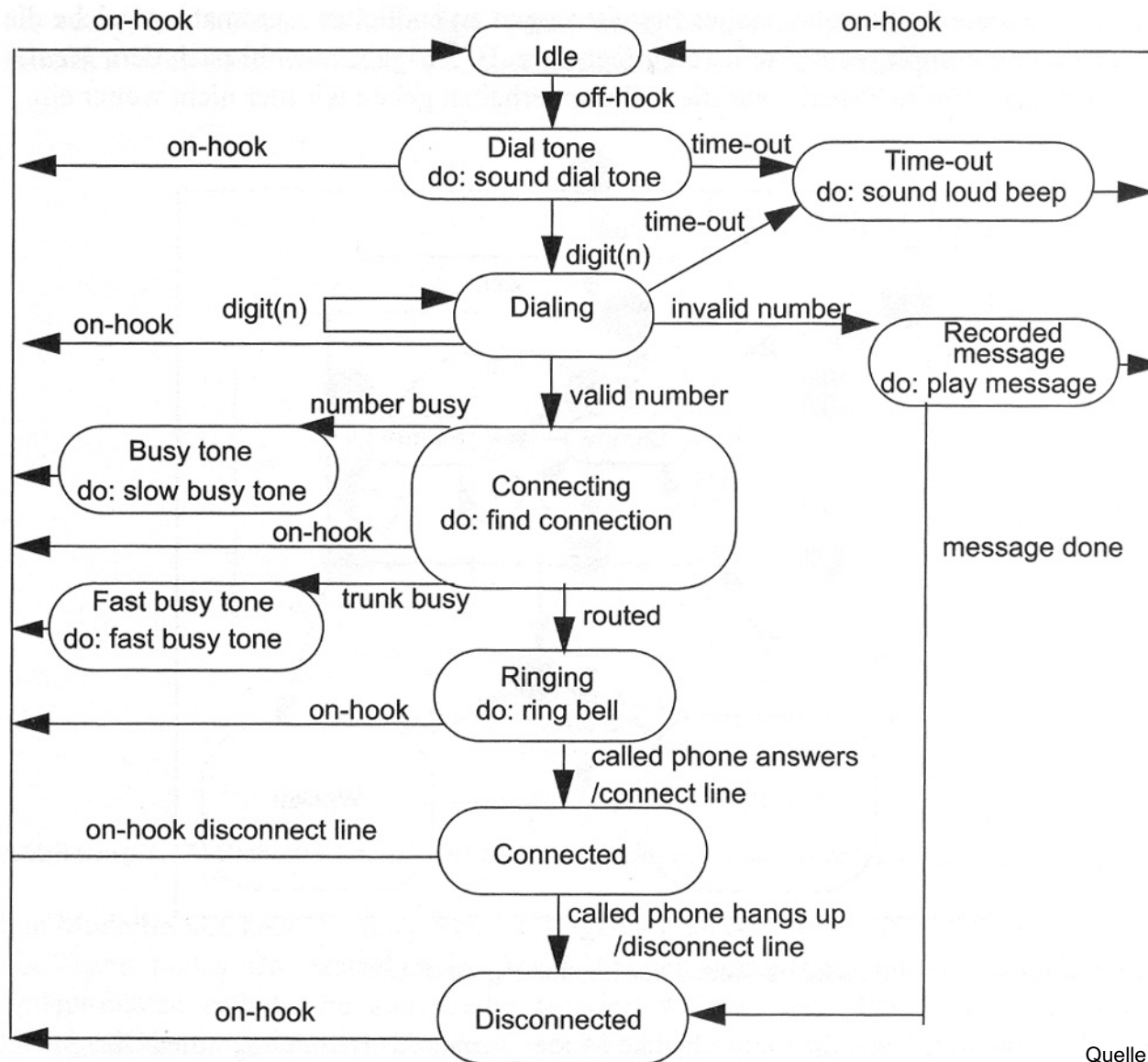
# Zustandsautomat

## Beispiel Kartenautomat im Parkhaus



# Zustandsautomat

## Beispiel Telefon



Quelle: Kastens, Kleine Büning: Modellierung, Hanser

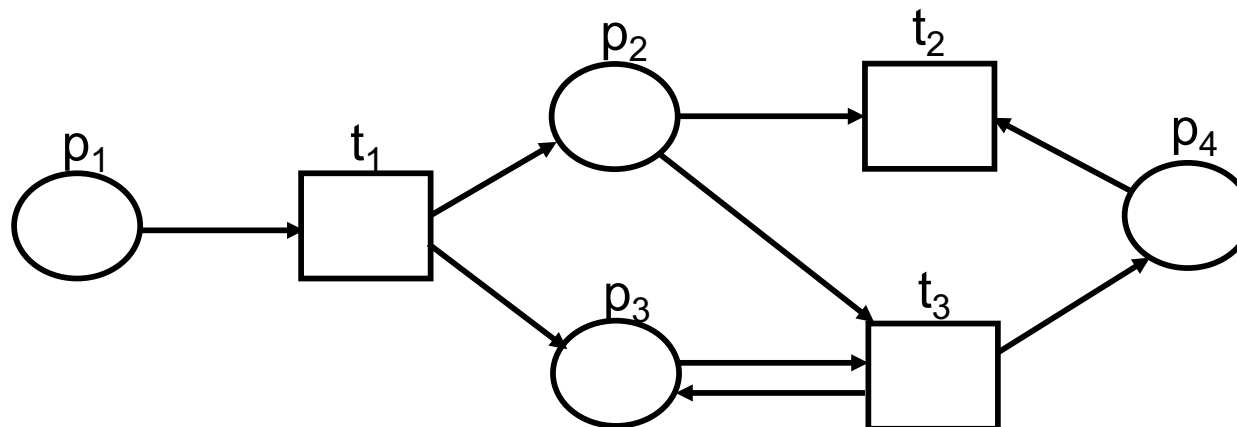
# Petrinetze

- Prof. Dr. Carl Adam Petri, 1926-2010
  - Deutscher Mathematiker
- "Erfinder" der Petrinetze  
Zahlreiche Forschungen (> 10.000 Publikationen)
- Forschungs-"Community" bestand bis 1985 hauptsächlich aus Theoretikern
- Danach zunehmende Bedeutung für die Praxis
  - Entwurf von Systemen
  - Analyse
  - Präsentation
  - Simulation



# Notation von Petrinetzen

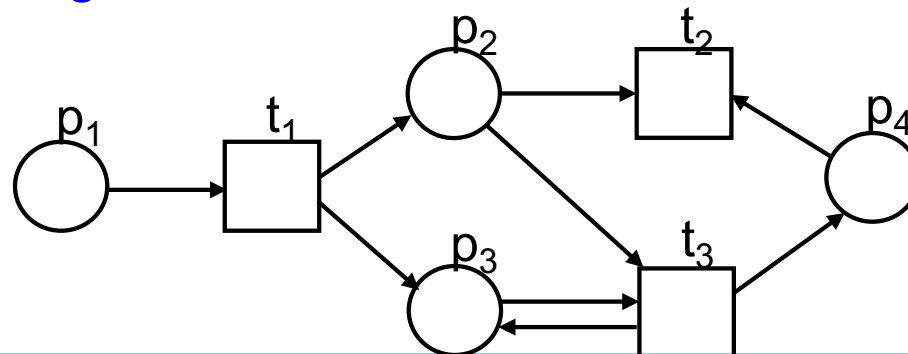
- Petrinetze basieren auf gerichteten Graphen mit zwei Knotentypen
- Ein Petrinetz ist ein graphisches System und besteht aus
  - Stellen (○)
  - Transitionen (□)
  - gerichteten Kanten (→)
  - Annotationen
- Stellen sind passive (= statische) Komponenten
- Transitionen sind aktive (= dynamische) Komponenten





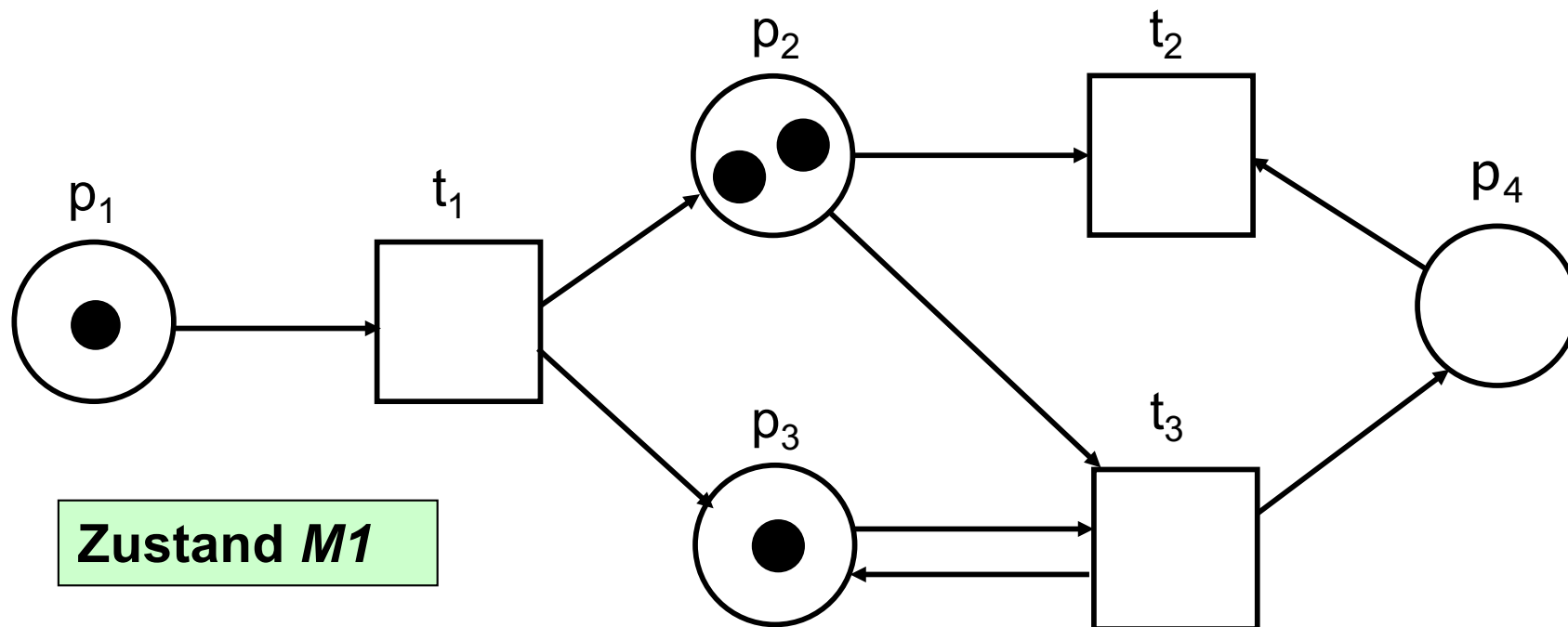
# Ein- und Ausgabeknoten

- Eine Stelle heißt *Eingabestelle* oder *Vorbereich* einer Transition, wenn eine Kante von der Stelle zur Transition verläuft
- Eine Stelle heißt *Ausgabestelle* oder *Nachbereich* einer Transition, wenn eine Kante von der Transition zur Stelle verläuft
- Analog definiert: Eingabetransition und Ausgabetransition
- Beispiel:
  - $p_4$  ist Eingabestelle oder Vorbereich von  $t_2$
  - $p_3$  ist Ausgabestelle oder Nachbereich von  $t_1$  und  $t_3$
  - $t_3$  ist Eingabetransition oder Vorbereich von  $p_3$  und  $p_4$
  - $t_1$  ist Ausgabetransition oder Nachbereich von  $p_1$



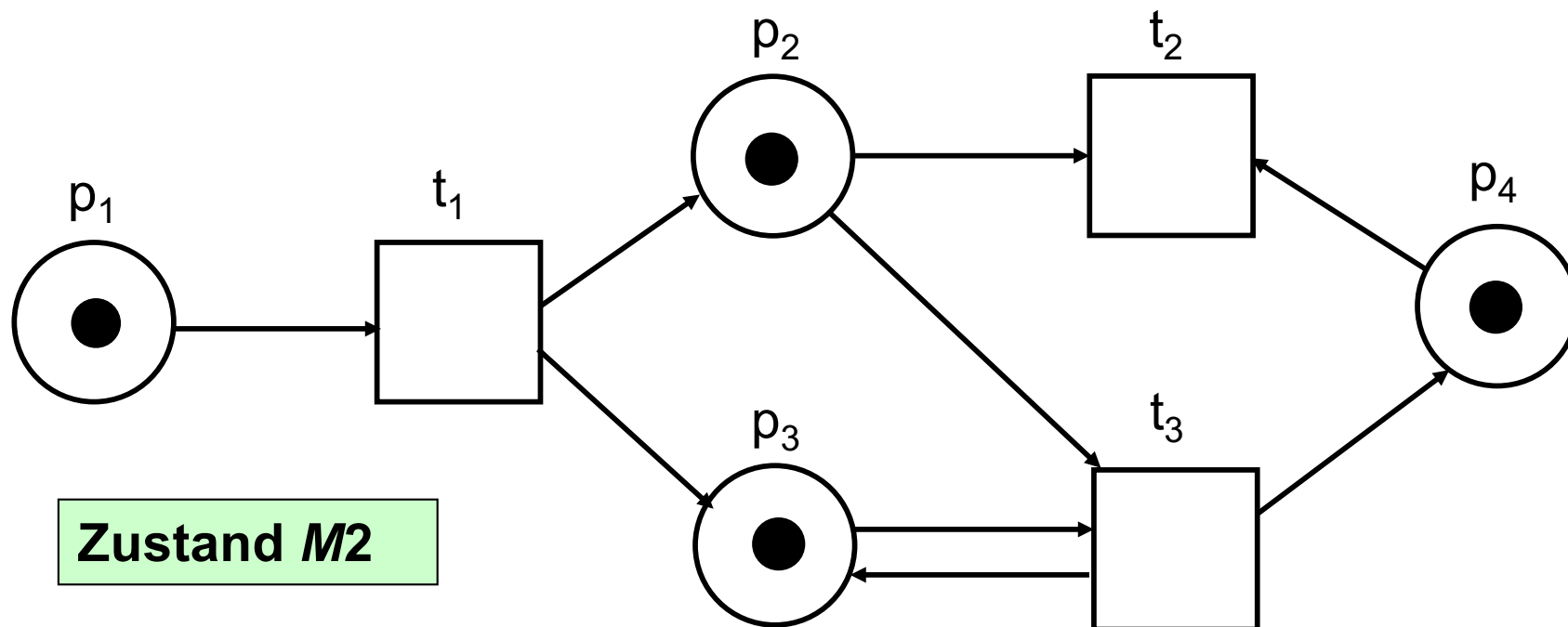
# Marken und Markierungen

- Jede Stelle eines Petrinetzes kann null, eine oder mehrere Marken ( ● ) enthalten – Englisch: Token
- Ein Zustand (Markierung) eines Petrinetzes ist definiert durch die Markenverteilung auf den Stellen



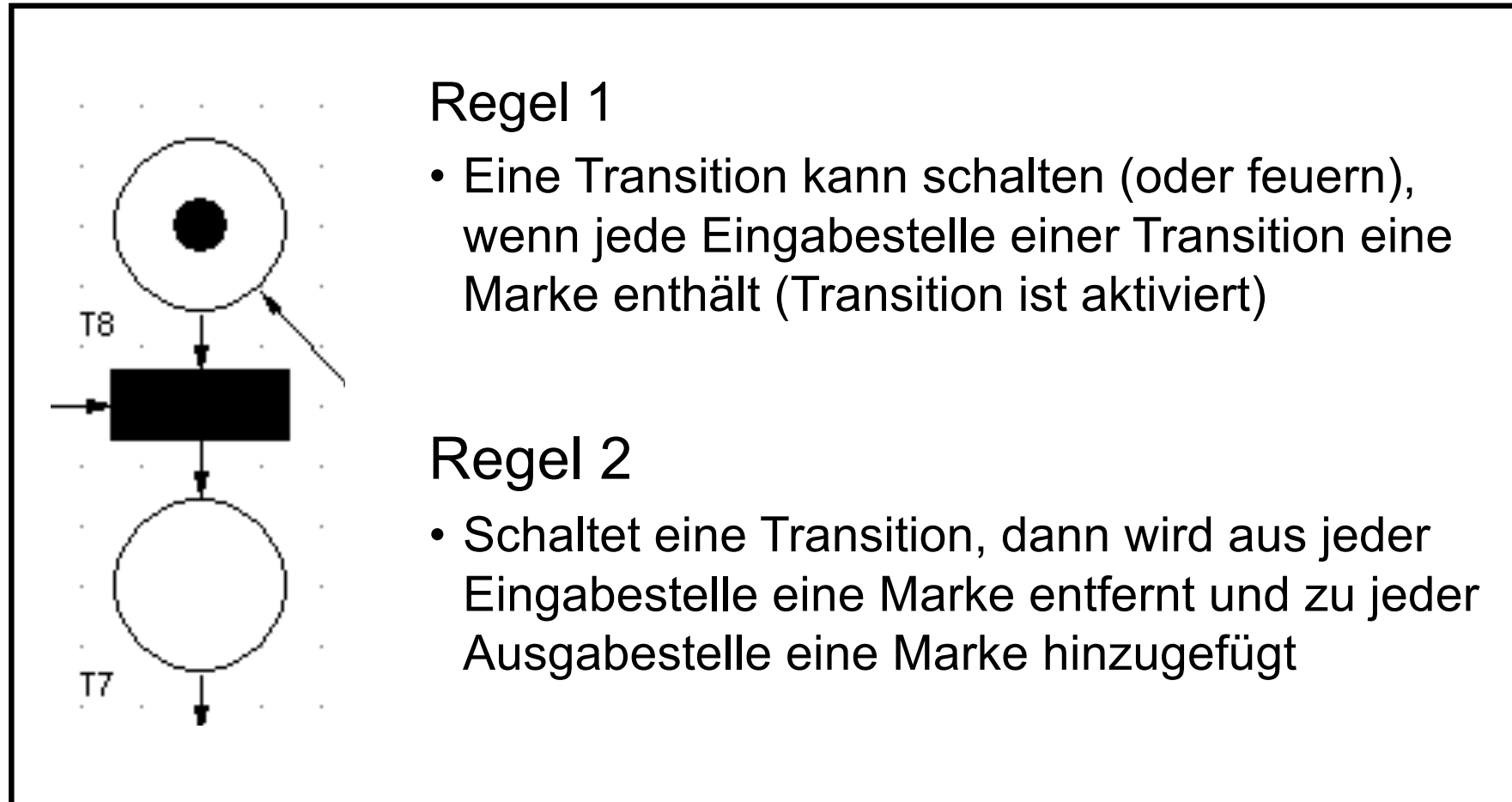
# Marken und Markierungen

- Jede Stelle eines Petrinetzes kann null, eine oder mehrere Marken ( ● ) enthalten – Englisch: Token
- Ein Zustand (Markierung) eines Petrinetzes ist definiert durch die Markenverteilung auf den Stellen



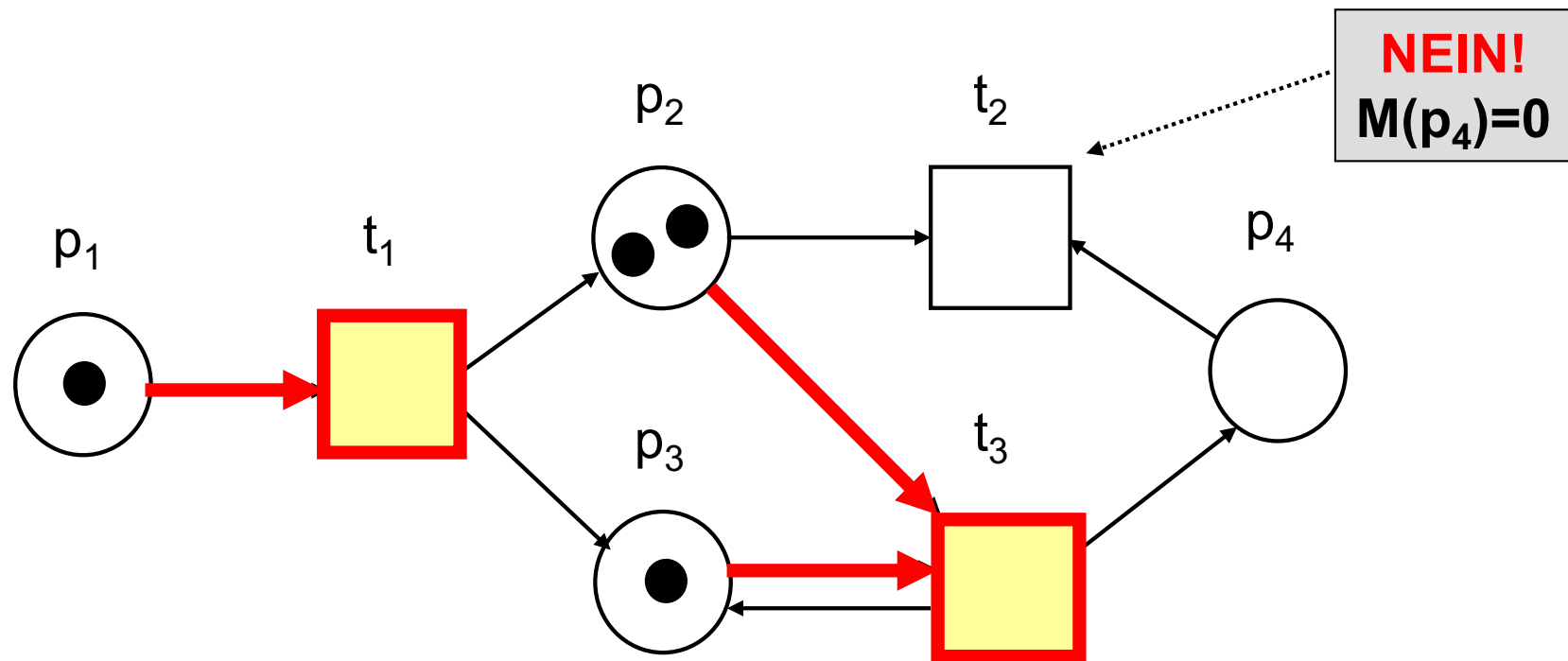
# Petrinetze

## Schaltregeln



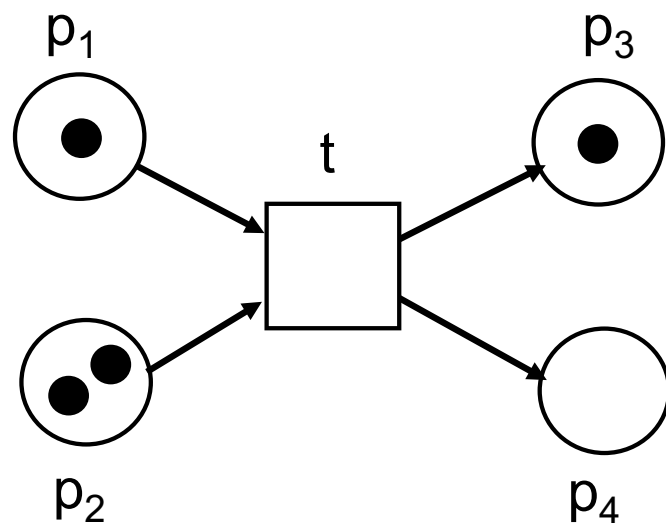
# Aktivierungsregel

- Es kann unter einer Markierung mehrere aktivierte Transitionen geben
- Beispiel: Welche Transitionen sind aktiviert?

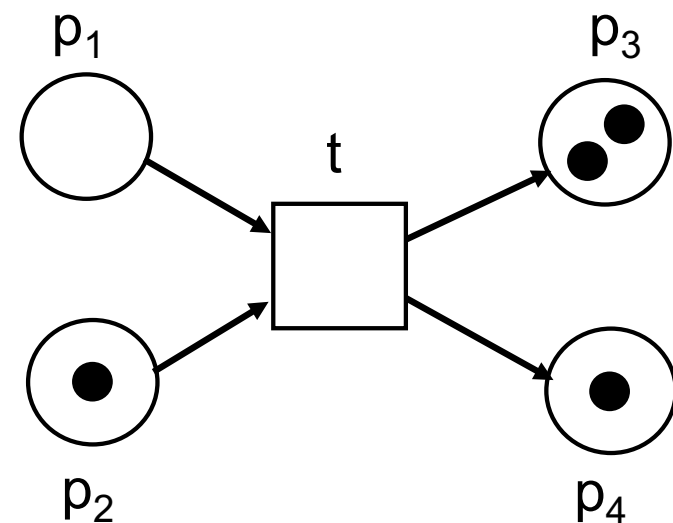
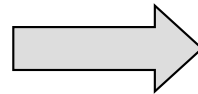


# Schaltregel

- Eine aktivierte Transition kann schalten
- Schaltregel: Wenn eine Transition schaltet, wird
  - je eine Marke von jeder Eingabestelle entfernt und
  - je eine Marke auf jeder Ausgabestelle abgelegt



$M(p_1)=M(p_3)=1, M(p_2)=2, M(p_4)=0$



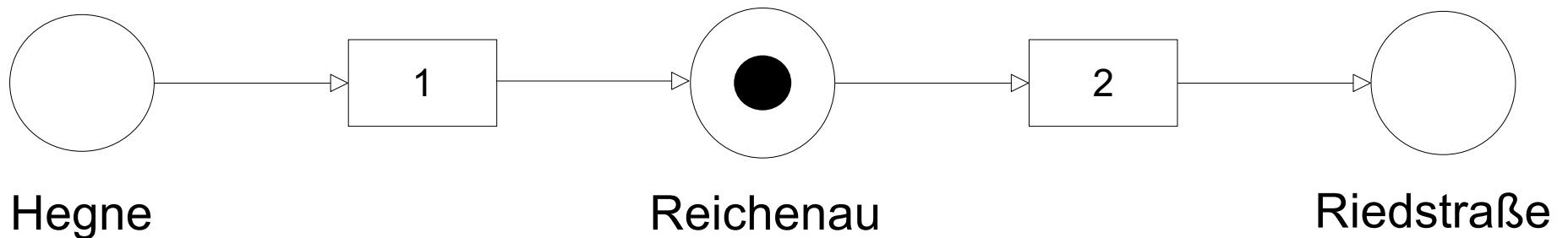
$M'(p_1)=0, M'(p_2)=M'(p_4)=1, M'(p_3)=2$

# Petrinetzmodelle

- Die Elemente eines Petrinetzes sollten einem "realen" System sorgfältig zugeordnet werden:
- Stellen
  - = Puffer, Orte, Bedingungen, Zustände
- Transitionen
  - = Ereignisse, Transformationen, ...
- Marken
  - = Objekte (Personen, Dinge, Maschinen), Informationen, Dokumente, Gültigkeit von Bedingungen oder Vorliegen von Zuständen
- Aktivierung
  - = Ein Ereignis kann im aktuellen Systemzustand eintreten
- Schalten
  - = Ein Ereignis tritt tatsächlich ein und verändert den aktuellen Systemzustand

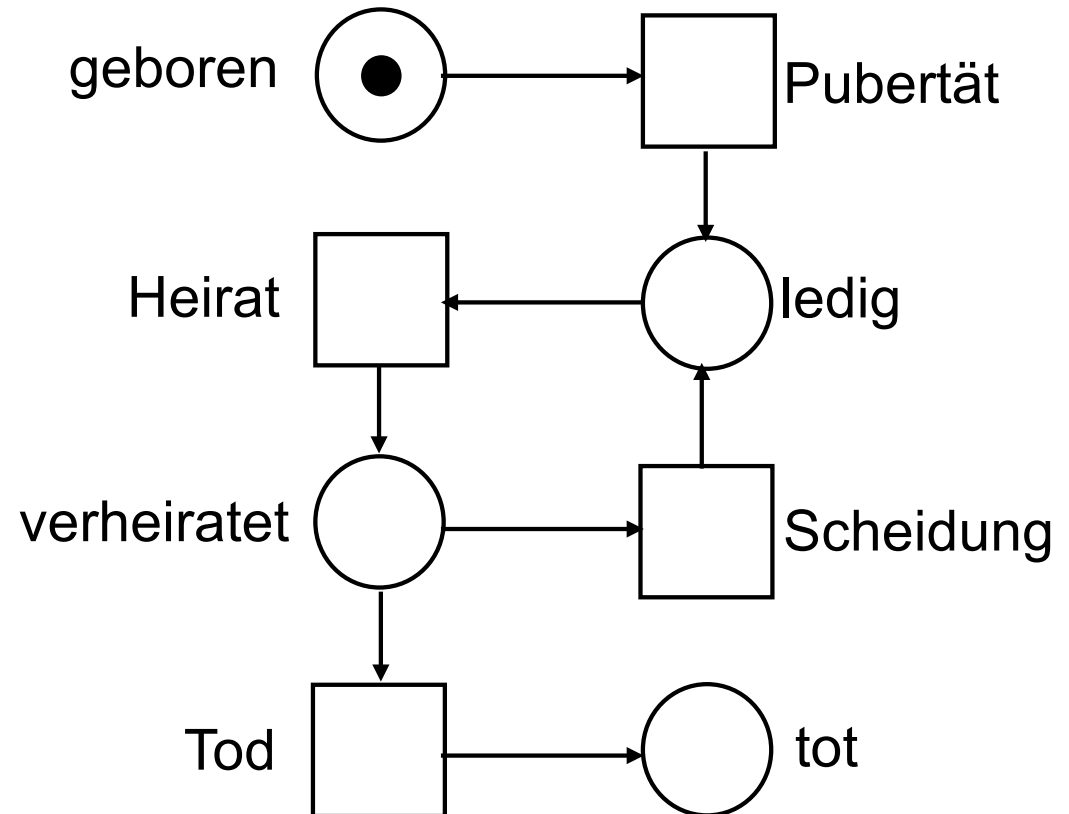
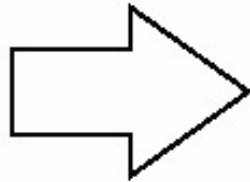
# Elementare Verknüpfungen

- Petri - Netze erlauben die Modellierung sequentieller, alternativer (verzweigender) und nebenläufiger Prozesse
- Folgende Auflistung fasst elementare Verknüpfungen von Stellen, Kanten und Transitionen zusammen

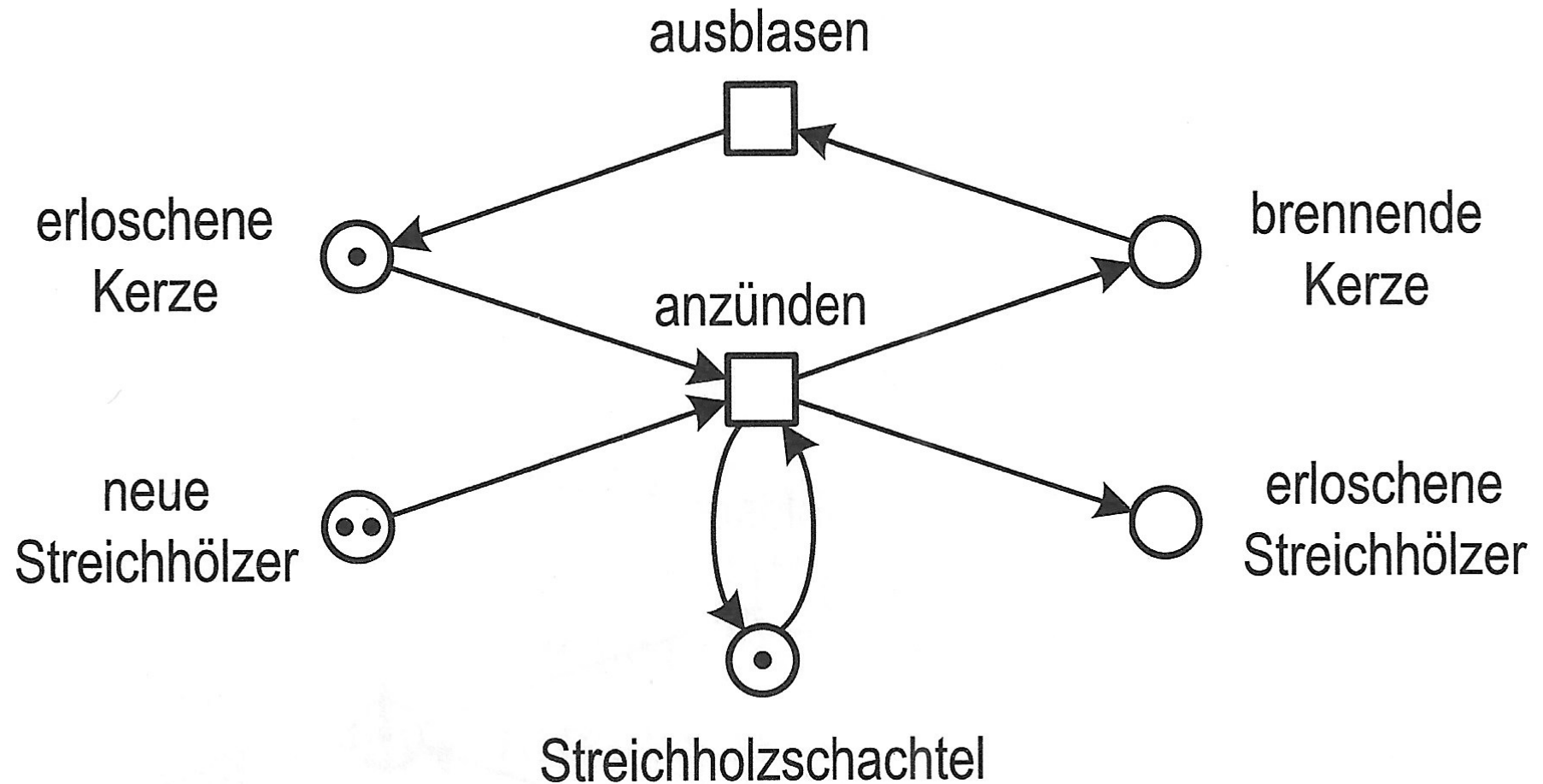




# Petrinetz – Beispiele



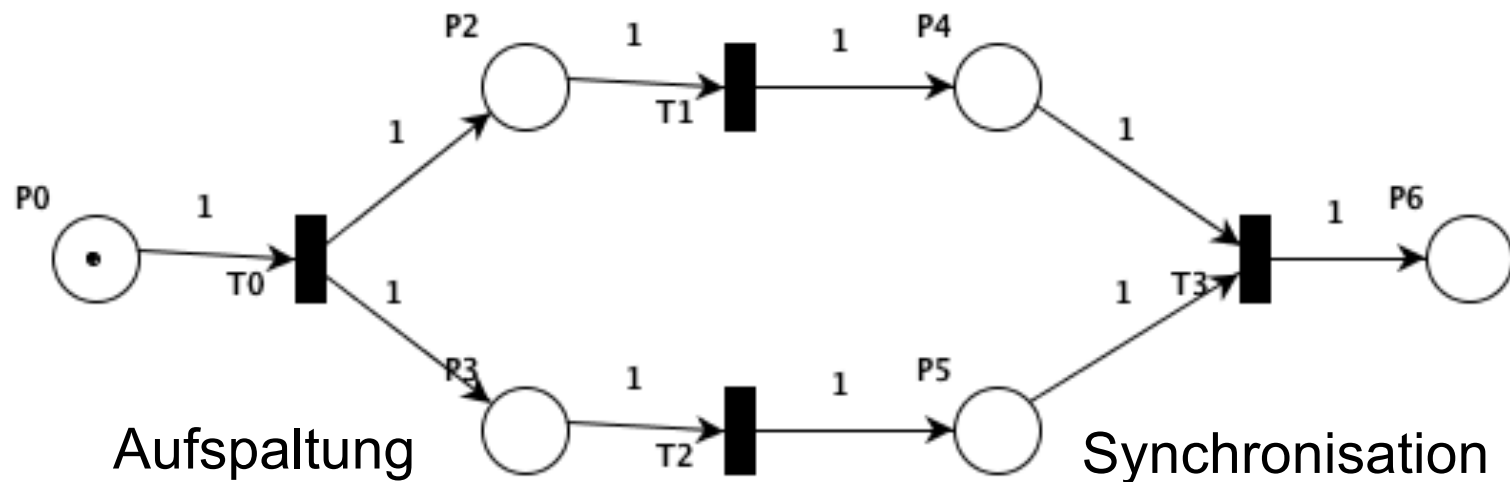
# Petrinetz – Beispiele



Quelle: Informatik Spektrum 37\_3\_2014

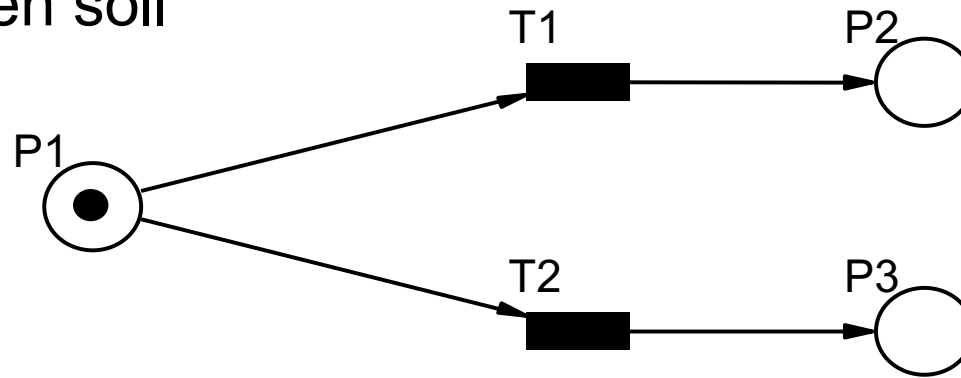
# Modellierung von Nebenläufigkeit

- Bei Nebenläufigkeit ist oft keine Reihenfolge festgelegt

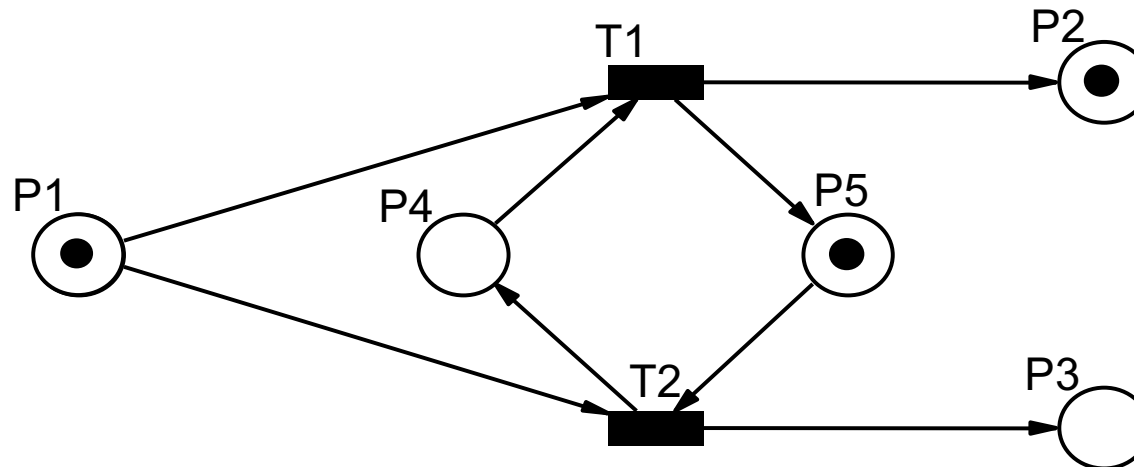


# Branch Conflict

- Schaltregel gibt keinen Aufschluss darüber, welche Transition schalten soll

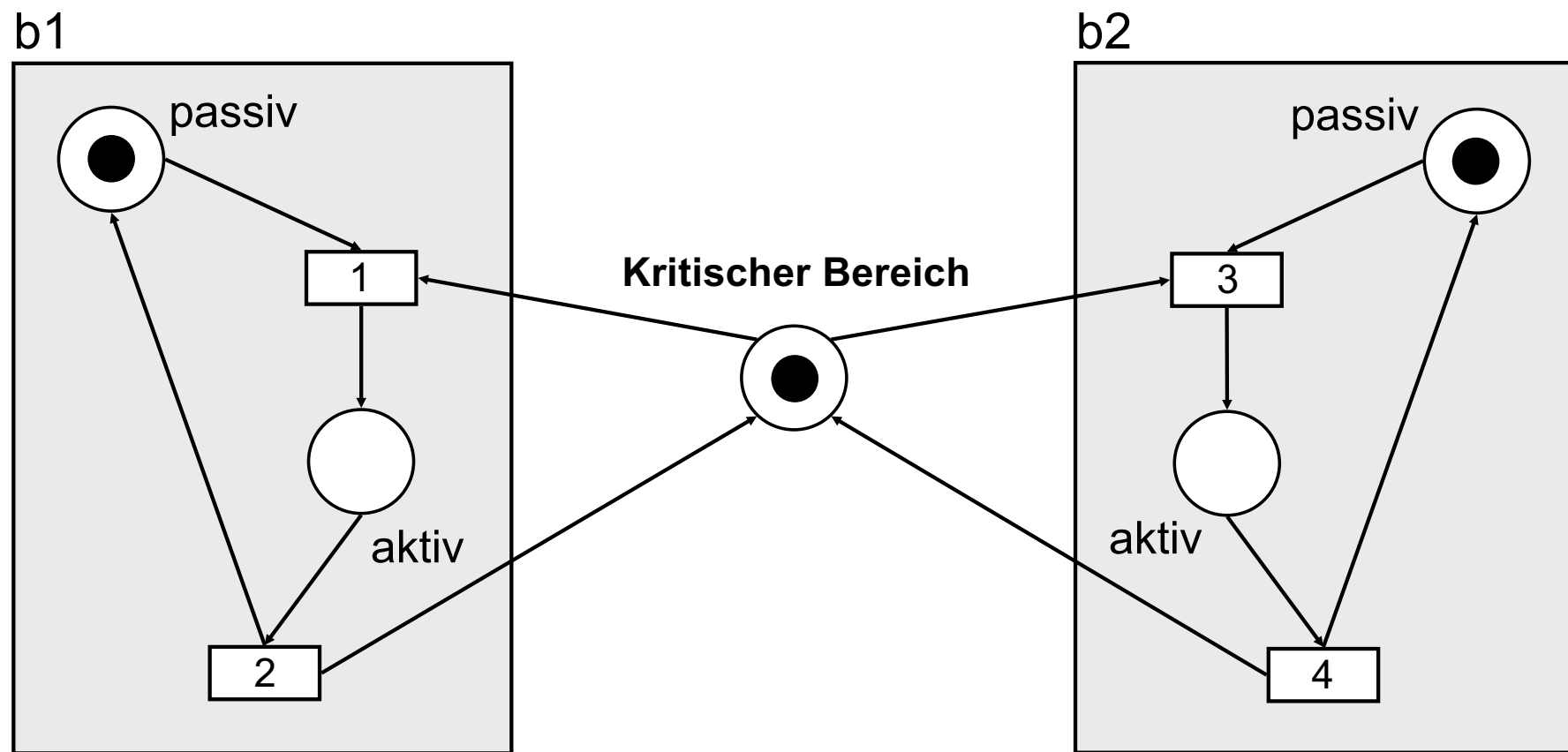


- Behebung des Problems: z.B. abwechselnd Schalten lassen



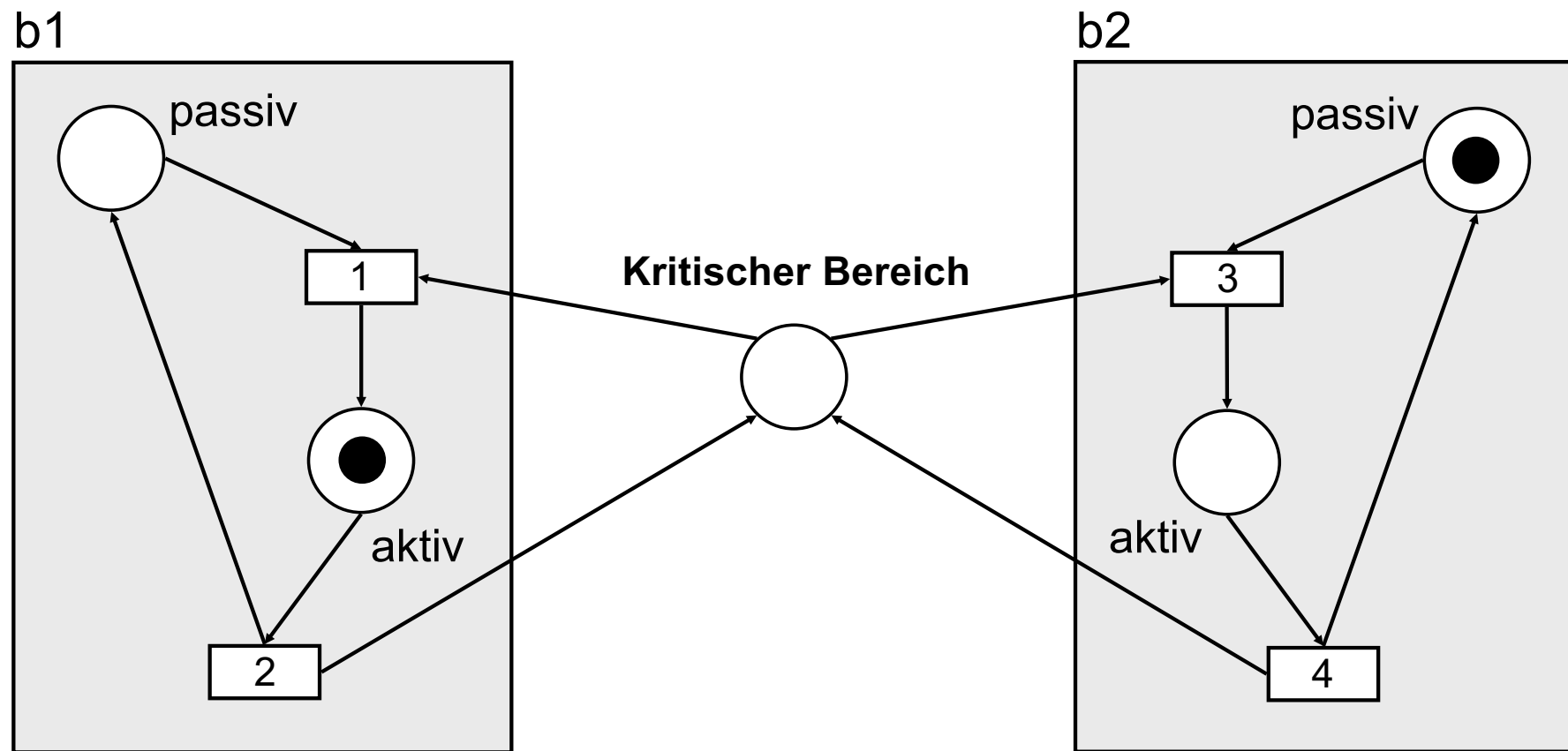
# Problem des gegenseitigen Ausschlusses

b1 und b2 sind Bereiche, die nicht gleichzeitig ablaufen sollen



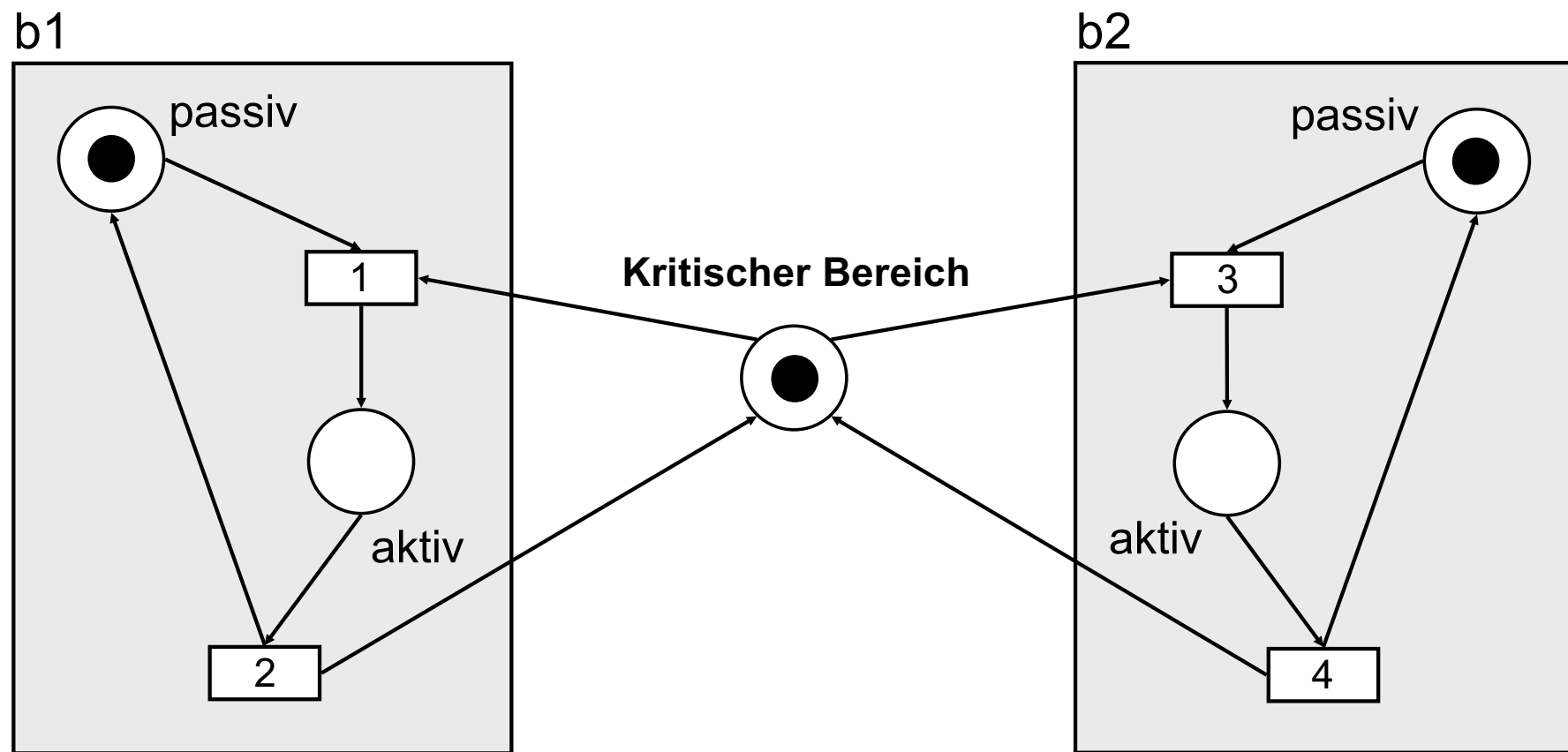
# Problem des gegenseitigen Ausschlusses

b1 und b2 sind Bereiche, die nicht gleichzeitig ablaufen sollen



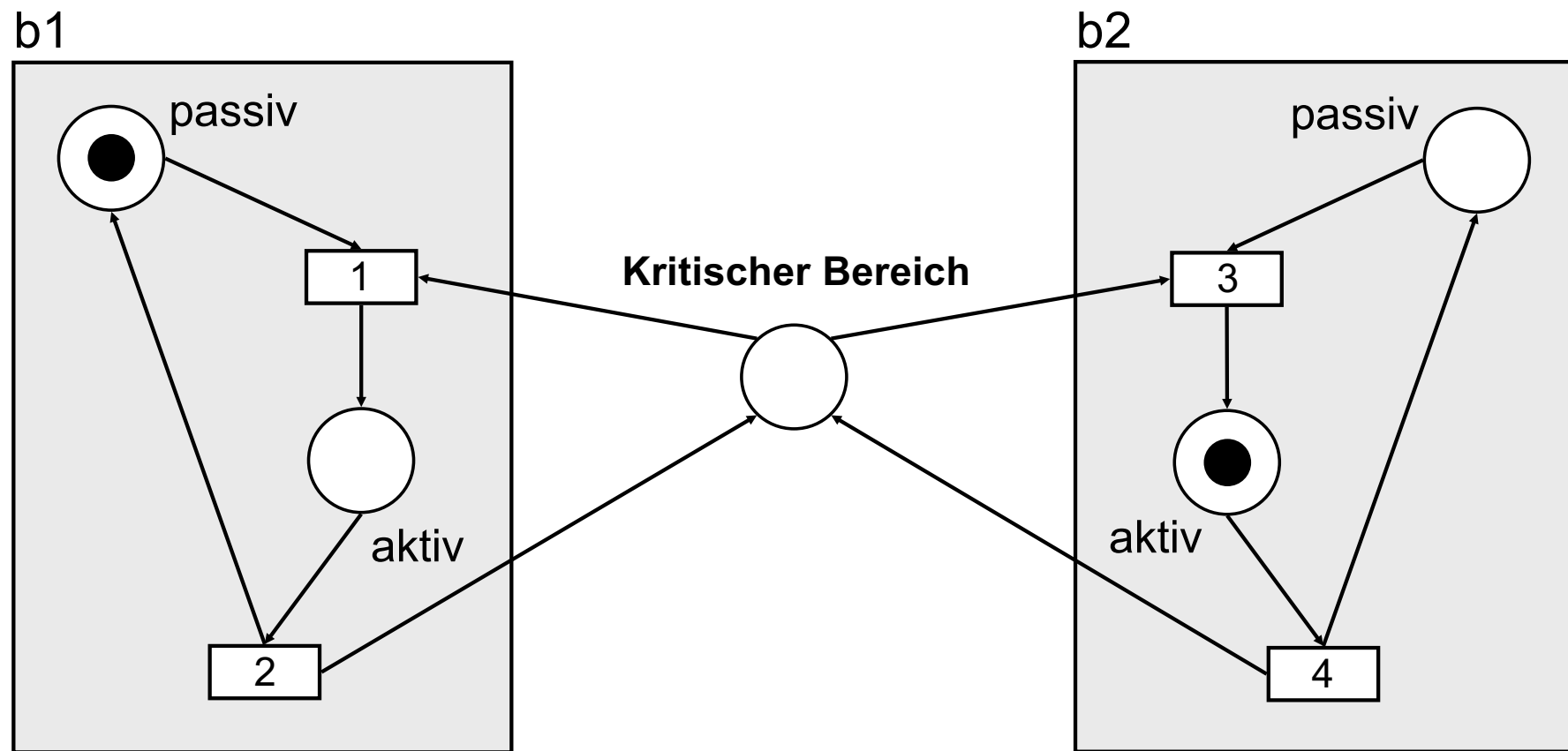
# Problem des gegenseitigen Ausschlusses

b1 und b2 sind Bereiche, die nicht gleichzeitig ablaufen sollen



# Problem des gegenseitigen Ausschlusses

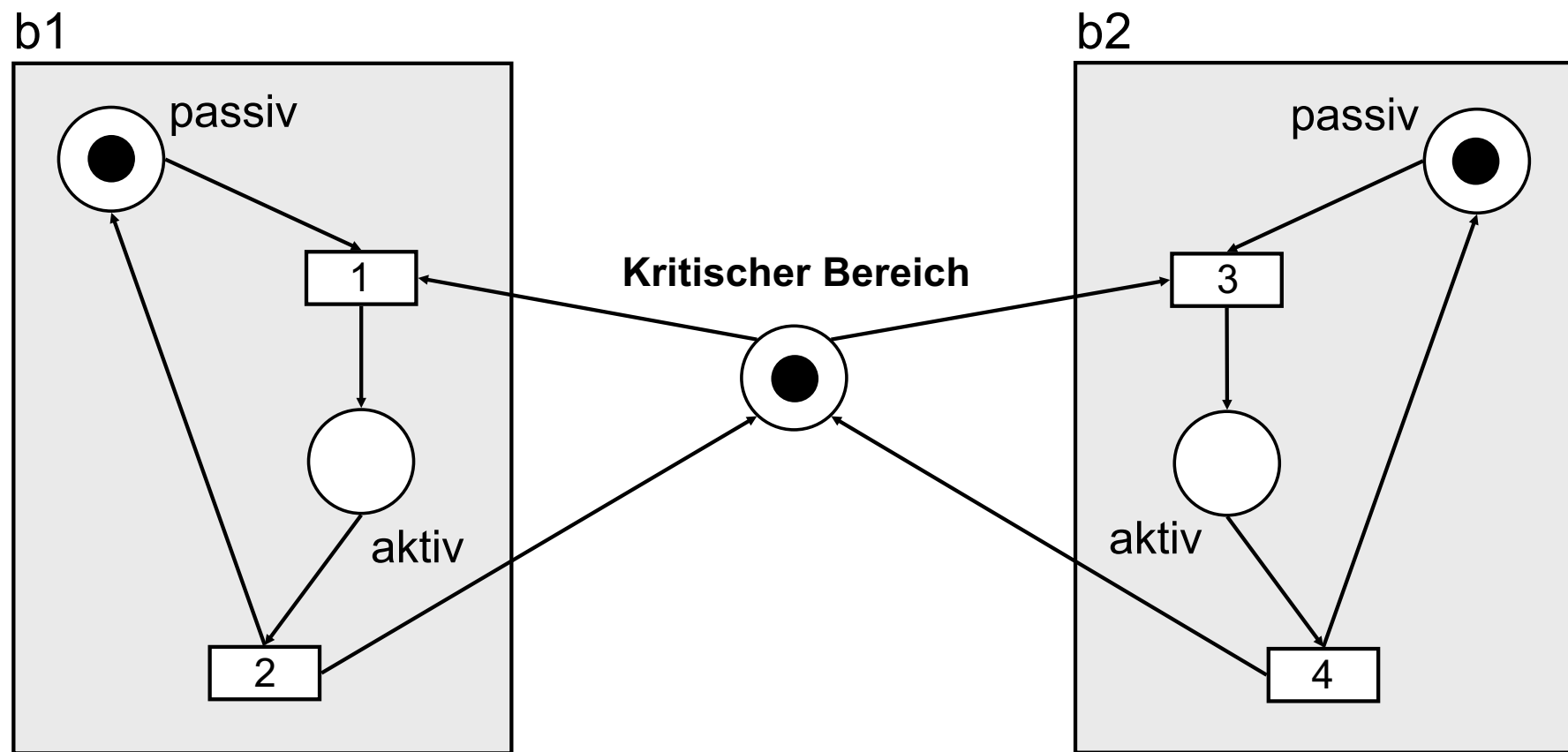
b1 und b2 sind Bereiche, die nicht gleichzeitig ablaufen sollen





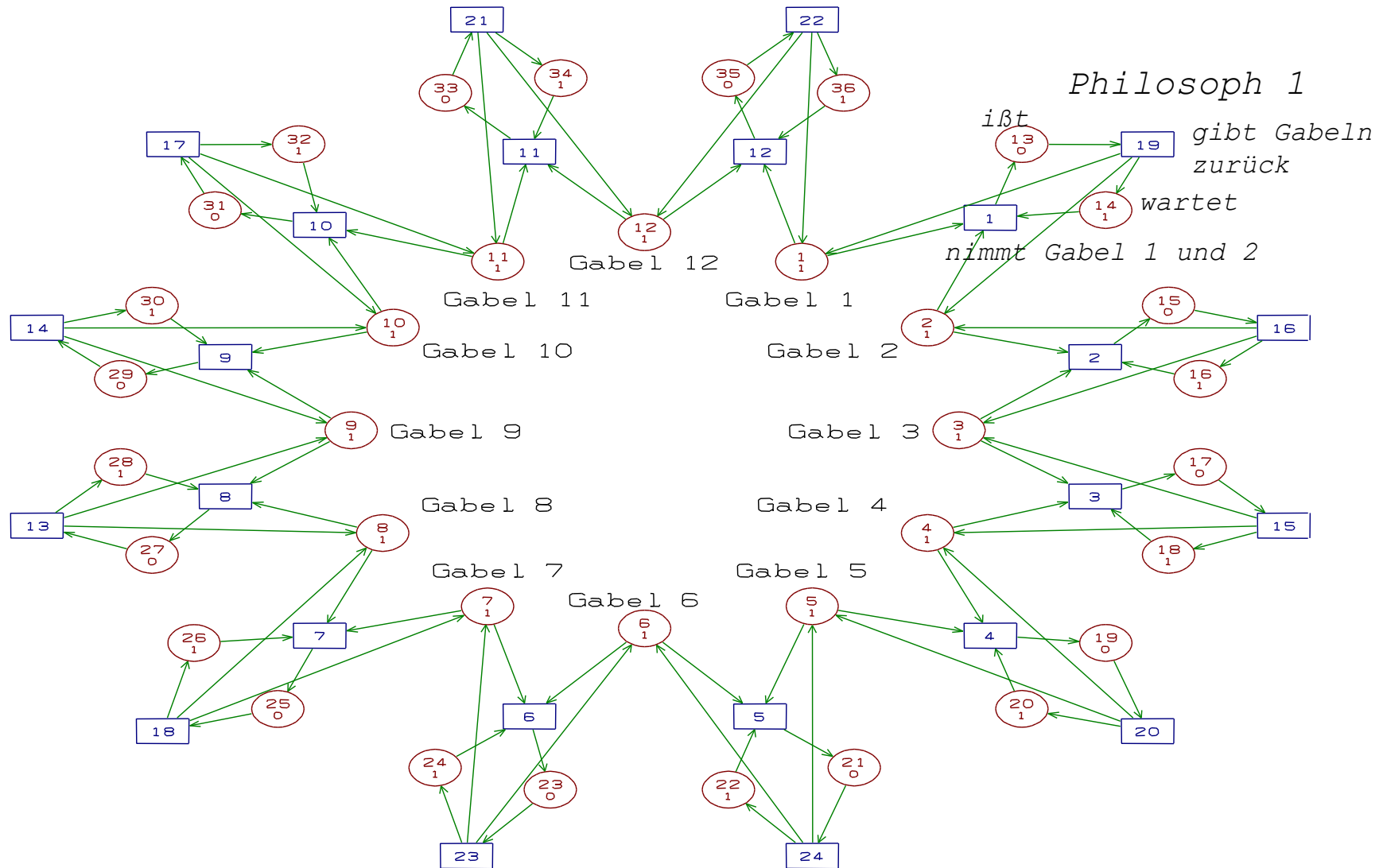
# Problem des gegenseitigen Ausschlusses

b1 und b2 sind Bereiche, die nicht gleichzeitig ablaufen sollen



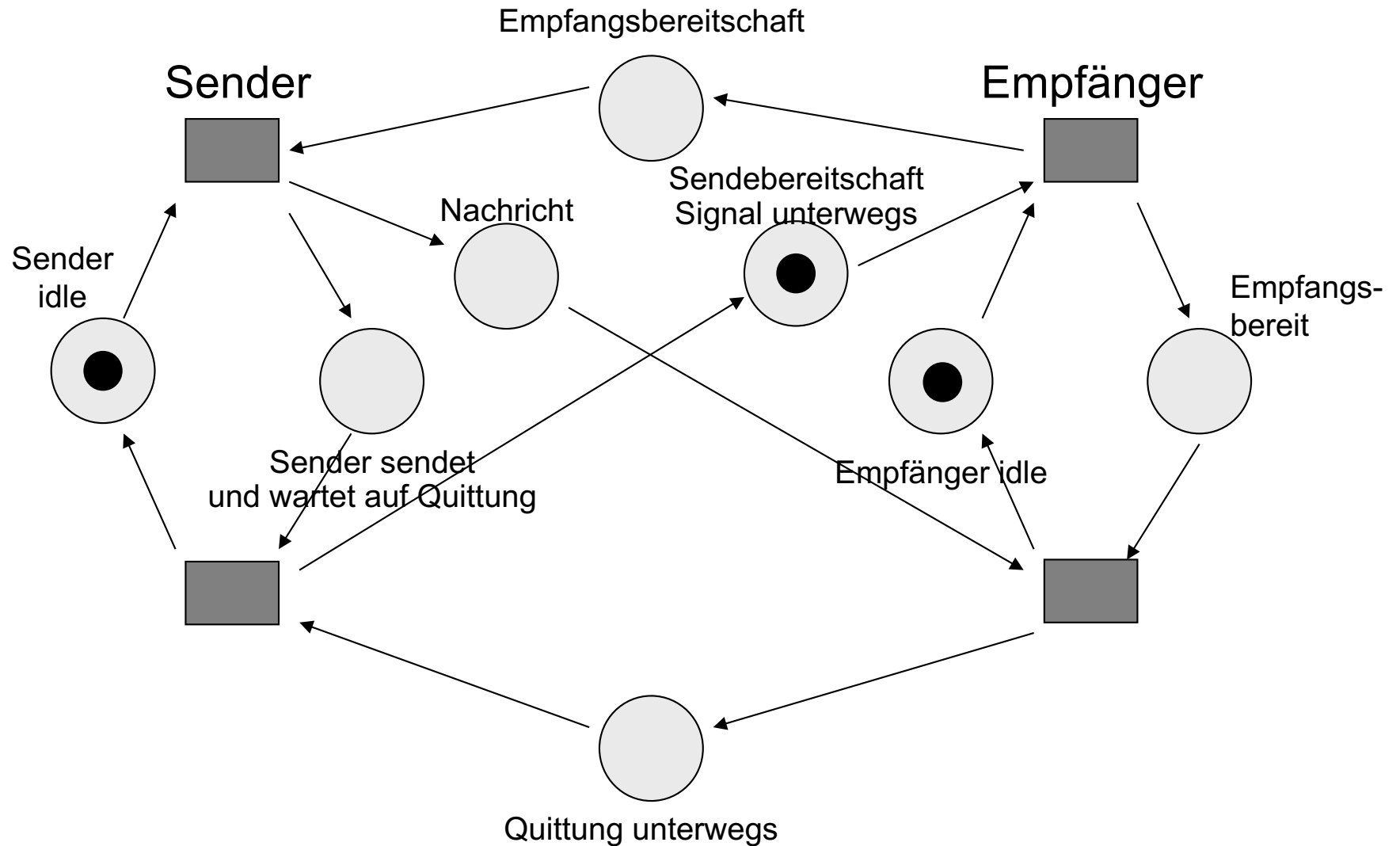
# Petrinetz – Beispiele

## Das Problem der Philosophen



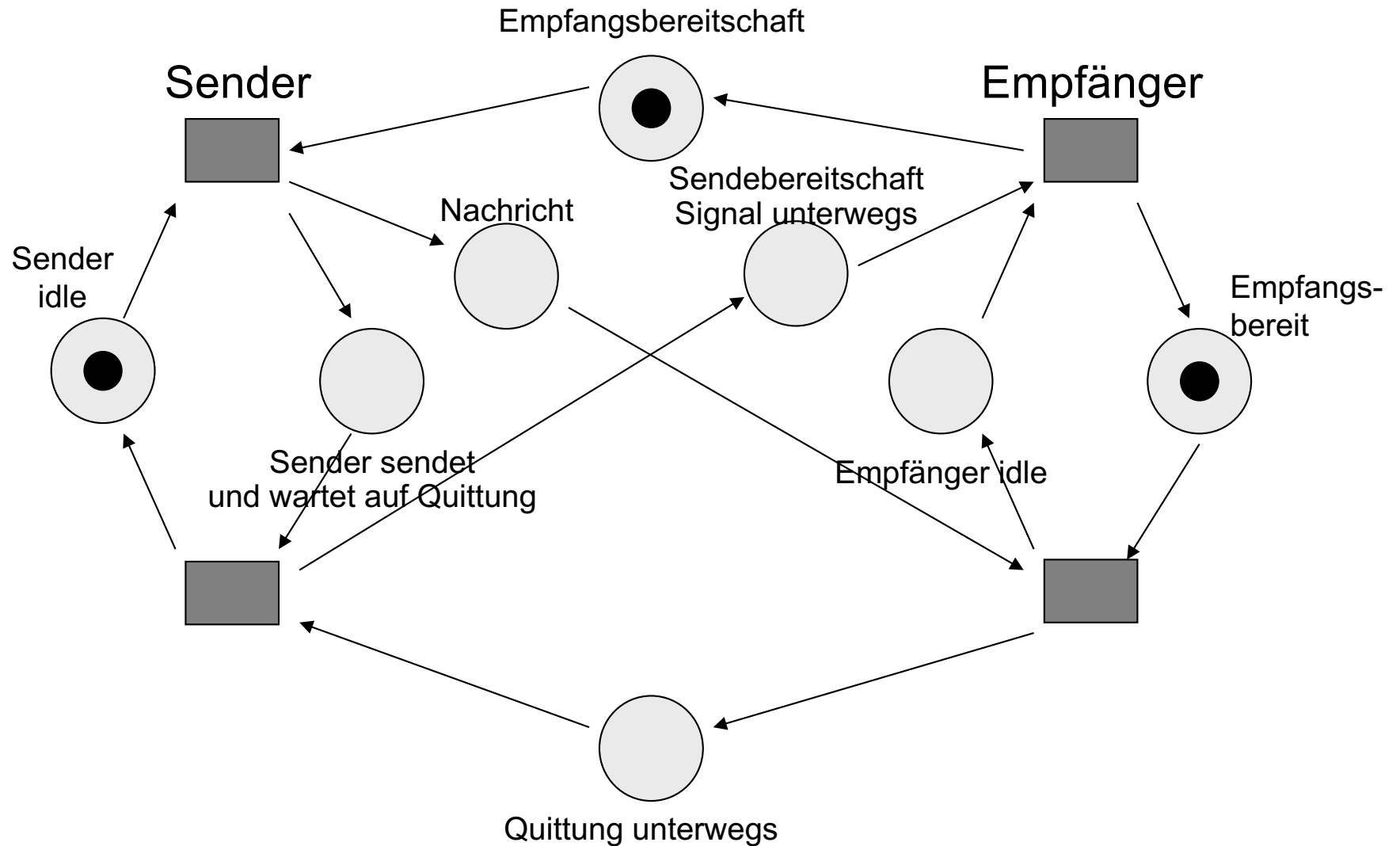
# Petrinetz – Beispiele

## Sender - Empfänger



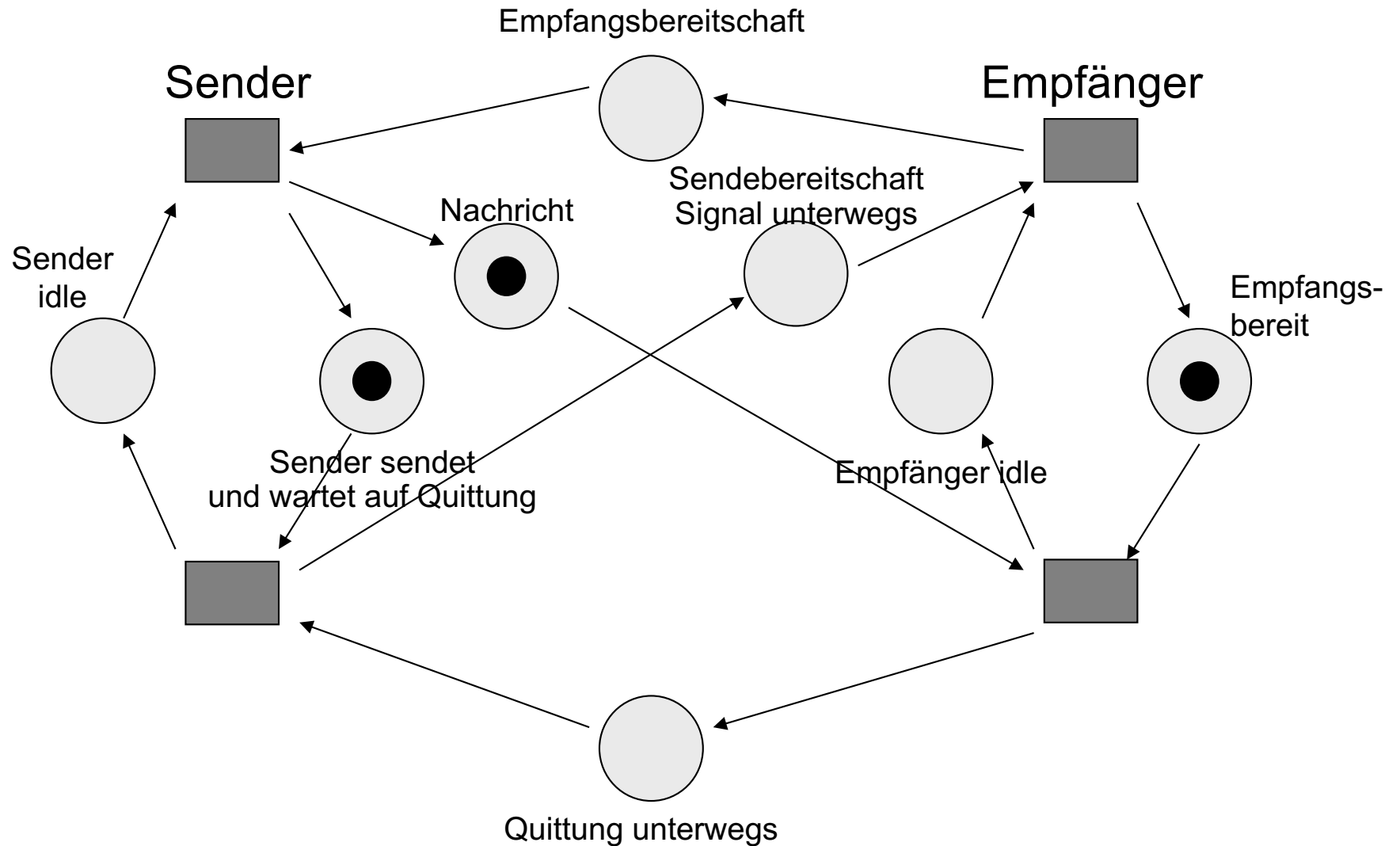
# Petrinetz – Beispiele

## Sender - Empfänger



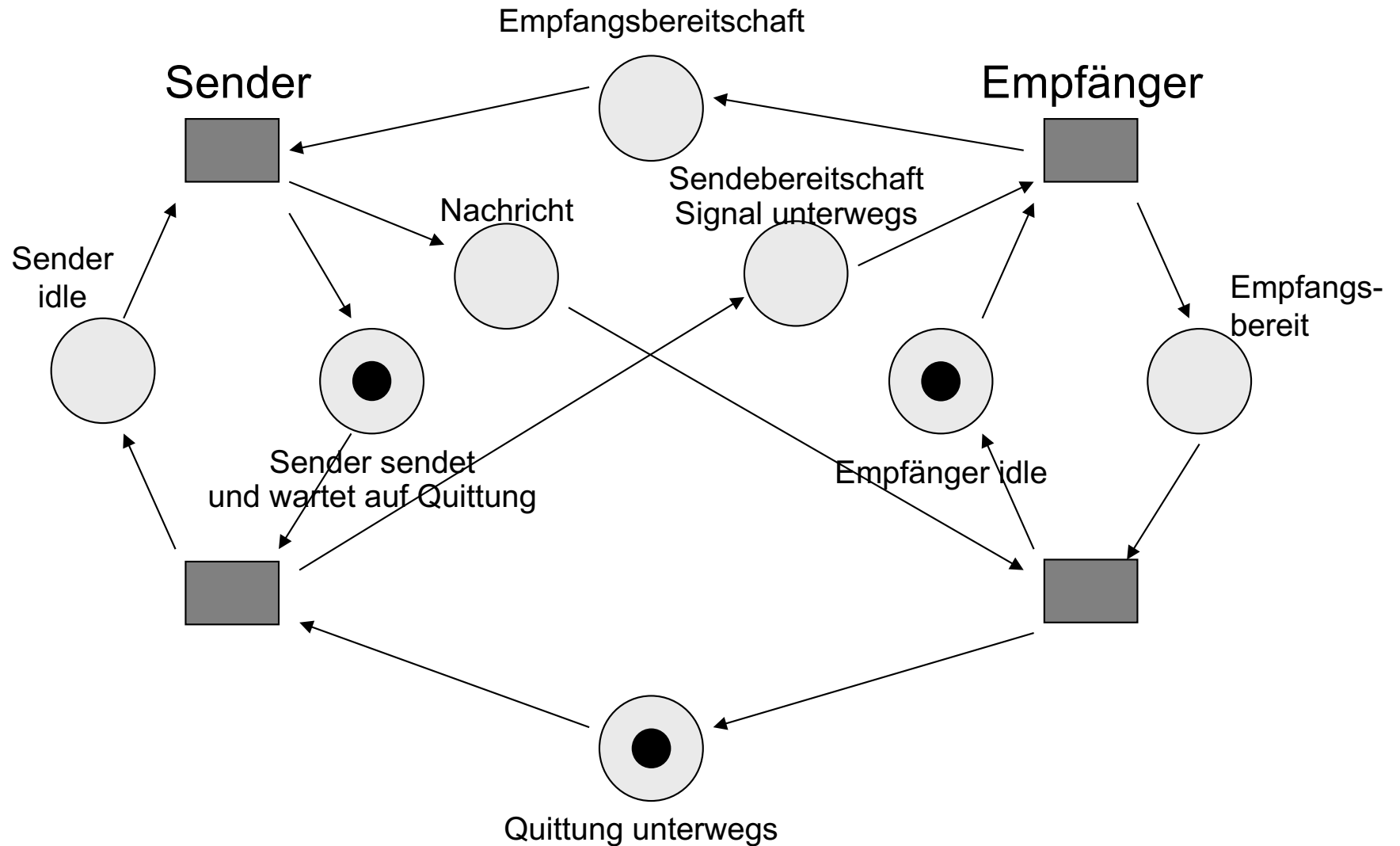
# Petrinetz – Beispiele

## Sender - Empfänger



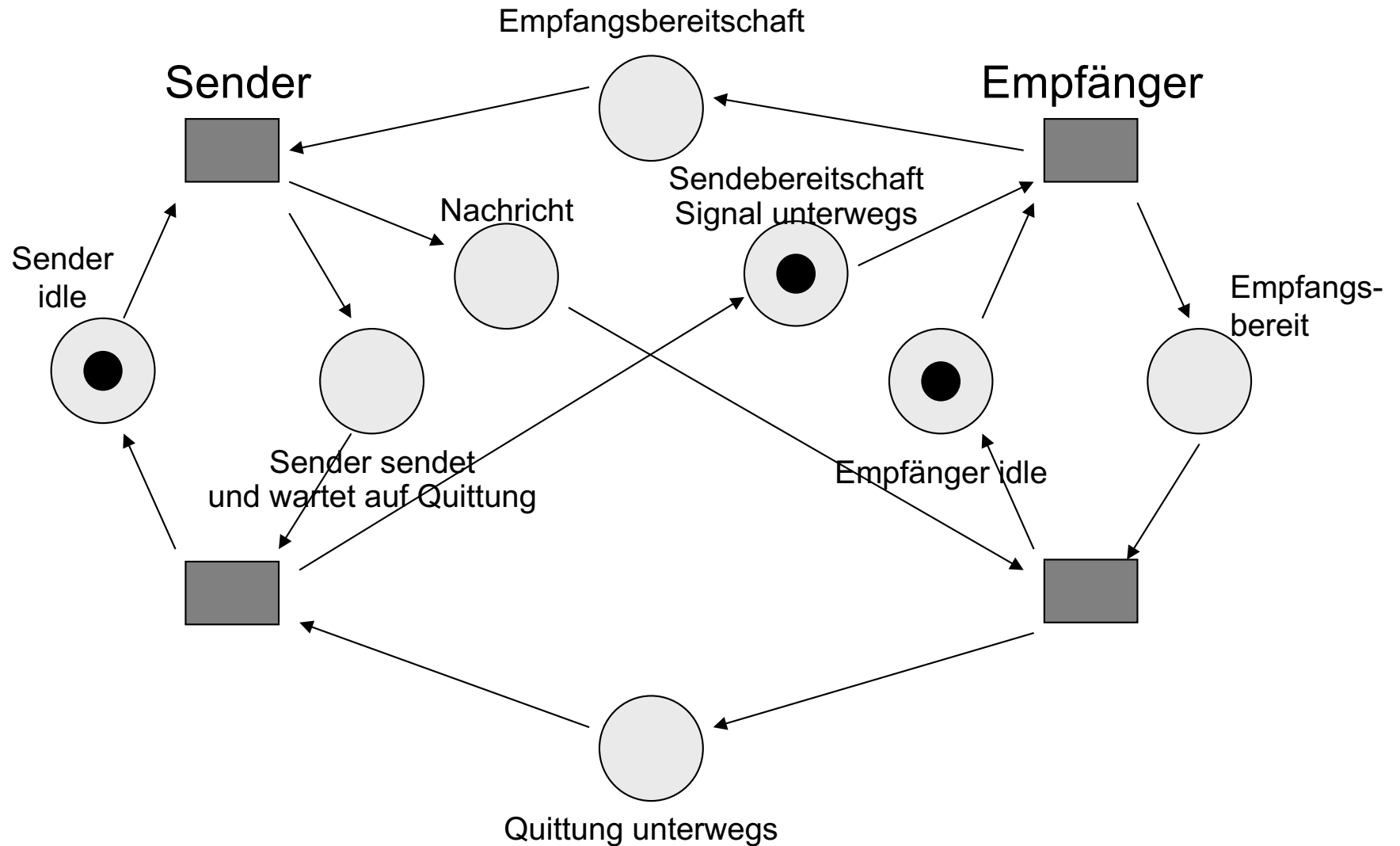
# Petrinetz – Beispiele

## Sender - Empfänger

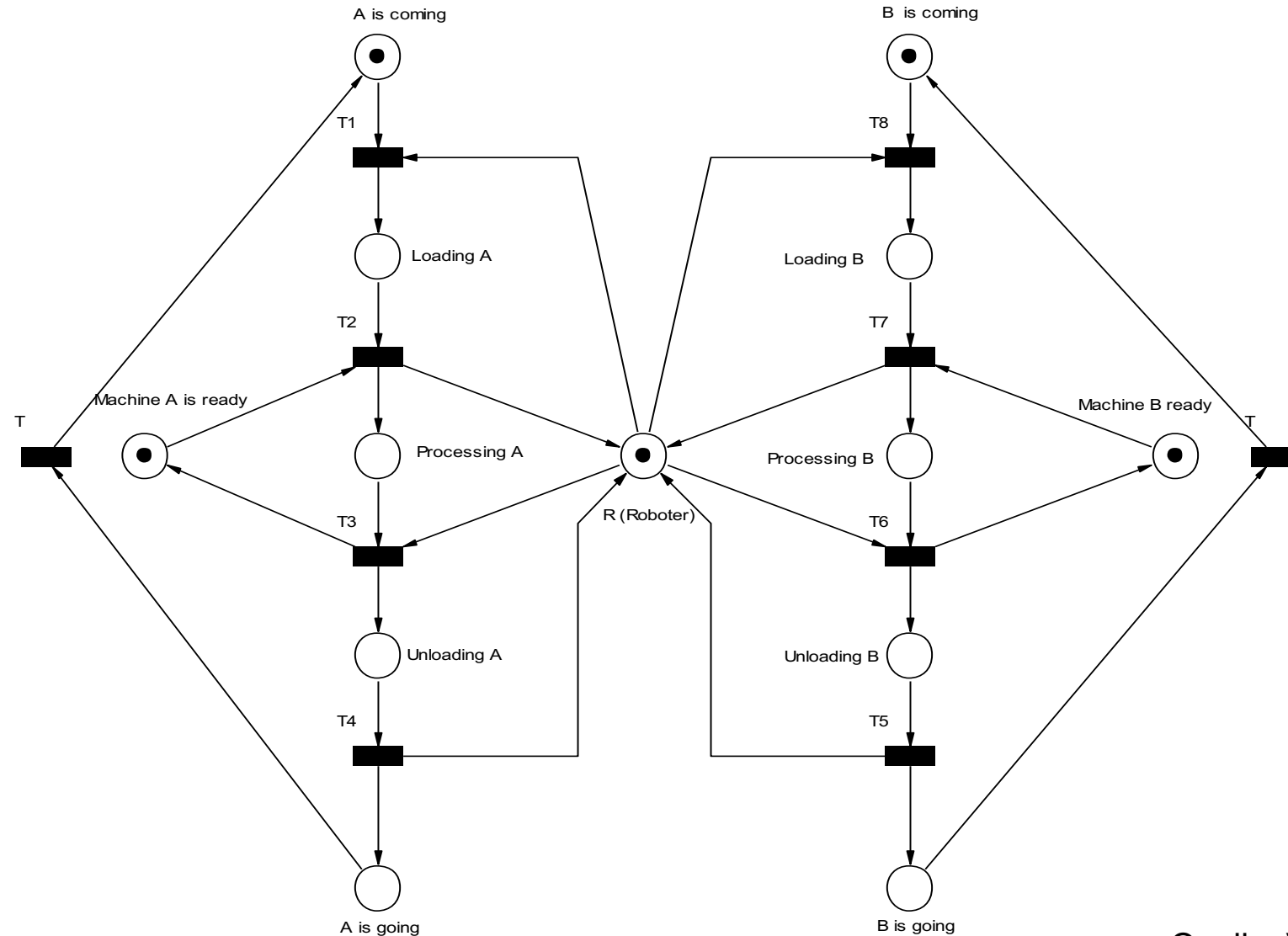


# Petrinetz – Beispiele

## Sender - Empfänger



# Beispiel: Roboterzelle

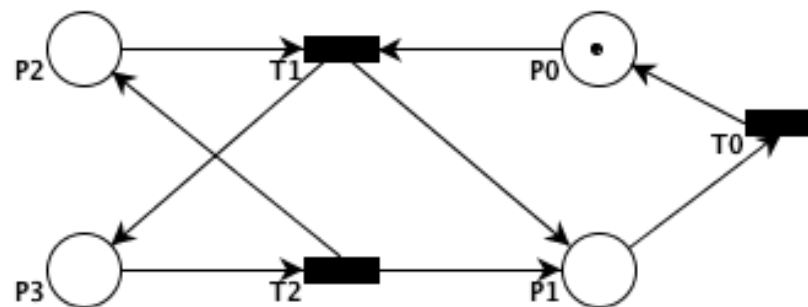
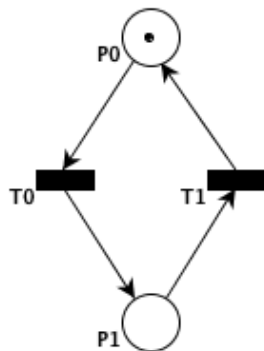


Quelle: Visual Object Net ++



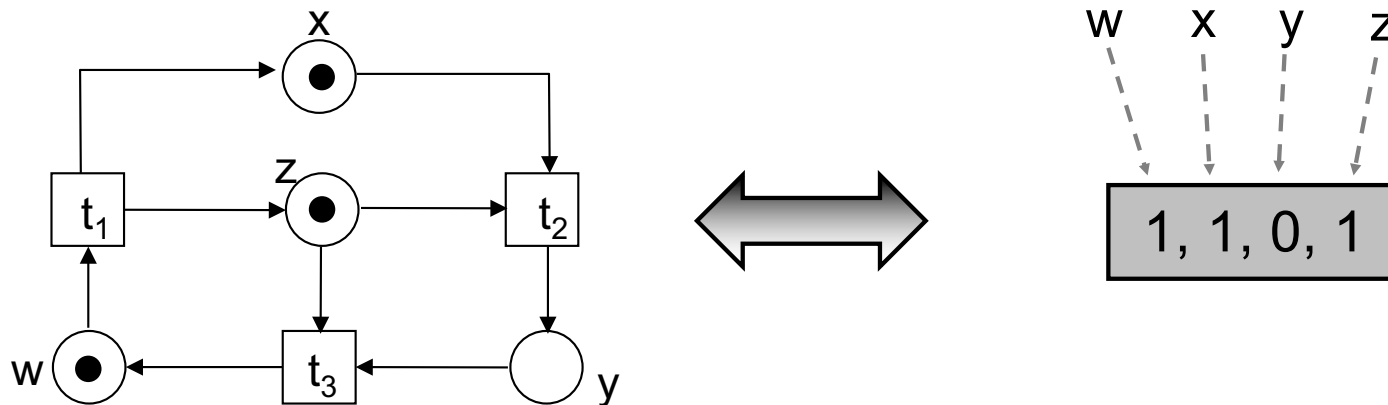
# Eigenschaften von Petrinetzen

- Eine Transition  $t$  ist
  - lebendig, falls es eine feuerbare Folge von Transitionen gibt, in der  $t$  vorkommt
  - tot, falls sie unter keiner Folgemarkierung aktiviert ist
- Ein Petrinetz ist
  - lebendig, wenn sie unter keiner Folgemarkierung tot ist
  - in einer Verklemmung (deadlock) wenn keine Transition schaltbar ist



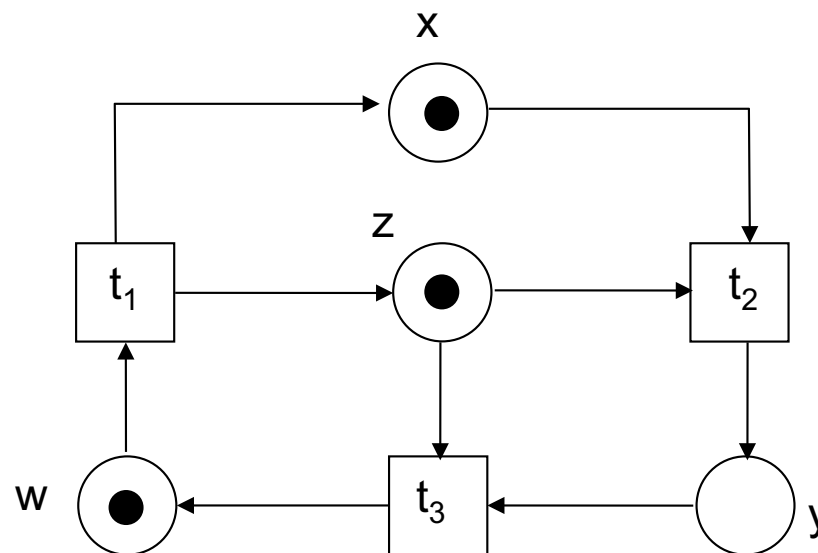
# Darstellung von Zustände von Petrinetzen

- Zur kürzeren Darstellung von Erreichbarkeitsgraphen kann der Zustand eines Petrinetzes durch die Anzahl an Marken pro Stelle beschrieben werden
- Beispiel



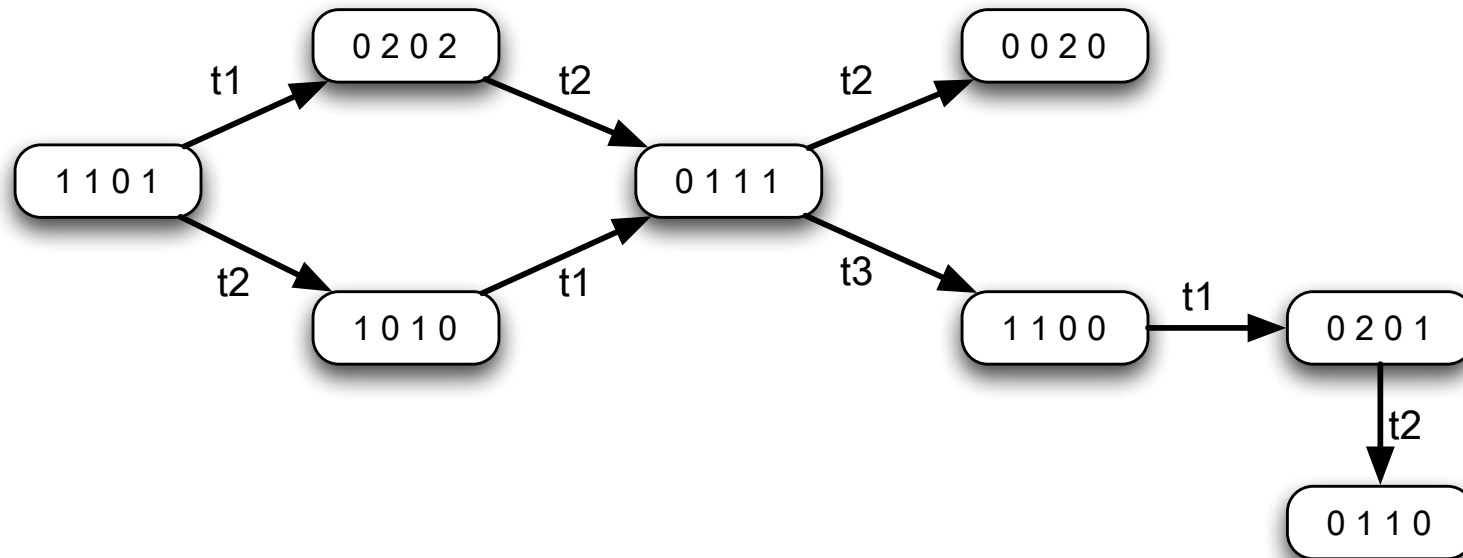
# Erreichbarkeitsgraph

- Graph, der alle Zustände beschreibt, die ein Petrinetz erreichen kann
  - Die Knoten des Graphen entsprechen Zustände des Petrinetzes
  - Die Kanten des Graphen entsprechen Transitionen des Petrinetzes
- Beispiel: Erreichbarkeitsgraph des folgenden Petrinetzes?



# Erreichbarkeitsgraph

- Lösung

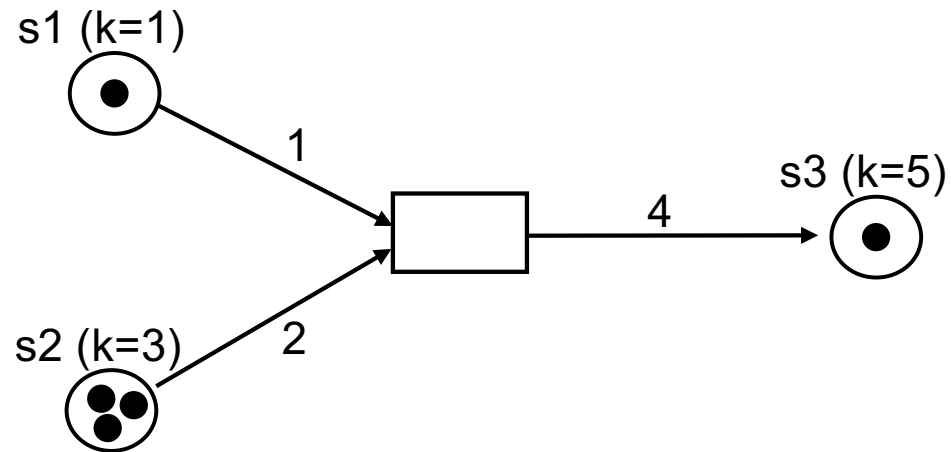


- Mögliche Schaltfolgen

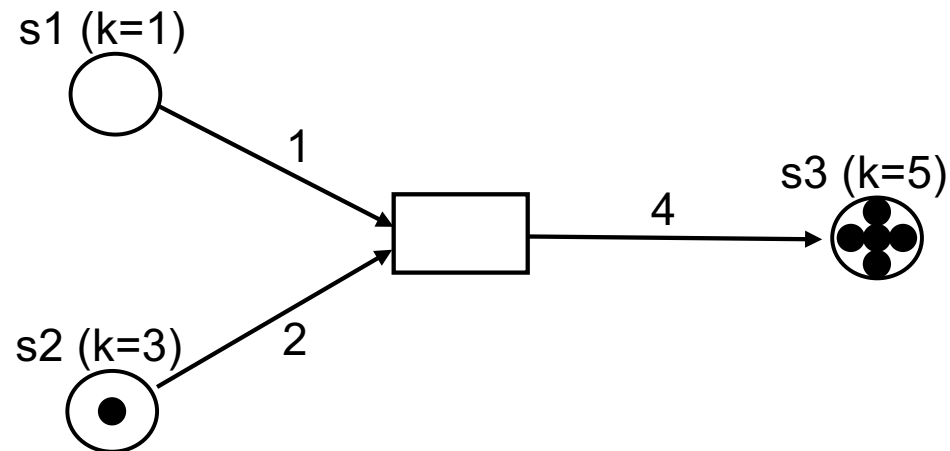
- t1 t2 t2
- t2 t1 t2
- t1 t2 t3 t1 t2
- t2 t1 t3 t1 t2

# Erweitertes Petrinetz

## Beispiel



 Folgemarkierung



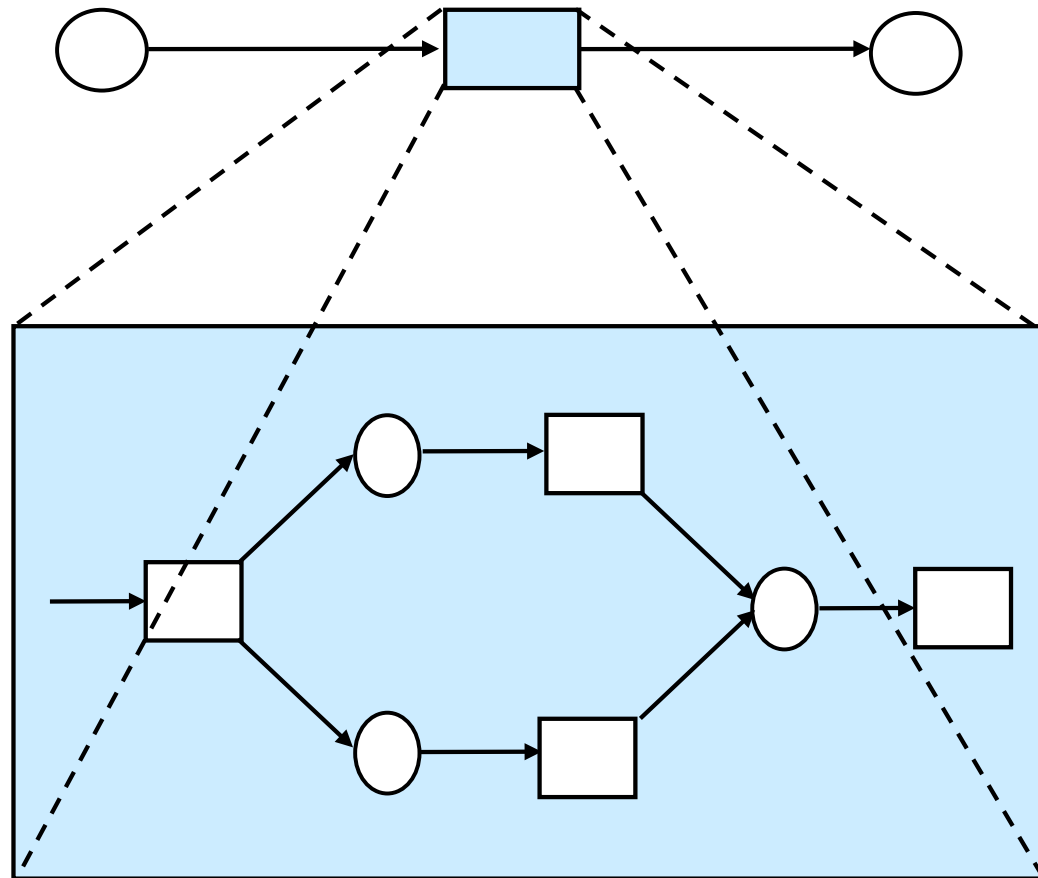
# Erweitertes Petrinetz

## Schaltregeln

- Eine Transition ist aktiviert, wenn alle Stellen, von denen eine Kante zu dieser hinführt, ausreichend markiert sind.
- Eine Transition ist nur dann aktiviert (schaltfähig, - bereit), wenn der Schaltvorgang der Transition zu einer zulässigen Folgemarkierung des Petri- Netzes führt.
- Eine zulässige Folgemarkierung liegt vor, wenn
  - (1) keine Stelle weniger als null Marken enthält und
  - (2) die Kapazität keiner Stelle überschritten wird
- Fordert man nur Bedingung (1), so spricht man von der schwachen Schaltregel
- Die starke Schaltregel setzt die Erfüllung beider Bedingungen voraus

# Hierarchische Petrinetze

- Strukturierung von Petrinetzen
  - Verfeinerung durch detaillierte Unternetze



# Beispielaufgabe

Entwerfen Sie ein Petri-Netz, das das folgende System beschreibt:

Drei Fabrikarbeiter A1 bis A3 arbeiten parallel: A3 schraubt pro Arbeitsgang ein Blech zusammen und benötigt dafür je drei Schrauben. A2 stellt pro Arbeitsgang fünf Schrauben her und benötigt dafür zwei Rohlinge. A1 stellt pro Arbeitsgang vier Rohlinge her. Der Vorrat an Rohlingen kann maximal sieben Teile, der an Schrauben maximal zwölf Teile enthalten.

Alle drei benötigen zum Arbeiten ein Werkzeug, das nur einmal vorhanden ist und nur von jeweils einem Arbeiter gleichzeitig benutzt werden kann.

Analysieren Sie das Modell. Kann es zu Deadlocks kommen?