

Prof. Dr.-Ing. Dr. h. c. J. Becker

becker@kit.edu

Karlsruher Institut für Technologie (KIT)

Institut für Technik der Informationsverarbeitung (ITIV)

Digitaltechnik

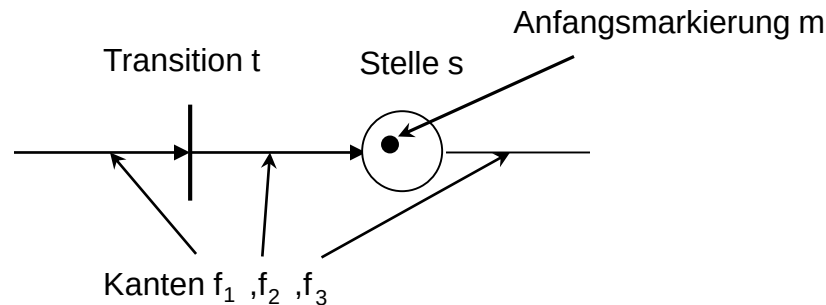
Petrinetze

Definition:

Ein **Petrinetz** ist ein **Tupel** $P = (S, T, F, M)$, wobei

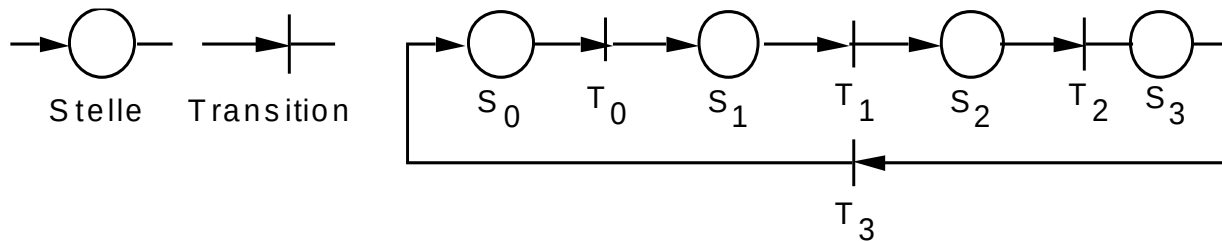
- S Menge der **Stellen**,
- T Menge der **Transitionen**,
- $F \subseteq S \times T \cup T \times S$ Kantenmenge und
- $M: S \rightarrow \mathbb{N}$ Anfangsmarkierung heißt

Die Menge $E = S \cup T$ ist die **Eckenmenge** des Graphen



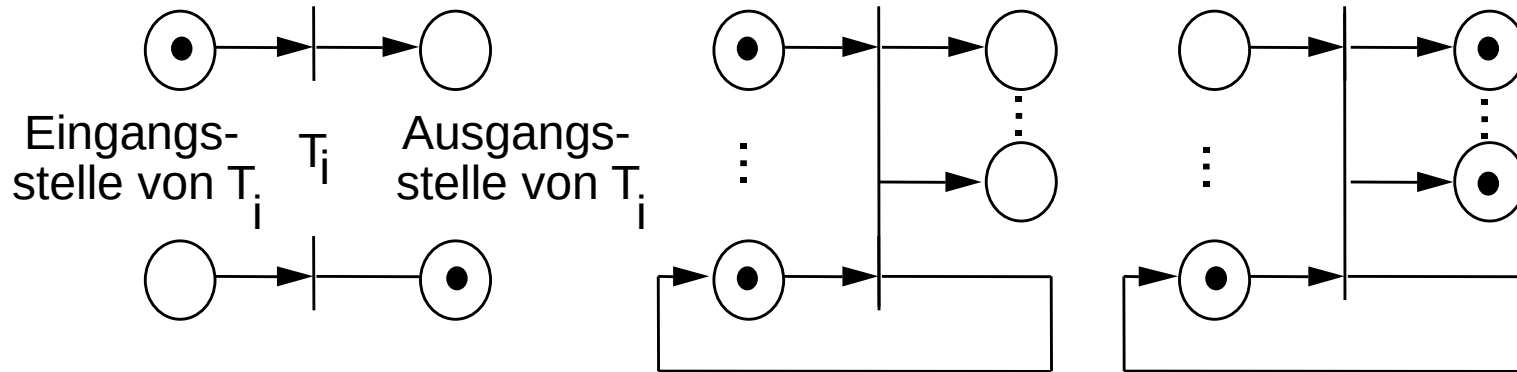
wobei $s \in S$; $t \in T$; $f_1, f_2, f_3 \in F$; $m \in M$ ist

Petrinetz: bipartiter, gerichteter Graph
Stellen (= Bedingungen, Situationen)
Transitionen (= Aktionen)



Beispiel für ein **Petrinetz** (siehe oben)

S_0	für Zündschlüssel bereit	T_0	für Zündung einschalten und Motor starten
S_1	für Zündung ein, Motor läuft	T_1	für Zündung ausschalten
S_2	für Motor steht	T_2	für Schlüssel abziehen
S_3	für Schlüssel abgezogen	T_3	für Schlüssel in Zündschloss stecken



Einfache Schaltregeln:

eine Transition ist

schaltbereit: wenn **alle Eingangsstellen** eine **Markierung** tragen,
alle **Ausgangsstellen leer** sind
(sofern diese nicht gleichzeitig Eingangsstellen sind)

schaltet: dabei **zieht** sie die Marken aller **Eingangsstellen ab**, so daß
diese leer werden, und **markiert alle Ausgangsstellen**

- Die **Stellen** des **Graphen** repräsentieren **Zustände** bzw. **Bedingungen** des modellierten Systems; Darstellung im Graph durch Kreise
- Die **Knoten** in **T** bezeichnen **Transitionen** oder Aktivitäten; Darstellung im Graph als Querstriche oder Kästchen
- Die **Kantenmenge F** repräsentiert kausale oder zeitliche Abhängigkeiten, wobei
 - eine **Kante $s \rightarrow t$** bedeutet, dass die **Bedingung s** erfüllt sein muss, damit die **Transition t schalten** (oder feuern) kann
 - **s heißt** in diesem Fall **Vorstelle** von **t**
 - eine **Kante $t \rightarrow s$** bedeutet, dass der **Zustand s erreicht** wird, wenn die **Transition t geschaltet** hat
 - **s heißt** dann **Nachstelle** von **t**
- Besitzt eine **Transition t mehrere Vorstellen**, müssen **alle Bedingungen erfüllt** sein, damit t schalten kann
- Besitzt eine **Transition t mehrere Nachstellen**, so werden durch das Schalten von t **alle Nachstellen gleichzeitig** erreicht

Petrinetze - Motivation (1)

Verwendungszwecke

- Simulation
- Planung
- Steuerung

Anwendungsgebiete

- Produktion
- Bürokommunikation (speziell Workflowsysteme)

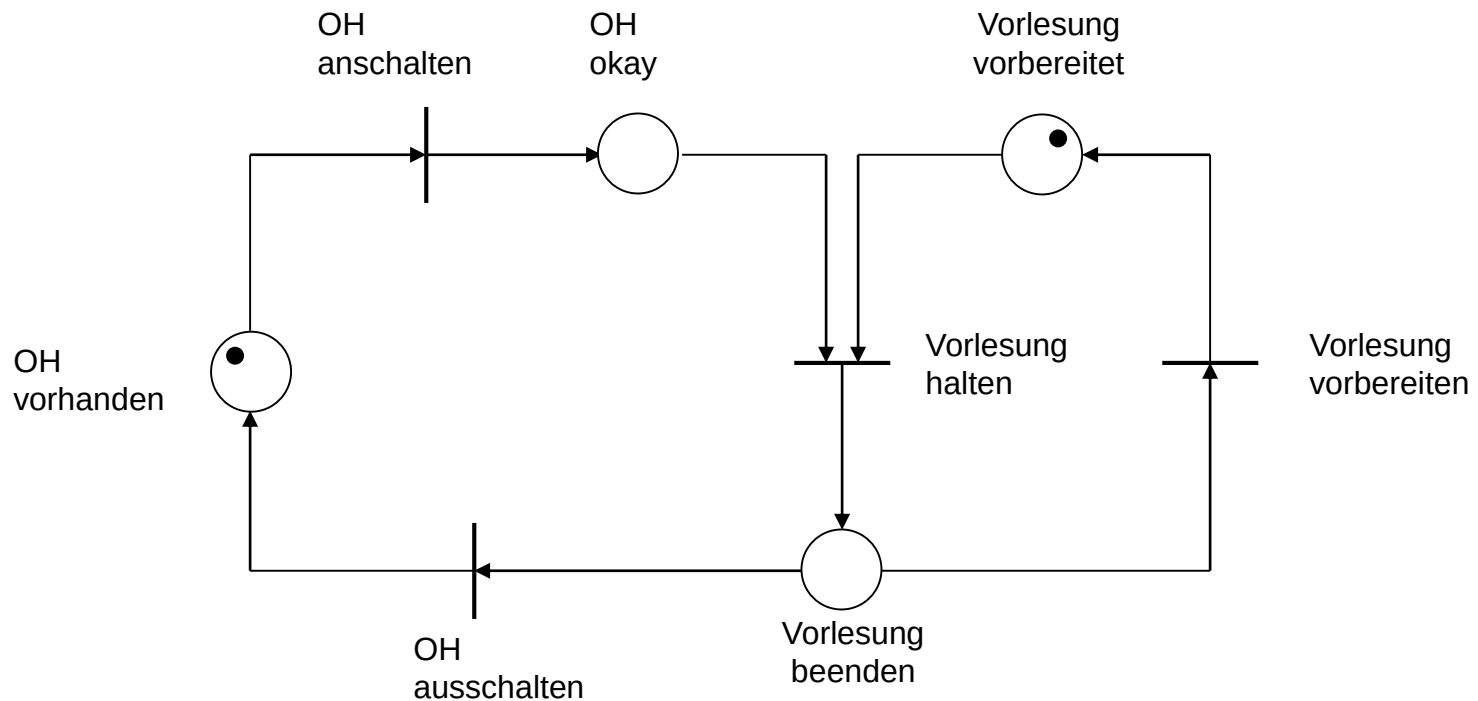
Klassische Anwendungen in der Kerninformatik

- Modellierung von Betriebssystemen
- Modellierung von gegenseitigem Ausschluss
- **Modellierung von Nebenläufigkeit**

Petrinetze - Motivation (2)

Petrinetze dienen zum **Modellieren von Systemen**, in denen **kausale Abhängigkeiten** eine vorrangige Bedeutung besitzen

Einführendes Beispiel:

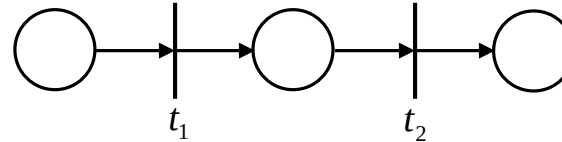


- Im Jahre 1962 von **C.A. Petri** im Rahmen seiner Dissertation vorgestellt
- Netztheorie zur **Modellierung** komplexer Vorgänge, die es erlaubt **unabhängige parallele Prozesse** darzustellen
- Die Beschränkung streng sequentieller Abläufe bisheriger Darstellungen entfällt
- Die Idee fand zunächst keinen praktischen Nutzen, jedoch änderte sich die Akzeptanz im Laufe der Zeit deutlich

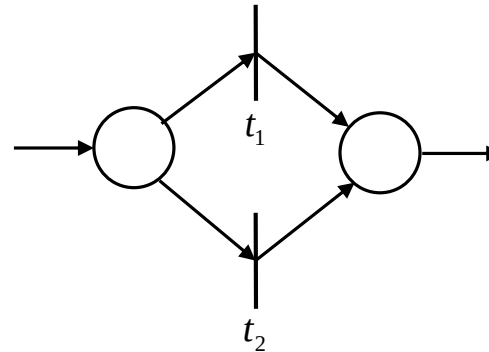
Petrinetze – Modellierung (1)

Was kann mit Petrinetzen modelliert werden?

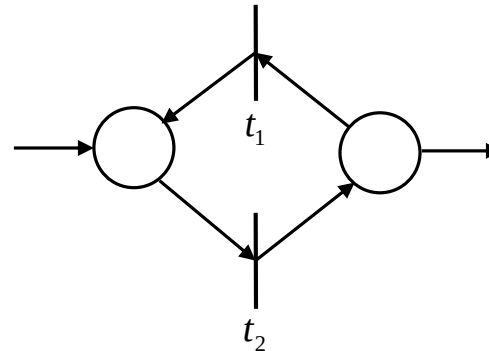
1. Sequentielle Abläufe:



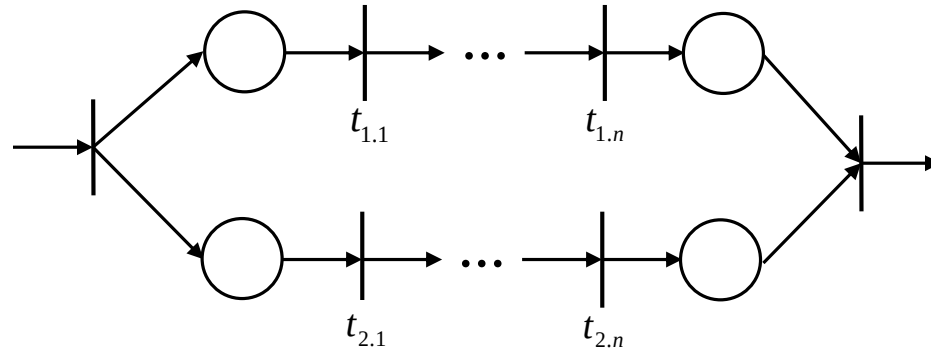
2. Alternative:



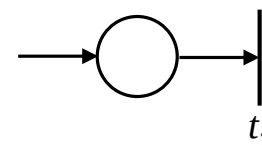
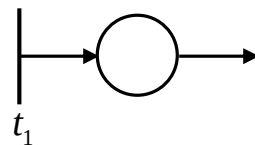
3. Wiederholung:



4. Nebenläufigkeit:



5. Eingabe von / Ausgabe zur Umwelt:



Weitere Notationen:

Vorbereich eines Netzelements x

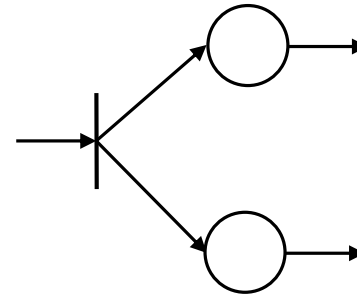
$$\bullet x = \{ y \mid (y, x) \in F \}$$

Nachbereich eines Netzelements x

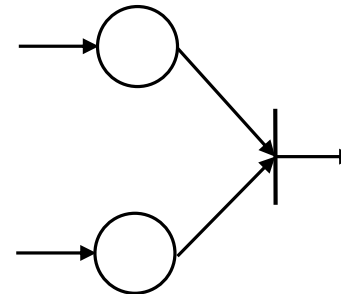
$$x\bullet = \{ y \mid (x, y) \in F \}$$

Verzweigungen: ein Knoten heißt

vorwärtsverzweigt, falls $|x\bullet| > 1$



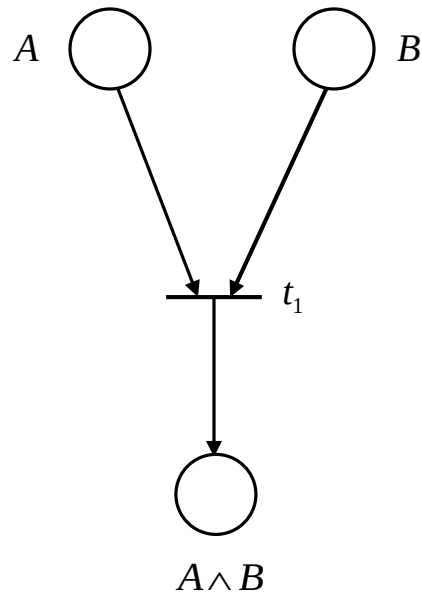
rückwärtsverzweigt, falls $|\bullet x| > 1$



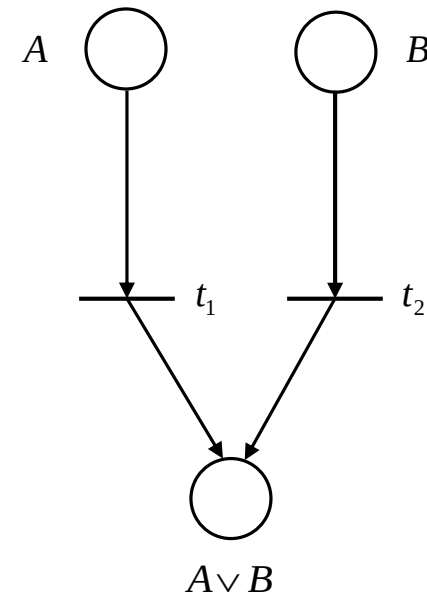
Petrinetze – Modellierung (4)

Logische Aussagen:

Logisches UND



Logisches ODER

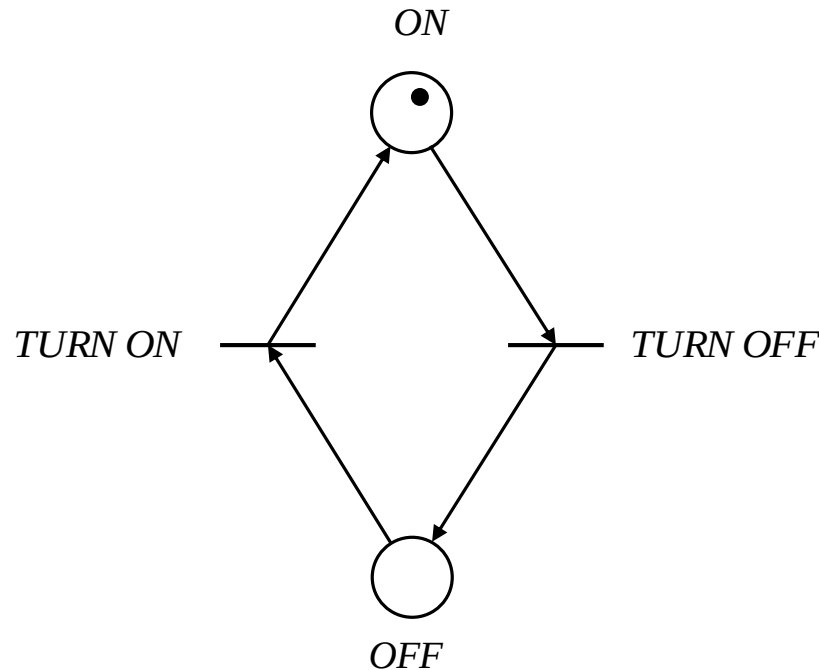


Petrinetze – Modellierung (5)

Schalter:

Jeder **Zustand** ist **repräsentiert** durch einen **Platz**

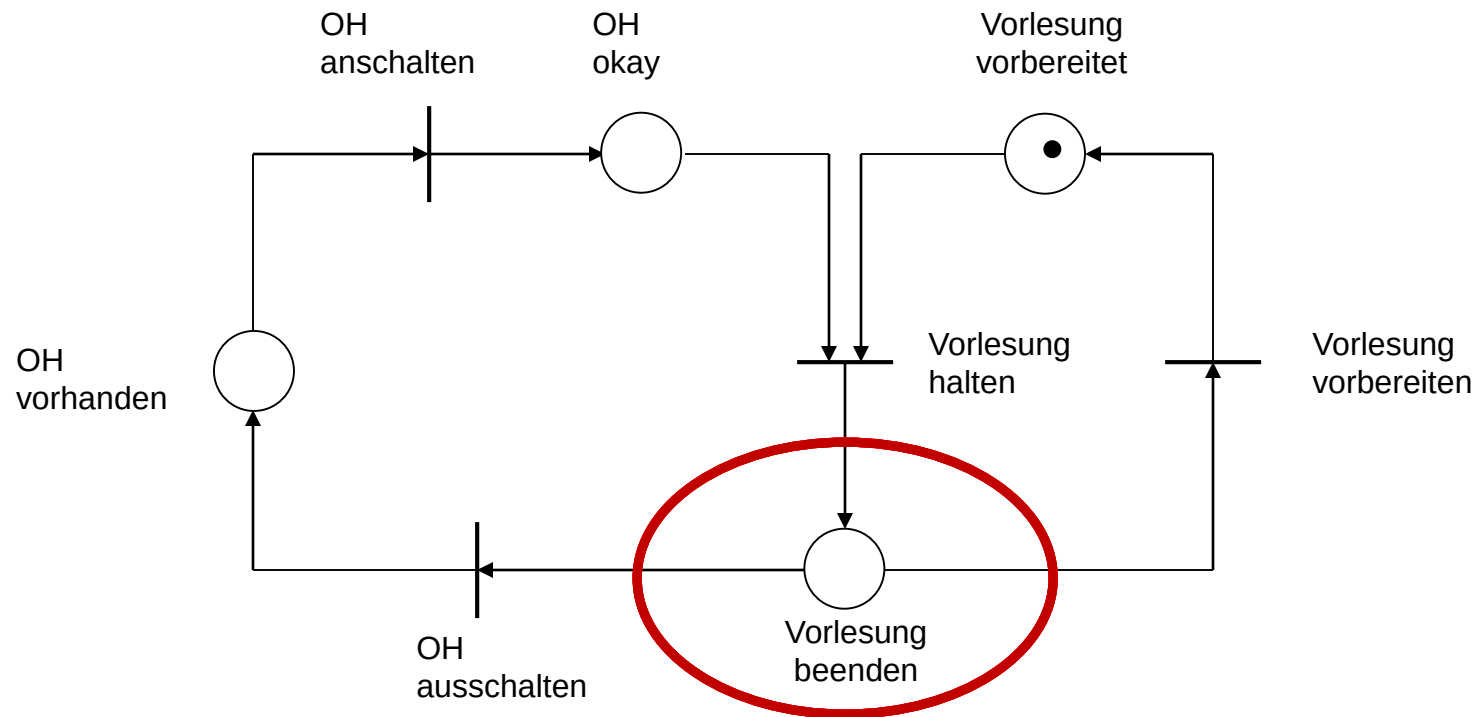
→ wenn ein **Token** in diesem Platz ist, bedeutet das,
dass sich das **Modell** in **diesem Zustand** befindet



Petrinetze – Verklemmungen

Nochmals das einführende Beispiel:

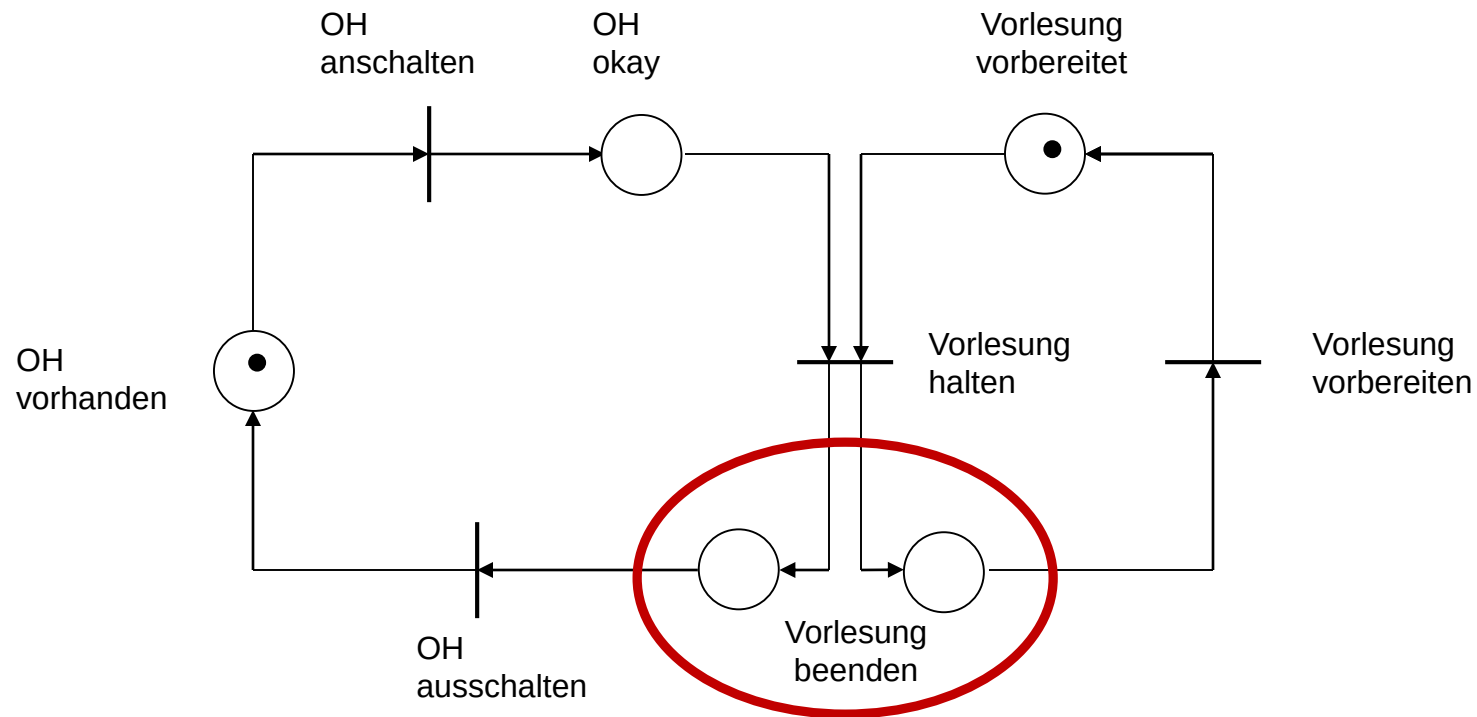
Hier kommt es **nach** dem **ersten Durchlauf** zu einem **Konflikt** und **anschließend** zu einer **Verklemmung**



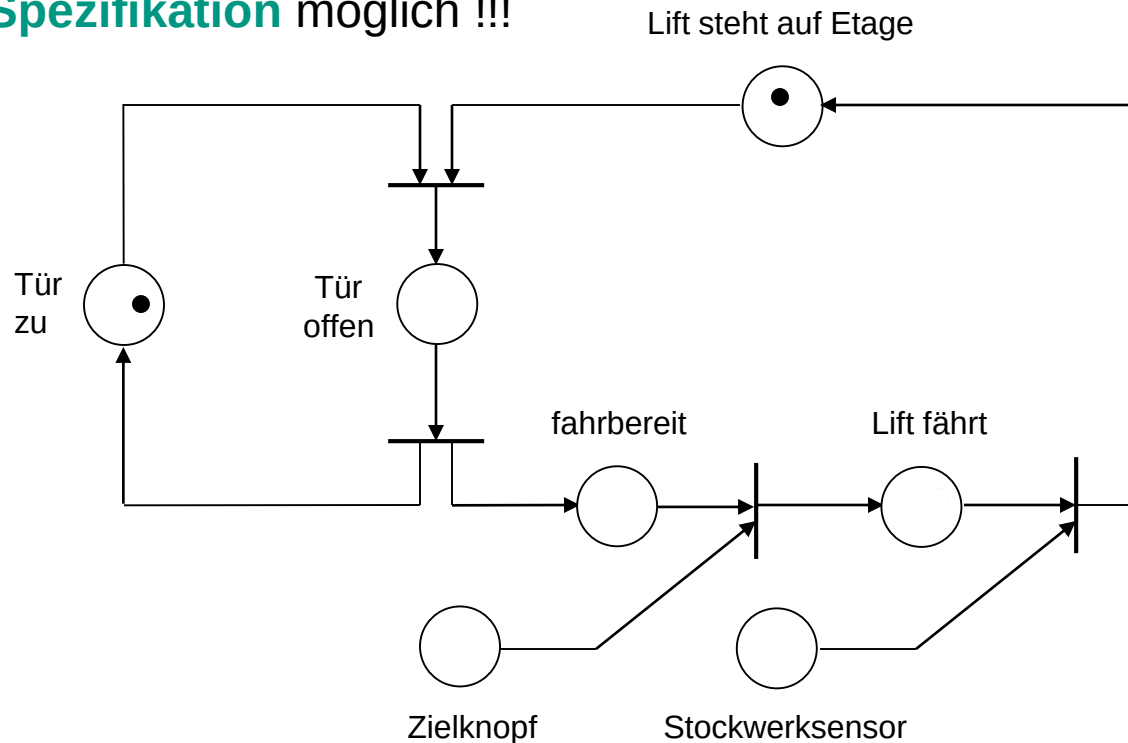
Nochmals das einführende Beispiel:

Hier kommt es **nach** dem **ersten Durchlauf** zu einem **Konflikt** und **anschließend** zu einer **Verklemmung**

→ Eine **Änderung** des Netzes hilft hier weiter !!!



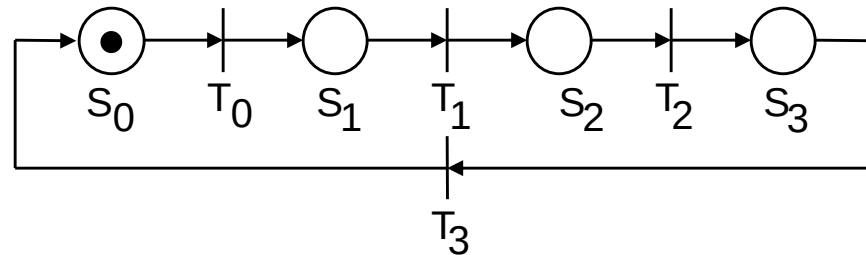
Beweis einer **sicherheitskritischen Eigenschaft** in einer **formalen Spezifikation** möglich !!!



Die **Sicherheitseigenschaft**, dass der Lift **nicht bei offener Tür** fahren darf, ist für dieses Modell **beweisbar**

Beispiel: **Motorsteuerung**

- rein sequentieller Ablauf ohne Nebenläufigkeiten
- ein deutlich kausaler Zusammenhang zwischen den Stellen S erkennbar



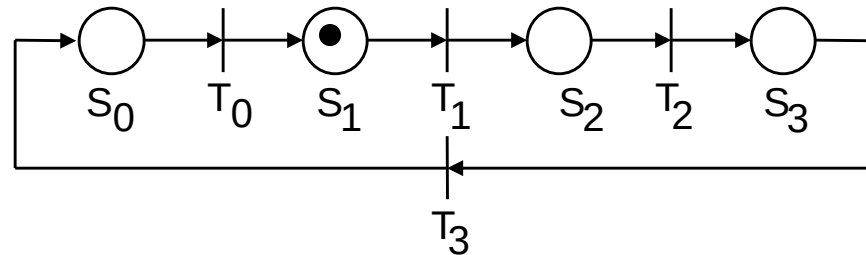
Zustände S_i (Stellen) und Transitionen T_j

S_0	für Zündschlüssel bereit	T_0	für Zündung einschalten und Motor starten
S_1	für Zündung ein, Motor läuft	T_1	für Zündung ausschalten
S_2	für Motor steht	T_2	für Schlüssel abziehen
S_3	für Schlüssel abgezogen	T_3	für Schlüssel in Zündschloss stecken

Petrinetze – Komplexes Beispiel (I)

Beispiel: **Motorsteuerung**

- rein sequentieller Ablauf ohne Nebenläufigkeiten
- ein deutlich kausaler Zusammenhang zwischen den Stellen S erkennbar



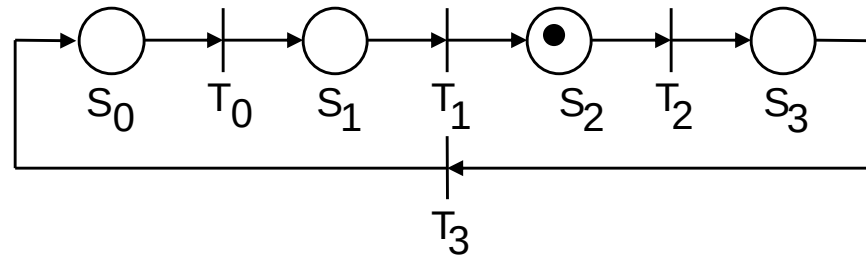
Zustände S_i (Stellen) und Transitionen T_j

S_0	für Zündschlüssel bereit	T_0	für Zündung einschalten und Motor starten
S_1	für Zündung ein, Motor läuft	T_1	für Zündung ausschalten
S_2	für Motor steht	T_2	für Schlüssel abziehen
S_3	für Schlüssel abgezogen	T_3	für Schlüssel in Zündschloss stecken

Petrinetze – Komplexes Beispiel (I)

Beispiel: **Motorsteuerung**

- rein sequentieller Ablauf ohne Nebenläufigkeiten
- ein deutlich kausaler Zusammenhang zwischen den Stellen S erkennbar



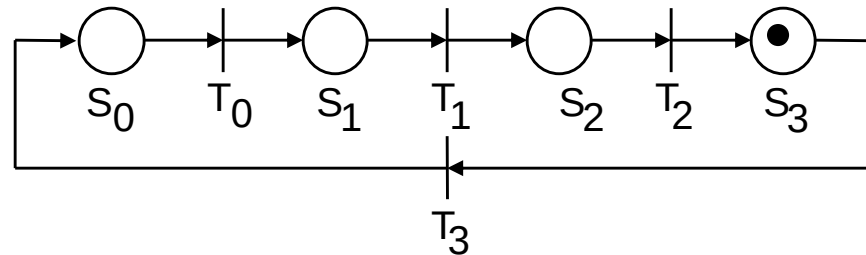
Zustände S_i (Stellen) und Transitionen T_j

S_0	für Zündschlüssel bereit	T_0	für Zündung einschalten und Motor starten
S_1	für Zündung ein, Motor läuft	T_1	für Zündung ausschalten
S_2	für Motor steht	T_2	für Schlüssel abziehen
S_3	für Schlüssel abgezogen	T_3	für Schlüssel in Zündschloss stecken

Petrinetze – Komplexes Beispiel (I)

Beispiel: **Motorsteuerung**

- rein sequentieller Ablauf ohne Nebenläufigkeiten
- ein deutlich kausaler Zusammenhang zwischen den Stellen S erkennbar

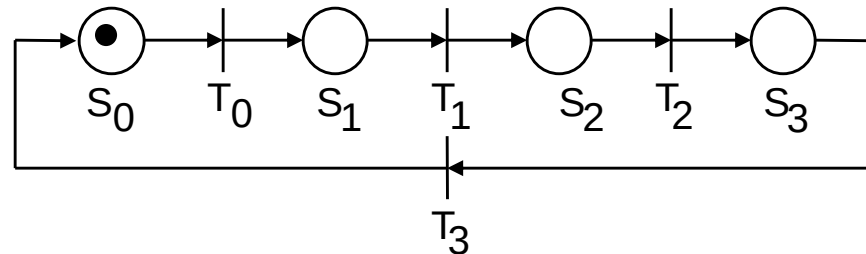


Zustände S_i (Stellen) und Transitionen T_j

S_0	für Zündschlüssel bereit	T_0	für Zündung einschalten und Motor starten
S_1	für Zündung ein, Motor läuft	T_1	für Zündung ausschalten
S_2	für Motor steht	T_2	für Schlüssel abziehen
S_3	für Schlüssel abgezogen	T_3	für Schlüssel in Zündschloss stecken

Beispiel: **Motorsteuerung**

- rein sequentieller Ablauf ohne Nebenläufigkeiten
- ein deutlich kausaler Zusammenhang zwischen den Stellen S erkennbar



Zustände S_i (Stellen) und Transitionen T_j

S_0	für Zündschlüssel bereit	T_0	für Zündung einschalten und Motor starten
S_1	für Zündung ein, Motor läuft	T_1	für Zündung ausschalten
S_2	für Motor steht	T_2	für Schlüssel abziehen
S_3	für Schlüssel abgezogen	T_3	für Schlüssel in Zündschloss stecken

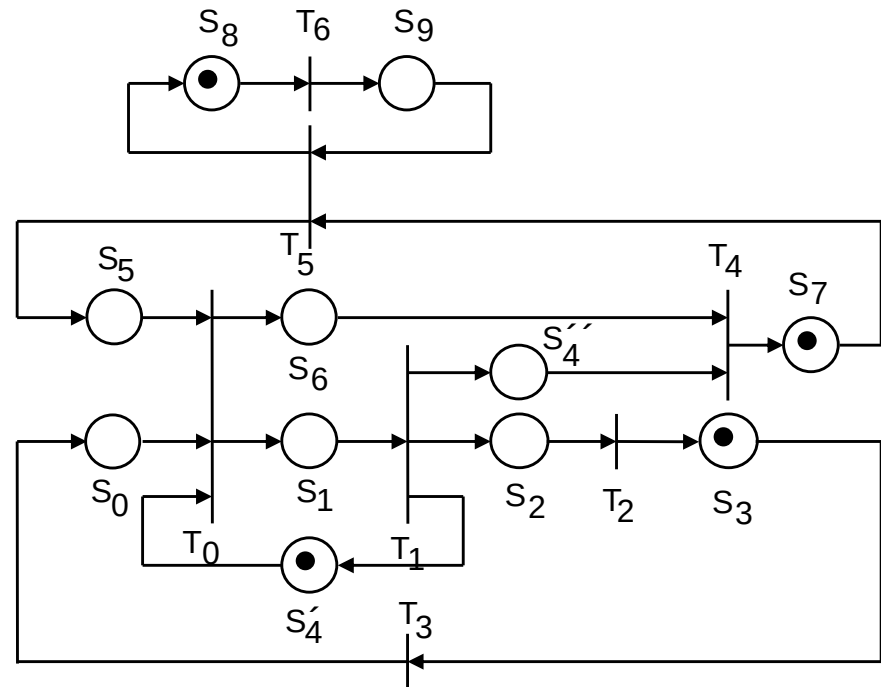
Petrinetze – Komplexes Beispiel (II)

Beispiel: Motorsteuerung

Zusätzliche Stellen und Transitionen:

S4 für Zündung aus
S5 für WFS ist deaktiviert
S6 für WFS bleibt deaktiviert
S7 für WFS aktiviert
S8 für Codeschlüssel für WFS bereit
S9 für Codeschlüssel für WFS abgezogen

T4 für WFS aktivieren
T5 für WFS deaktivieren
T6 für Codeschlüssel gibt WFS frei



→ vier Anfangsmarken im Initialzustand

→ Nebenläufigkeiten werden erfasst

→ dieses einfache Modell erfasst jedoch nicht alle Betriebssituationen des Fahrzeugs

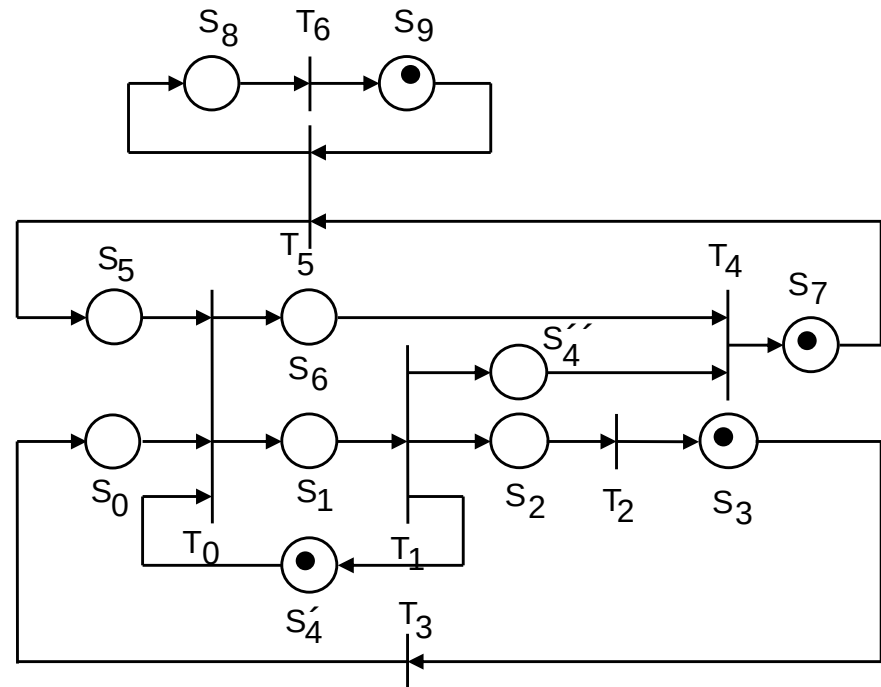
Petrinetze – Komplexes Beispiel (II)

Beispiel: **Motorsteuerung**

Zusätzliche **Stellen** und **Transitionen**:

S4 für Zündung aus
S5 für WFS ist deaktiviert
S6 für WFS bleibt deaktiviert
S7 für WFS aktiviert
S8 für Codeschlüssel für WFS bereit
S9 für Codeschlüssel für WFS abgezogen

T4 für WFS aktivieren
T5 für WFS deaktivieren
T6 für Codeschlüssel gibt WFS frei



→ vier Anfangsmarken im Initialzustand

→ Nebenläufigkeiten werden erfasst

→ dieses einfache Modell erfasst jedoch nicht alle Betriebssituationen des Fahrzeugs

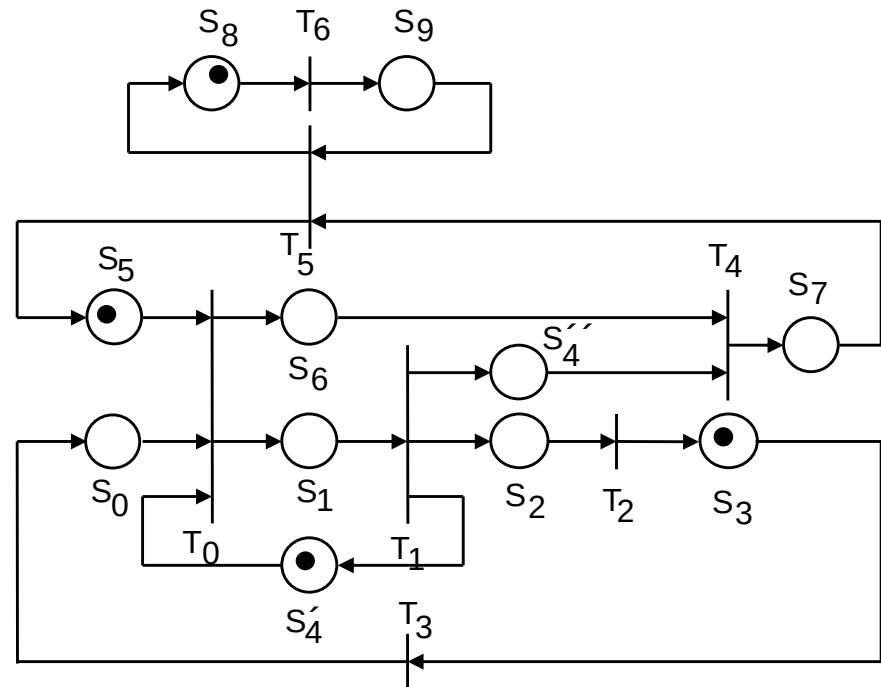
Petrinetze – Komplexes Beispiel (II)

Beispiel: **Motorsteuerung**

Zusätzliche **Stellen** und **Transitionen**:

S4 für Zündung aus
S5 für WFS ist deaktiviert
S6 für WFS bleibt deaktiviert
S7 für WFS aktiviert
S8 für Codeschlüssel für WFS bereit
S9 für Codeschlüssel für WFS abgezogen

T4 für WFS aktivieren
T5 für WFS deaktivieren
T6 für Codeschlüssel gibt WFS frei



→ vier Anfangsmarken im Initialzustand

→ Nebenläufigkeiten werden erfasst

→ dieses einfache Modell erfasst jedoch nicht alle Betriebssituationen des Fahrzeugs

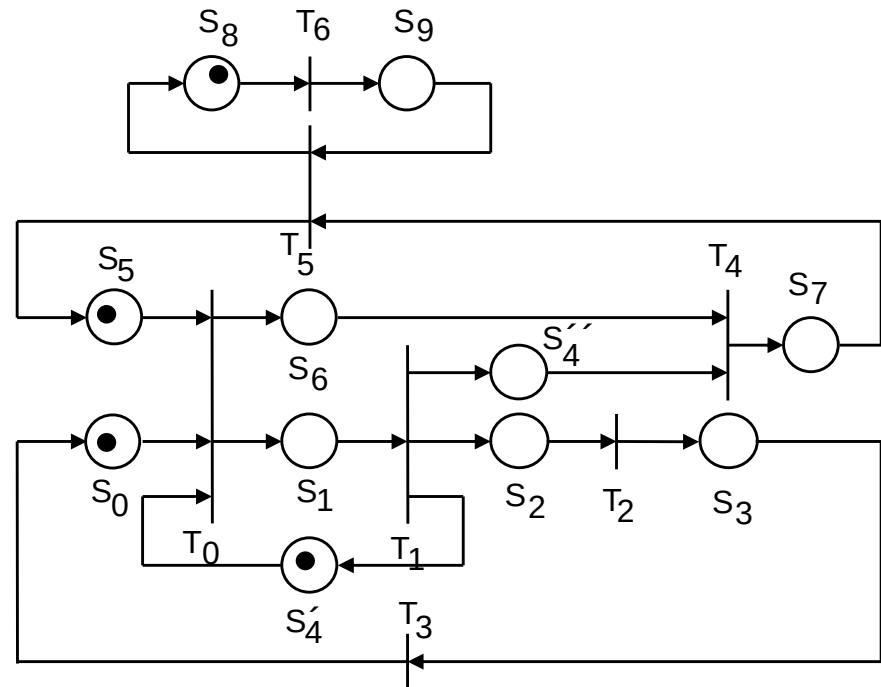
Petrinetze – Komplexes Beispiel (II)

Beispiel: Motorsteuerung

Zusätzliche Stellen und Transitionen:

S4 für Zündung aus
S5 für WFS ist deaktiviert
S6 für WFS bleibt deaktiviert
S7 für WFS aktiviert
S8 für Codeschlüssel für WFS bereit
S9 für Codeschlüssel für WFS abgezogen

T4 für WFS aktivieren
T5 für WFS deaktivieren
T6 für Codeschlüssel gibt WFS frei



→ vier Anfangsmarken im Initialzustand

→ Nebenläufigkeiten werden erfasst

→ dieses einfache Modell erfasst jedoch nicht alle Betriebssituationen des Fahrzeugs

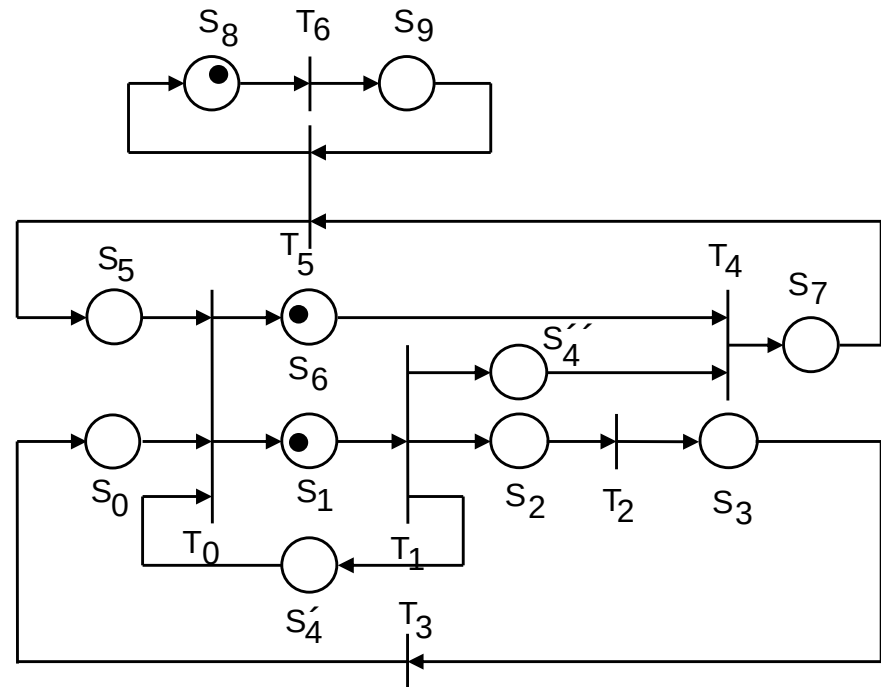
Petrinetze – Komplexes Beispiel (II)

Beispiel: Motorsteuerung

Zusätzliche Stellen und Transitionen:

S4 für Zündung aus
S5 für WFS ist deaktiviert
S6 für WFS bleibt deaktiviert
S7 für WFS aktiviert
S8 für Codeschlüssel für WFS bereit
S9 für Codeschlüssel für WFS abgezogen

T4 für WFS aktivieren
T5 für WFS deaktivieren
T6 für Codeschlüssel gibt WFS frei



→ vier Anfangsmarken im Initialzustand

→ Nebenläufigkeiten werden erfasst

→ dieses einfache Modell erfasst jedoch nicht alle Betriebssituationen des Fahrzeugs

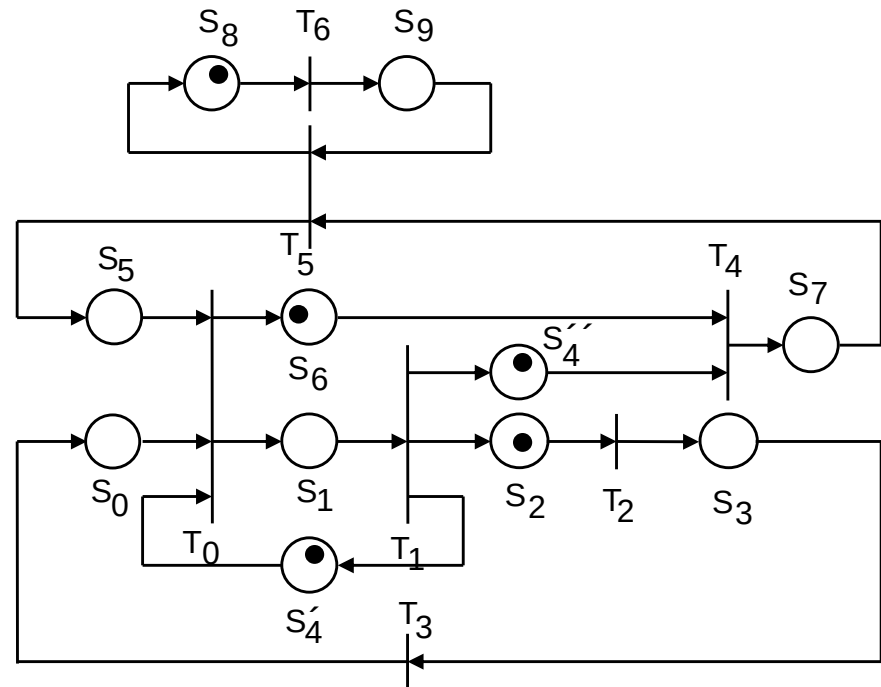
Petrinetze – Komplexes Beispiel (II)

Beispiel: Motorsteuerung

Zusätzliche Stellen und Transitionen:

S4 für Zündung aus
S5 für WFS ist deaktiviert
S6 für WFS bleibt deaktiviert
S7 für WFS aktiviert
S8 für Codeschlüssel für WFS bereit
S9 für Codeschlüssel für WFS abgezogen

T4 für WFS aktivieren
T5 für WFS deaktivieren
T6 für Codeschlüssel gibt WFS frei



→ vier Anfangsmarken im Initialzustand

→ Nebenläufigkeiten werden erfasst

→ dieses einfache Modell erfasst jedoch nicht alle Betriebssituationen des Fahrzeugs

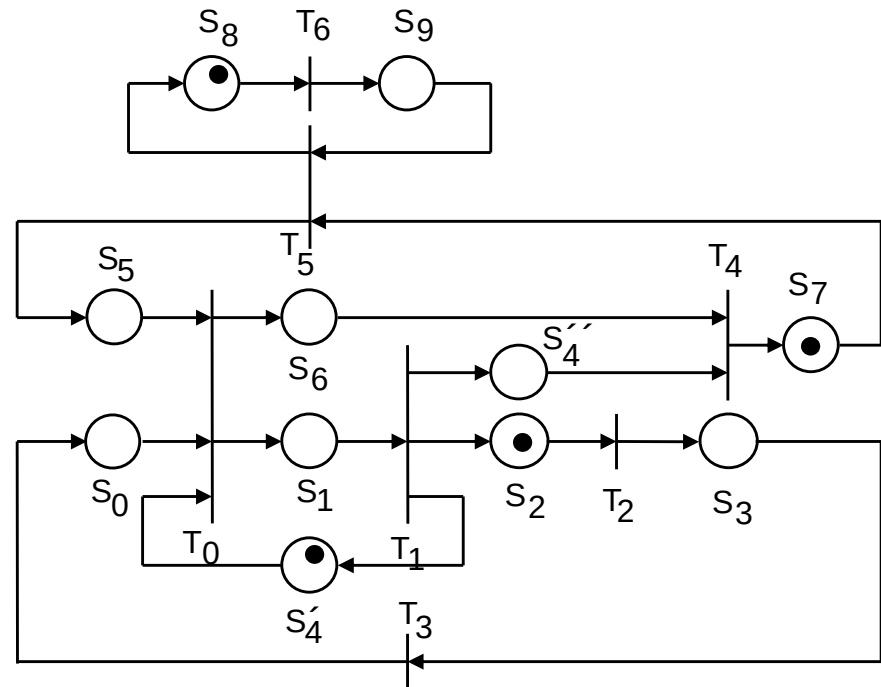
Petrinetze – Komplexes Beispiel (II)

Beispiel: Motorsteuerung

Zusätzliche Stellen und Transitionen:

S4 für Zündung aus
S5 für WFS ist deaktiviert
S6 für WFS bleibt deaktiviert
S7 für WFS aktiviert
S8 für Codeschlüssel für WFS bereit
S9 für Codeschlüssel für WFS abgezogen

T4 für WFS aktivieren
T5 für WFS deaktivieren
T6 für Codeschlüssel gibt WFS frei



→ vier Anfangsmarken im Initialzustand

→ Nebenläufigkeiten werden erfasst

→ dieses einfache Modell erfasst jedoch nicht alle Betriebssituationen des Fahrzeugs

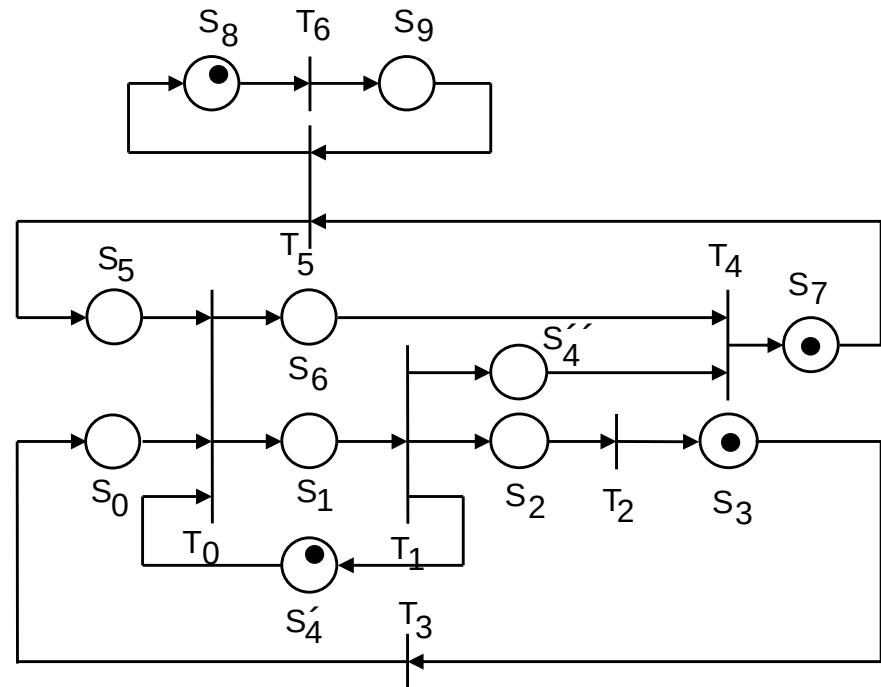
Petrinetze – Komplexes Beispiel (II)

Beispiel: Motorsteuerung

Zusätzliche Stellen und Transitionen:

S4 für Zündung aus
S5 für WFS ist deaktiviert
S6 für WFS bleibt deaktiviert
S7 für WFS aktiviert
S8 für Codeschlüssel für WFS bereit
S9 für Codeschlüssel für WFS abgezogen

T4 für WFS aktivieren
T5 für WFS deaktivieren
T6 für Codeschlüssel gibt WFS frei



→ vier Anfangsmarken im Initialzustand

→ Nebenläufigkeiten werden erfasst

→ dieses einfache Modell erfasst jedoch nicht alle Betriebssituationen des Fahrzeugs

Petrinetze – Komplexes Beispiel (III)

Beispiel: **Motorsteuerung**

- der vorgeführte Ablauf stellt nur *ein mögliches Szenario* für die Funktion des Modells dar
- eine vollständige Verifikation der Funktionsrichtigkeit in allen Situationen ist damit nicht erreicht