Petri-Netze

Einführung

Magdeburg Research and Competence Cluster
Arbeitsgruppe Wirtschaftsinformatik

Fakultät für Informatik
Institut für Technische und Betriebliche Informationssysteme

Otto-von-Guericke-Universität Magdeburg

Prof. Dr. Klaus Turowski

klaus.turowski@ovgu.de
https://mrcc.ovgu.de

Überblick

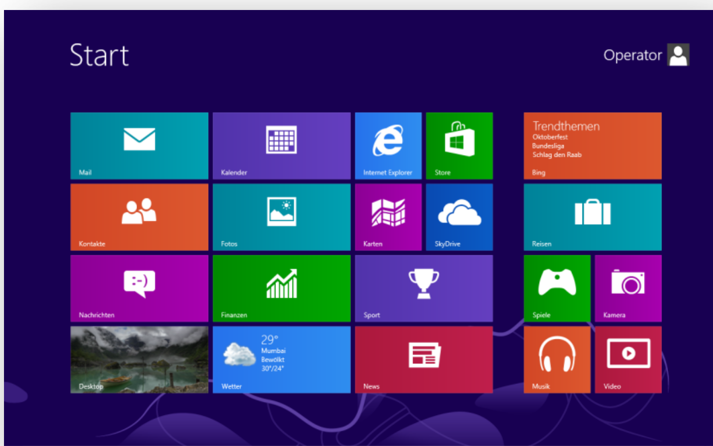
- Einsatzgebiete und Geschichte
- Graphische Grundlagen
- Netze
  - Grundsituationen
- Bedingungs/Ereignis-Netze
  - Netzeigenschaften
- Stellen/Transitionen-Netze
  - Weitere Eigenschaften
- Erreichbarkeitsgraph
- Analysemöglichkeiten und Unterarten von Petri-Netzen

Petri-Netze: Einsatzgebiete

- Geschäftsvorgänge, betriebliche Organisationsstrukturen, Behördenwege, (Computer-)Kommunikation, Bedienungs-anleitungen, Bauanleitungen
- Diese beinhalten Prozesse, bei denen durch Ereignisse und Aktivitäten der Systemzustand verändert wird
- Der Systemzustand wird gekennzeichnet durch
  - Abstrakte (Zahlen oder Genehmigungen) oder materielle (Bauteile, Werkzeuge oder Münzen) Objekte, die benötigt, verbraucht oder erzeugt werden sowie
  - Bedingungen (z.B. ein Gegenstand liegt an einem Ort vor) oder Tatsachen (Variable muss Wert x haben), welche benötigt, aufgehoben oder herbeigeführt werden.

Petri-Netze: Geschichte

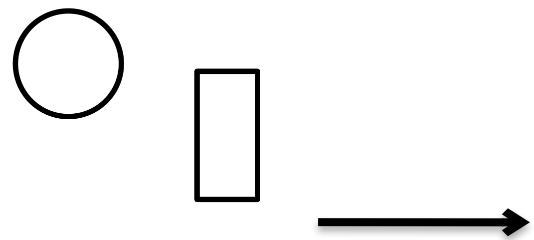
- Carl Adam Petri (1926–2010): Kommunikation mit Automaten (Dissertationsschrift 1962)
- Zunächst unbeachtet, erst durch die Softwarekrise praktische Relevanz

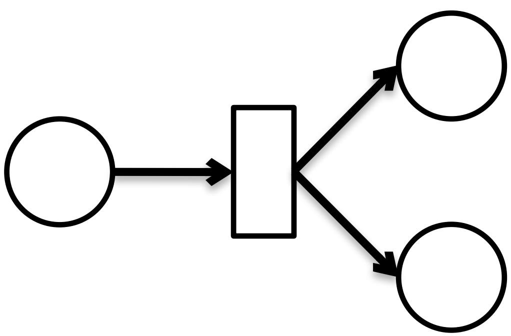


- Eingesetzt in vielen Anwendungsgebieten
  - Betriebssystemmodellierung
  - Workflow-Management
  - Softwareentwicklung
  - Geschäftsprozessmodellierung

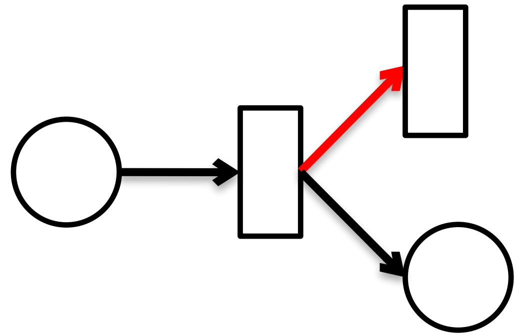
Petri-Netze als Graph

- Ein Petri-Netz ist ein gerichteter, bipartiter Graph
- Bipartit: Es gibt zwei Arten von Knoten und jede Kante verbindet einen Knoten der einen Art mit einem der anderen
- Elemente
  - Knoten „Stelle“ (Place)
  - Knoten „Transition“
  - Gerichtete Kante





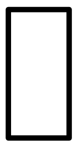
erlaubt



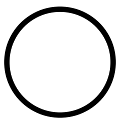
nicht erlaubt

Beispiel Bibliotheksbenutzung

- Transitionen repräsentieren Ereignisse
- Stellen repräsentieren Objekte und Bedingungen

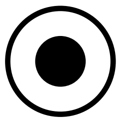


Nutzer trifft ein



Buch ausleihbar

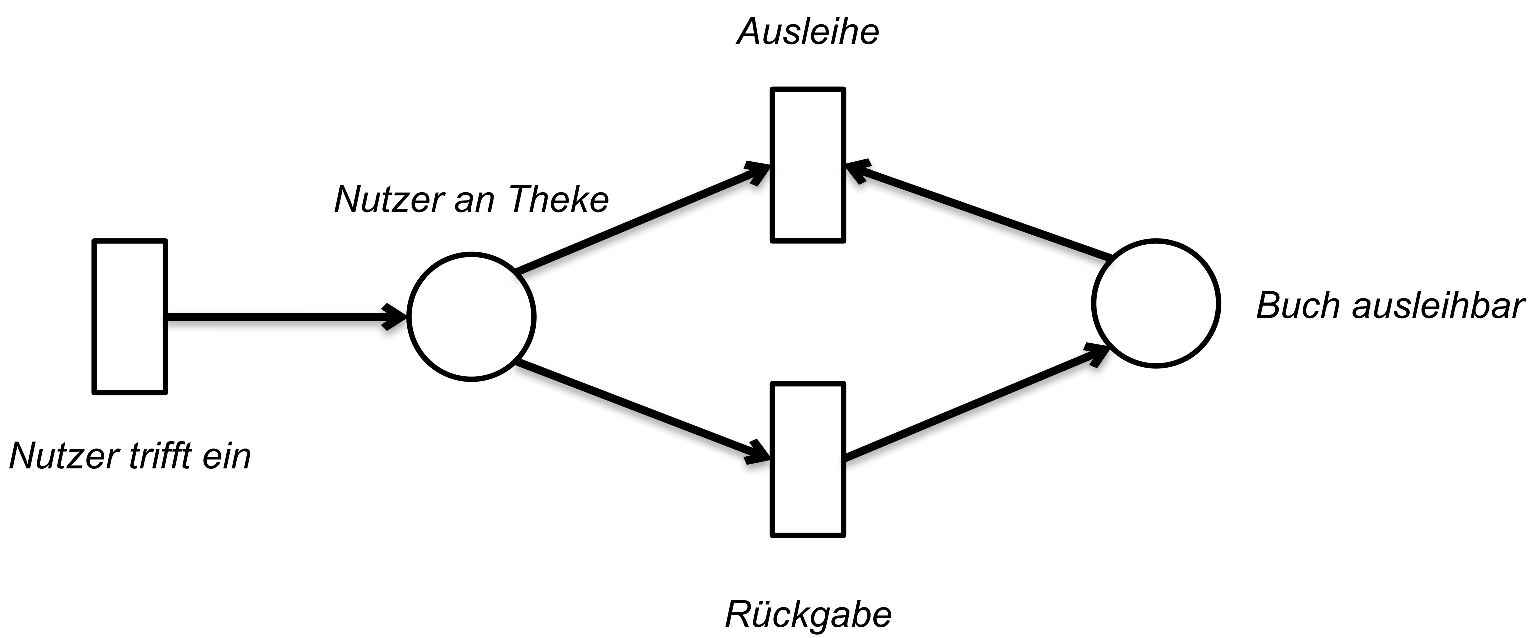
- Eine Stelle kann markiert werden; die Marke (Token) zeigt, dass eine Bedingung bzw. ein Objekt vorliegt



Buch ausleihbar

- Markierung aller Stellen bestimmt den Systemzustand

Beispiel Bibliotheksbenutzung Fortsetzung



Bedingungs/Ereignis-Netze: Definitionen

- ## Definition 1

Ein Netz ist ein Tripel $N = (S, T, F)$ mit

(i) einer Menge $S = \{s_1, s_2, \ldots, s_n\}$ von Stellen,
(ii) einer Menge $T = \{t_1, t_2, \ldots, t_m\}$ von Transitionen,
(iii) $S \cap T = \emptyset$ und
(iv) einer zweistelligen Flussrelation $F \subseteq (S \times T) \cup (T \times S)$ als Menge der Kanten.

- ## Definition 2

Sei $N = (S, T, F)$ ein Netz. Für einen Knoten $x \in (S \cup T)$ ist

(i) $\bullet x = \{y \in (S \cup T) | (y, x) \in F\}$ der Vorbereich und
(ii) $x \bullet = \{y \in (S \cup T) | (x, y) \in F\}$ der Nachbereich von x.

Netze: Definitionen

- Definition 3

Sei $N = (S, T, F)$ ein Netz.

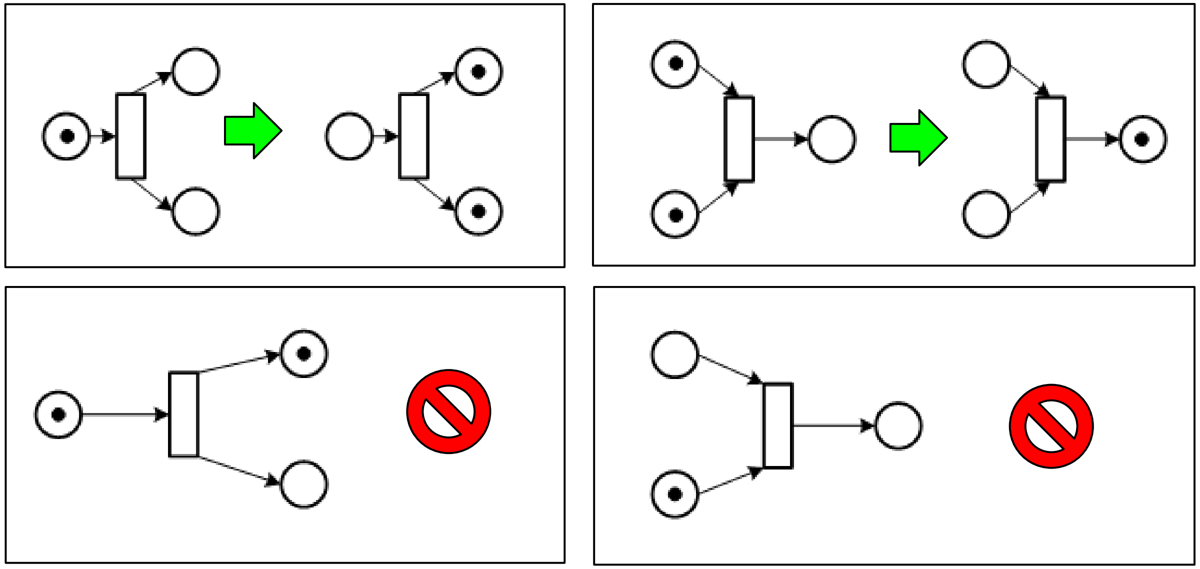
(i) Eine Teilmenge $c \subseteq S$ heißt Fall oder Markierung.

(ii) Eine Transition $t \in T$ ist c-aktiviert, genau dann wenn $\bullet t \subseteq c \wedge t \bullet \cap c = \emptyset$.

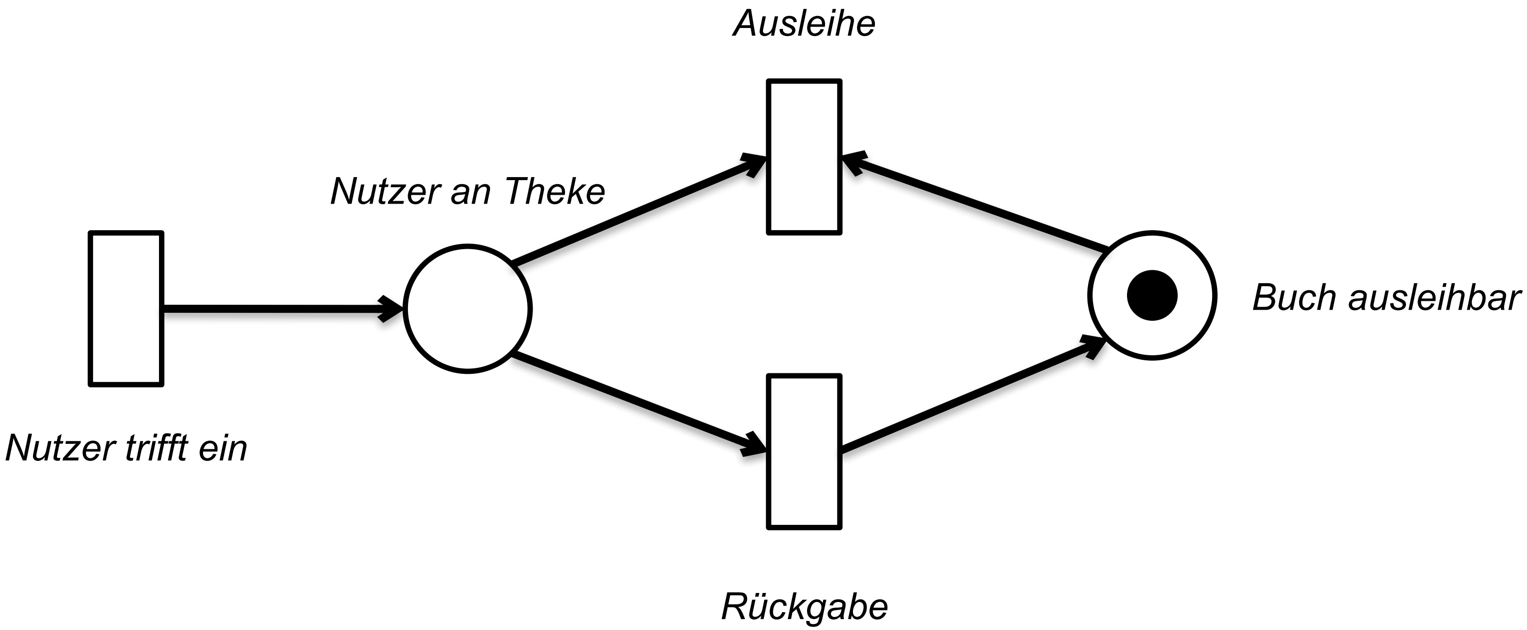
(iii) Sei $t \in T$ c-aktiviert mit $c \subseteq S$. $c' = (c \setminus \bullet t) \cup t\bullet$ heißt Folgefall oder Folgemarkierung von c unter t. Notation: $c >_t c'$

- Alle markierten Stellen bilden die Menge , den Systemzustand
- Ist eine Transition c-aktiviert, „schaltet“ oder „feuert“ sie
  - Marken im Vorbereich werden vernichtet
  - Marken im Nachbereich werden erzeugt
- Es entsteht die Folgemarkierung c’, der neue Systemzustand

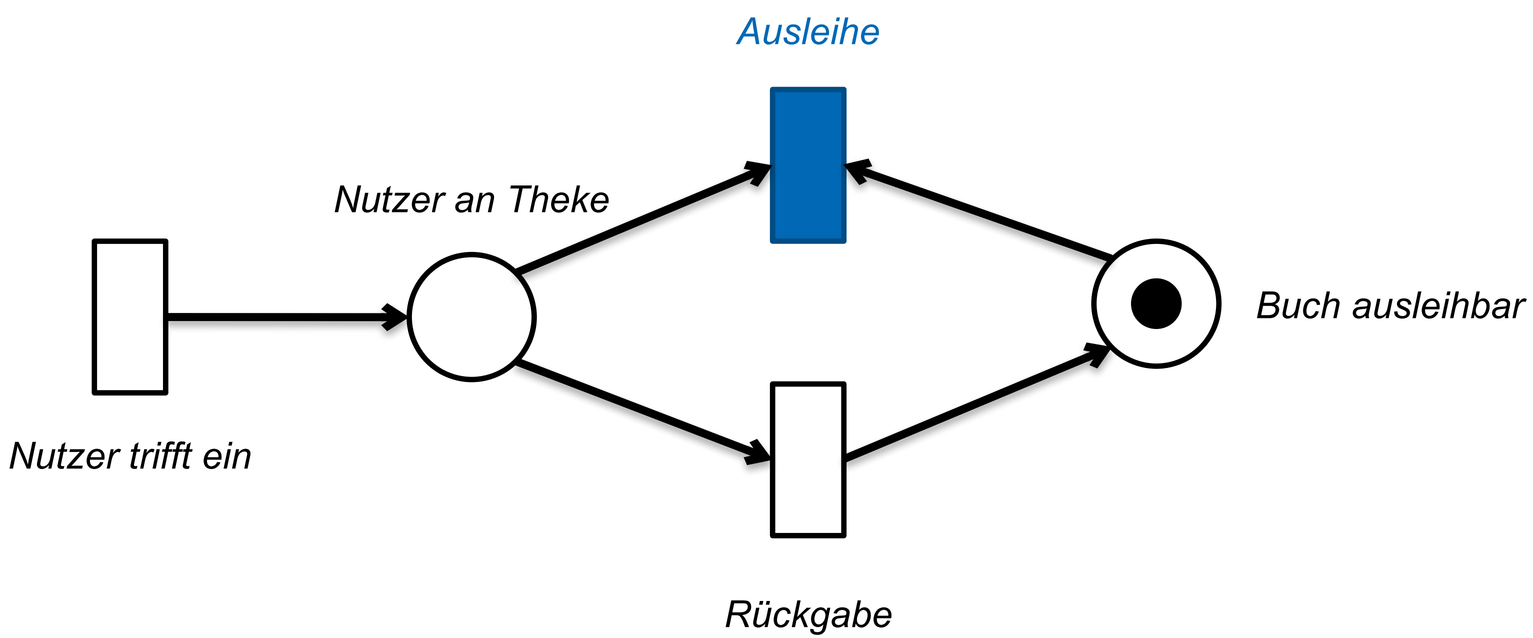
Schalten eines Petri-Netzes



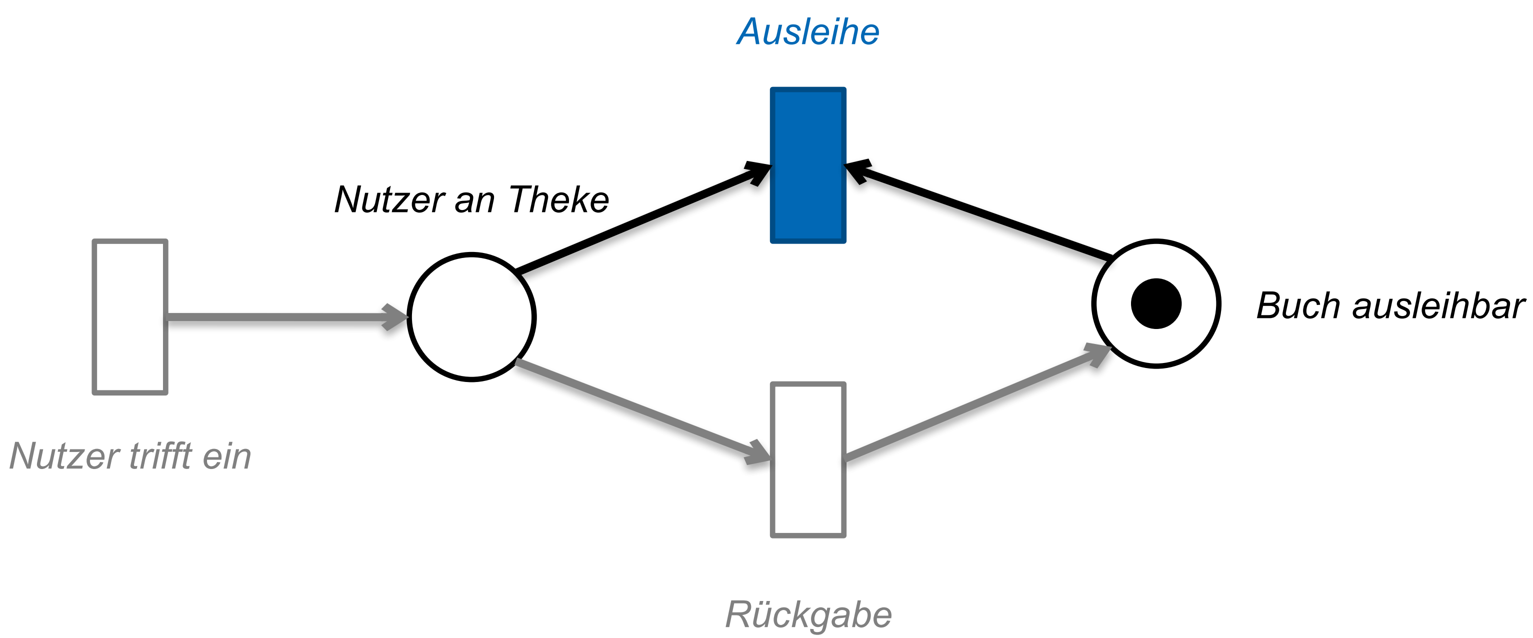
Schaltbeispiel – Anfangsfall C_0



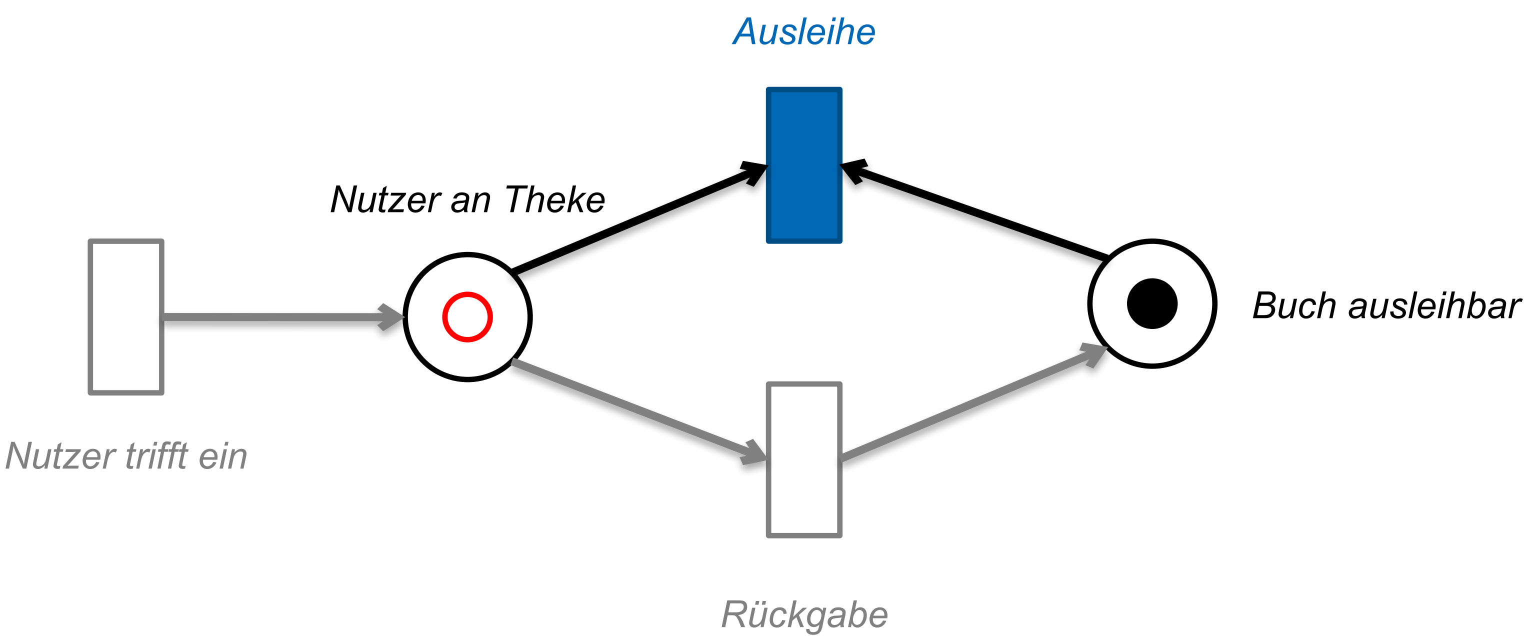
Schaltbeispiel – „Ausleihe“ aktiv?



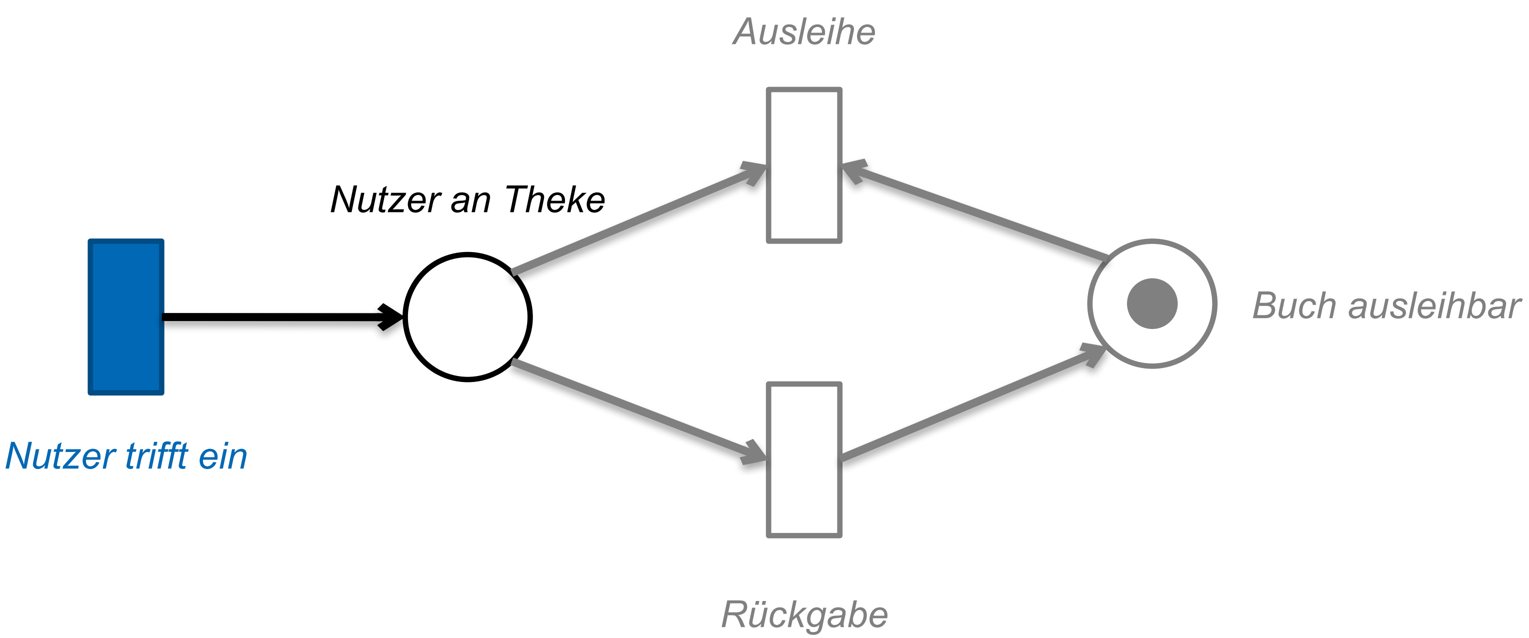
Schaltbeispiel – „Ausleihe“ aktiv?



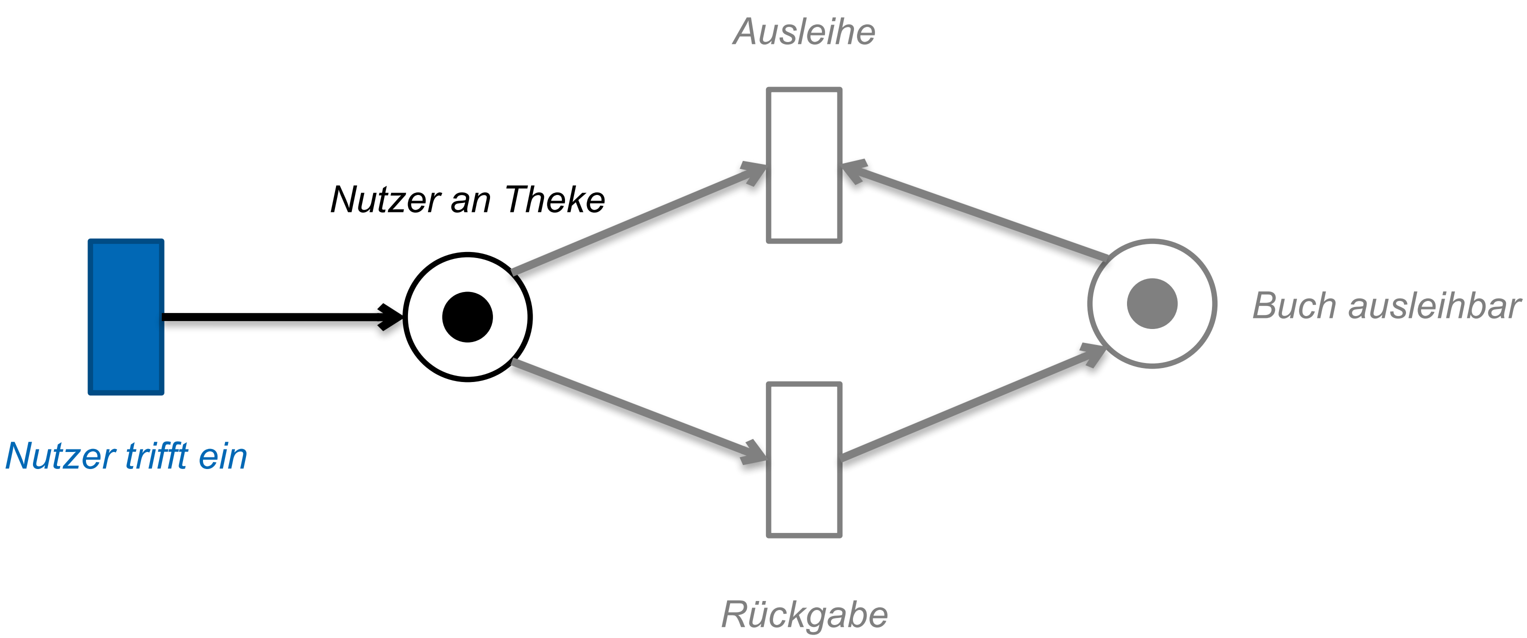
Schaltbeispiel – „Ausleihe“ aktiv?



Schaltbeispiel – „Nutzer trifft ein“ aktiv?

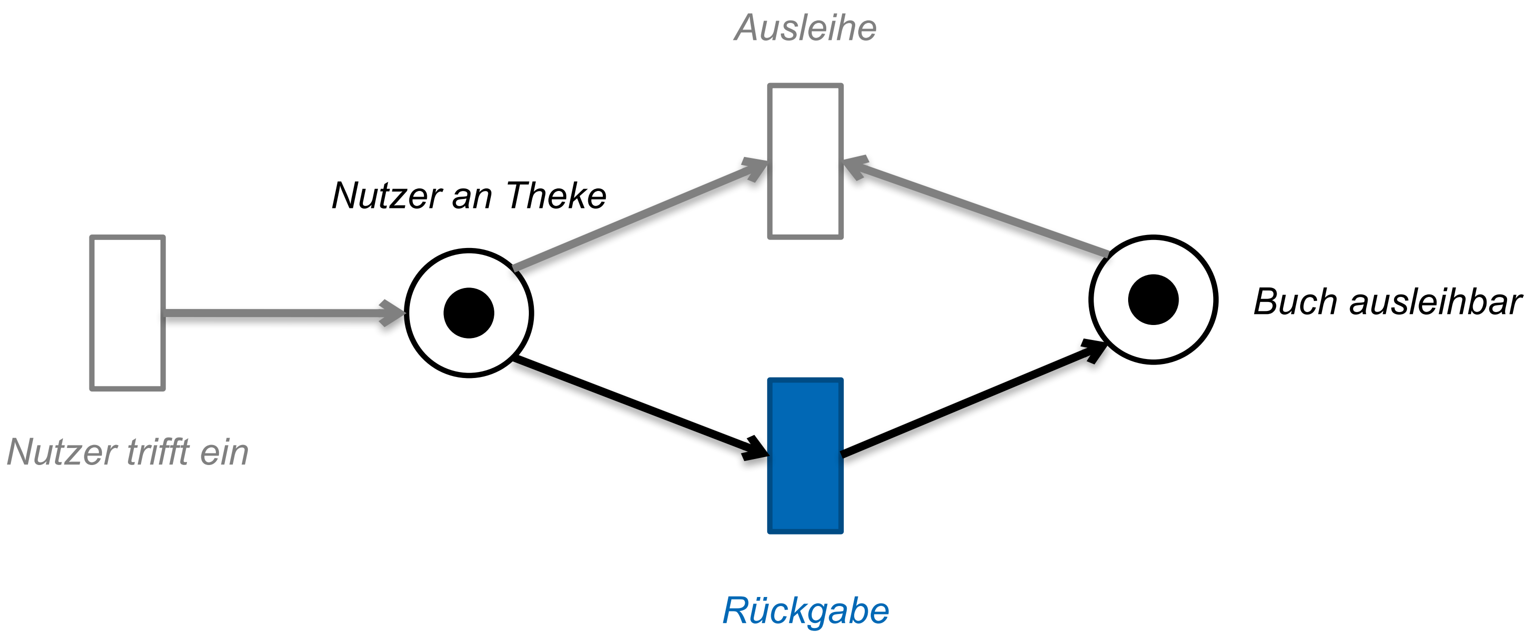


Schaltbeispiel – „Nutzer trifft ein“ feuert

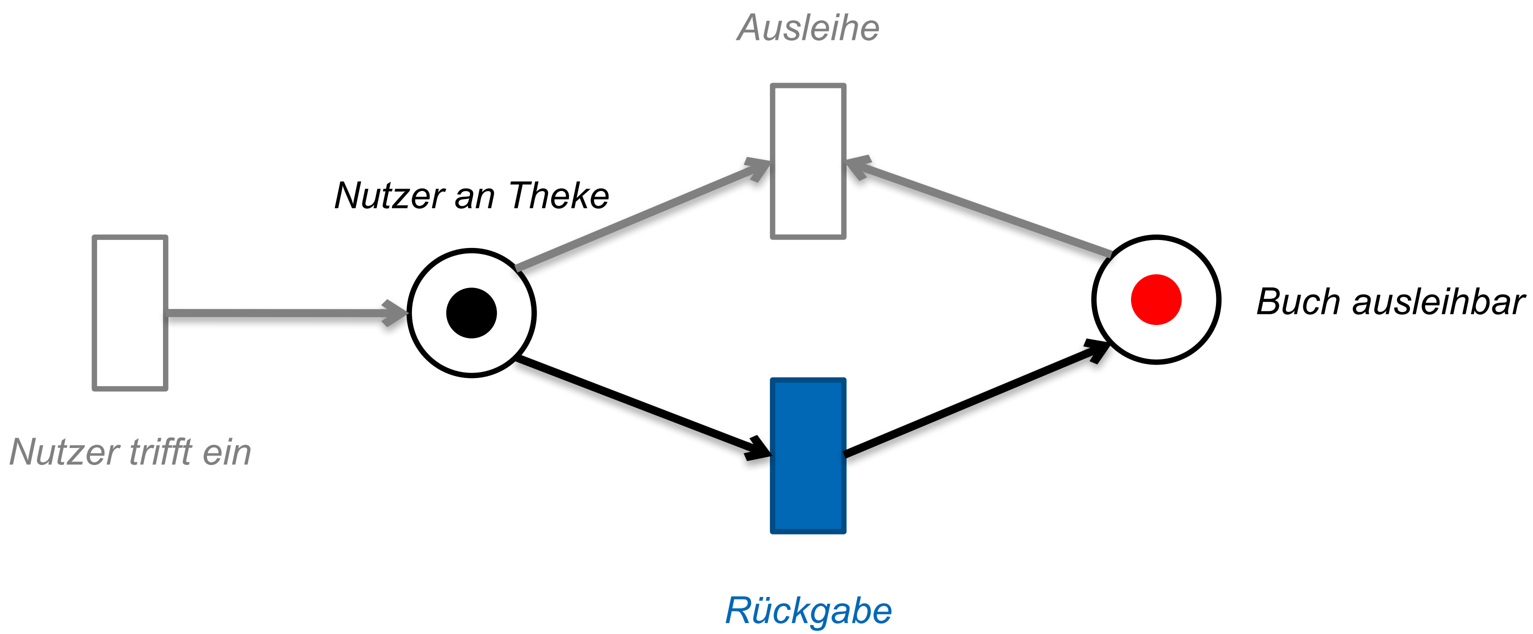


- Neue Markierung c_1

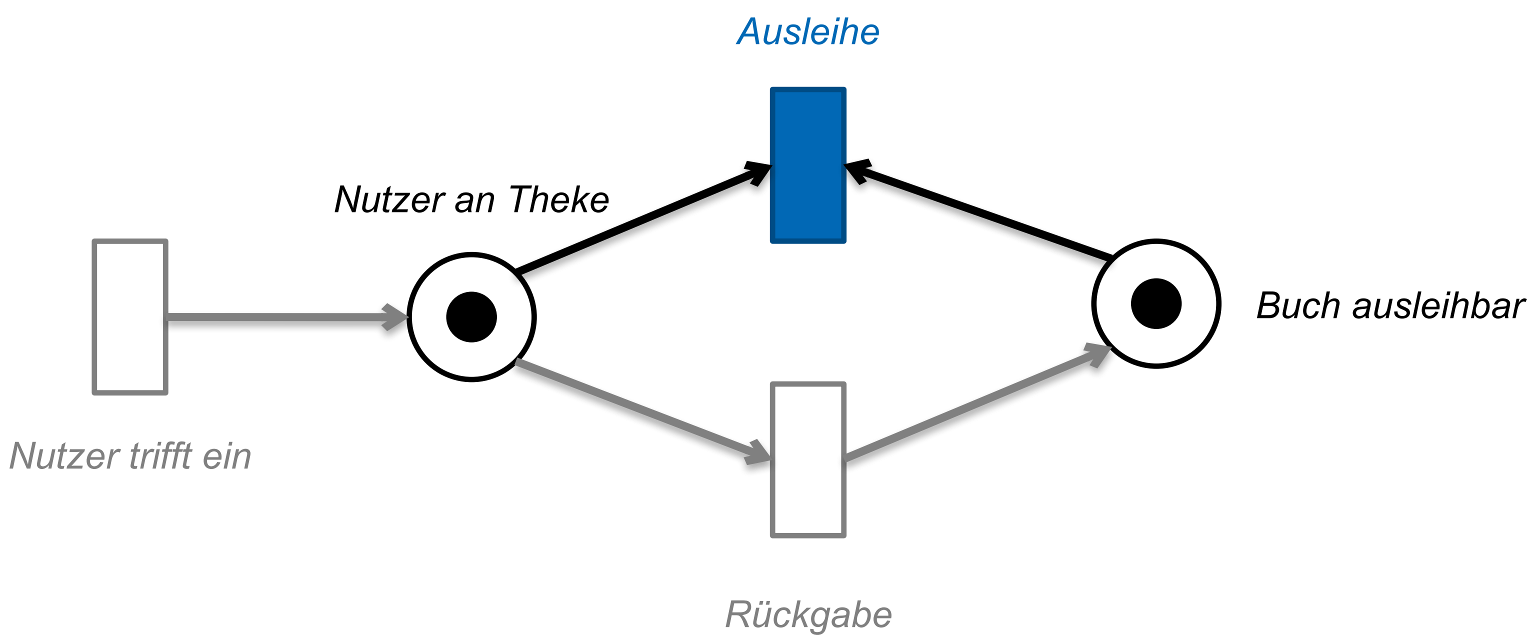
Schaltbeispiel – „Rückgabe“ aktiv?



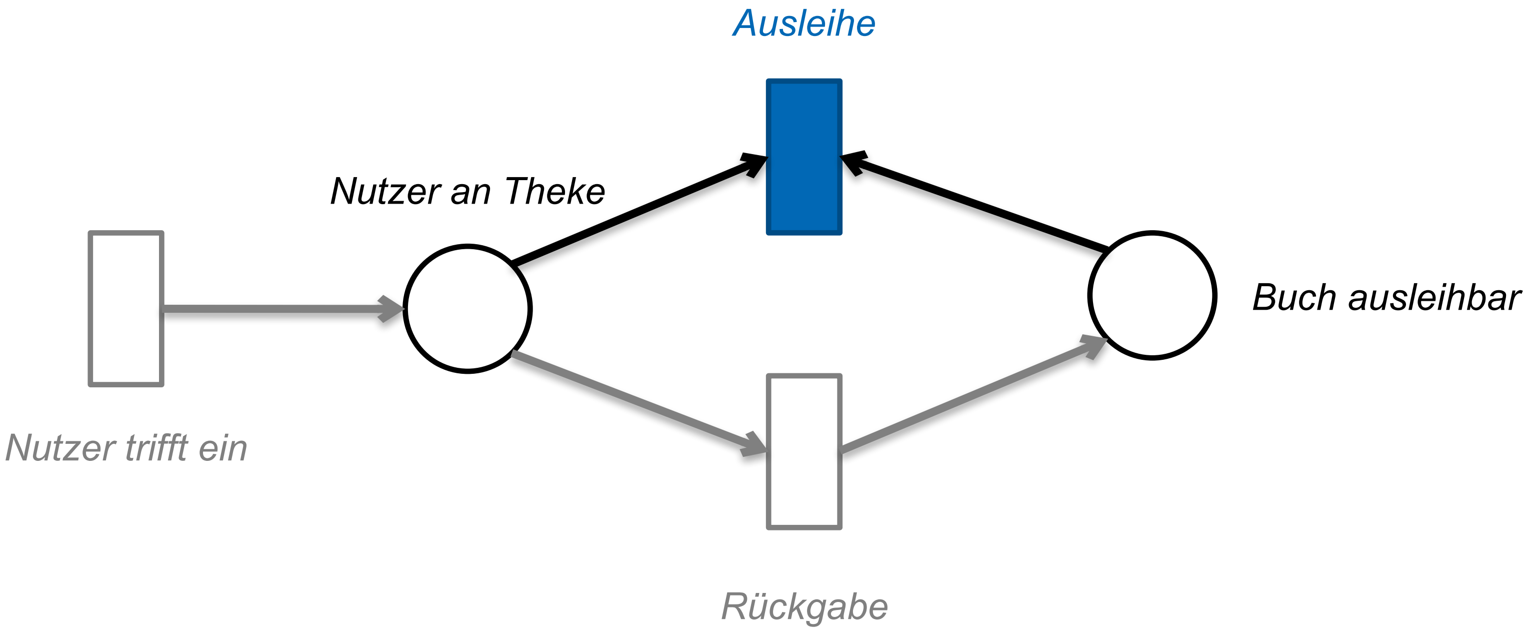
Schaltbeispiel – „Rückgabe“ aktiv?



Schaltbeispiel – „Ausleihe“ aktiv?

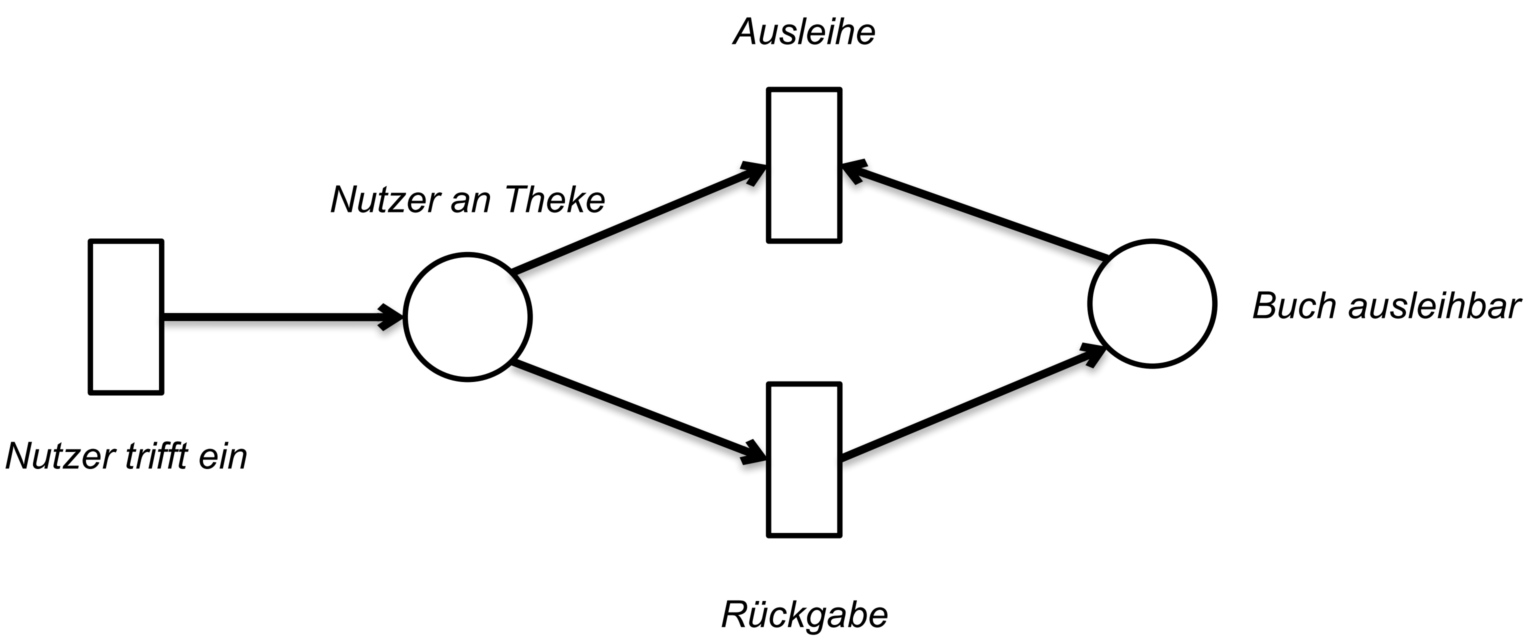


Schaltbeispiel – „Ausleihe“ feuert

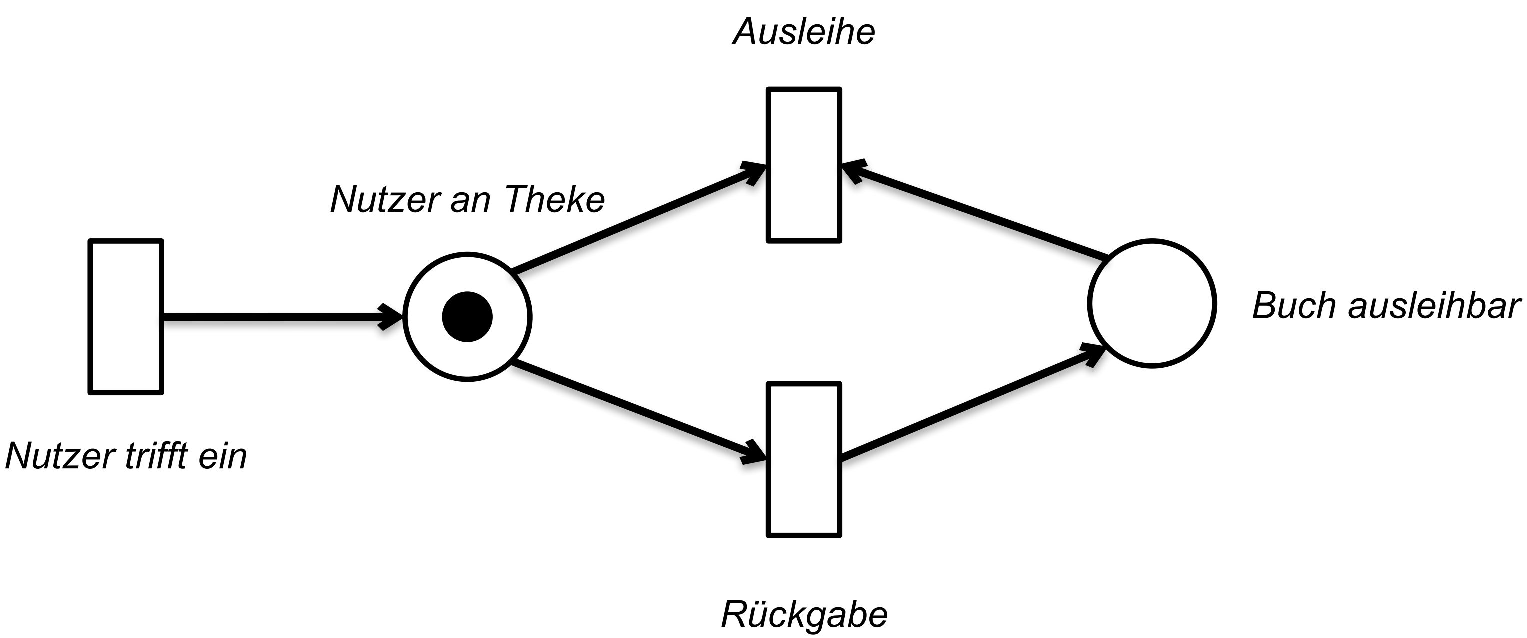


Neue Markierung c_2

Schaltbeispiel – weiterer Verlauf

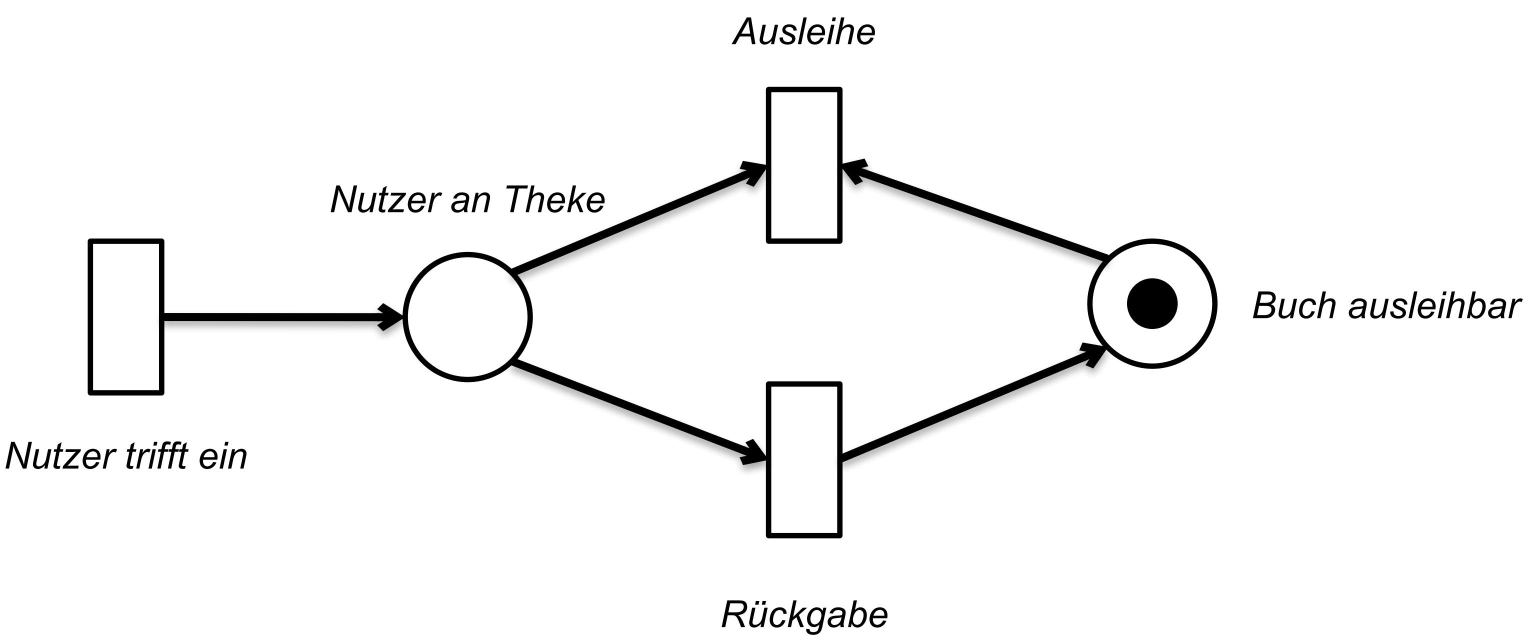


Schaltbeispiel – weiterer Verlauf



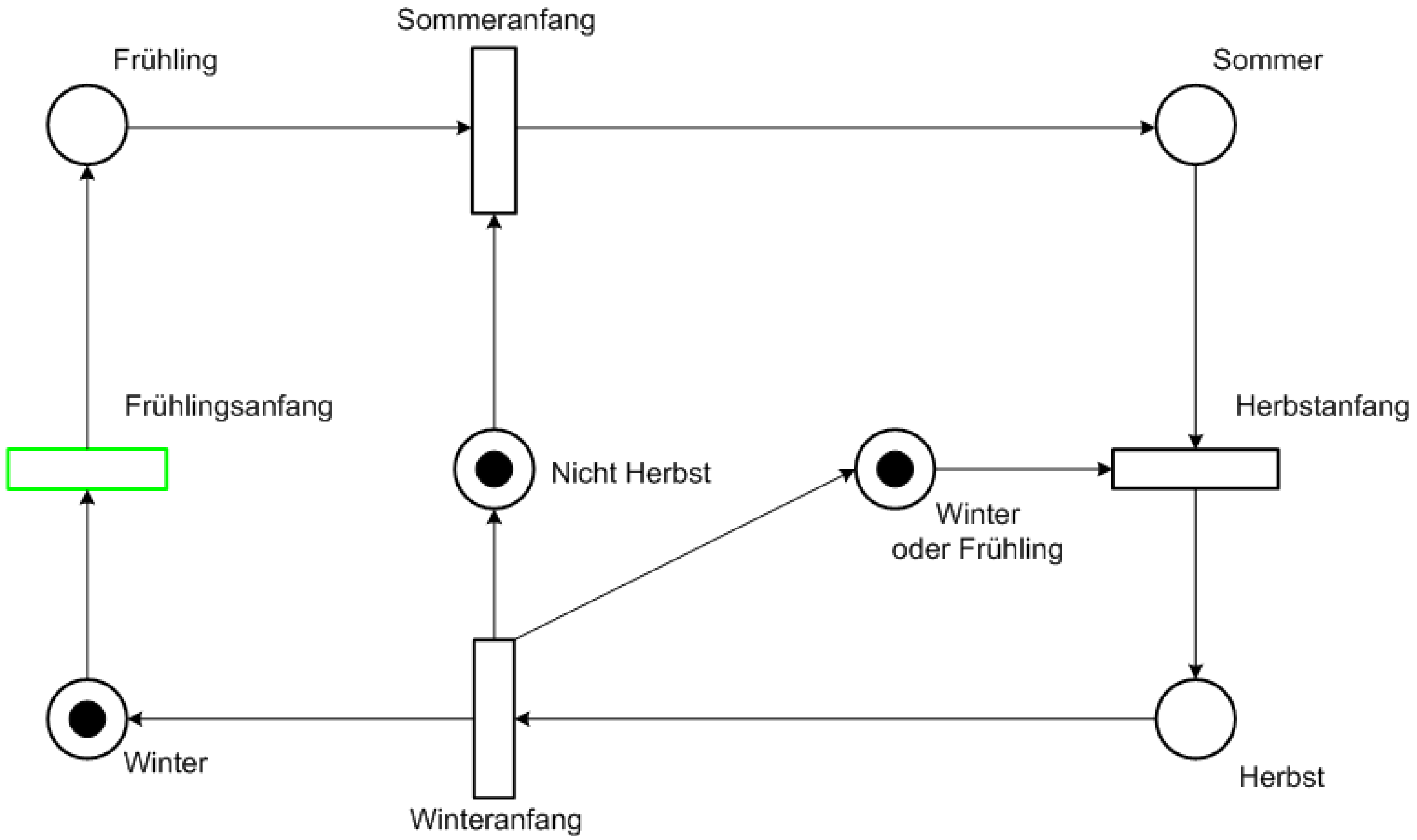
Neue Markierung c_3

Schaltbeispiel – weiterer Verlauf

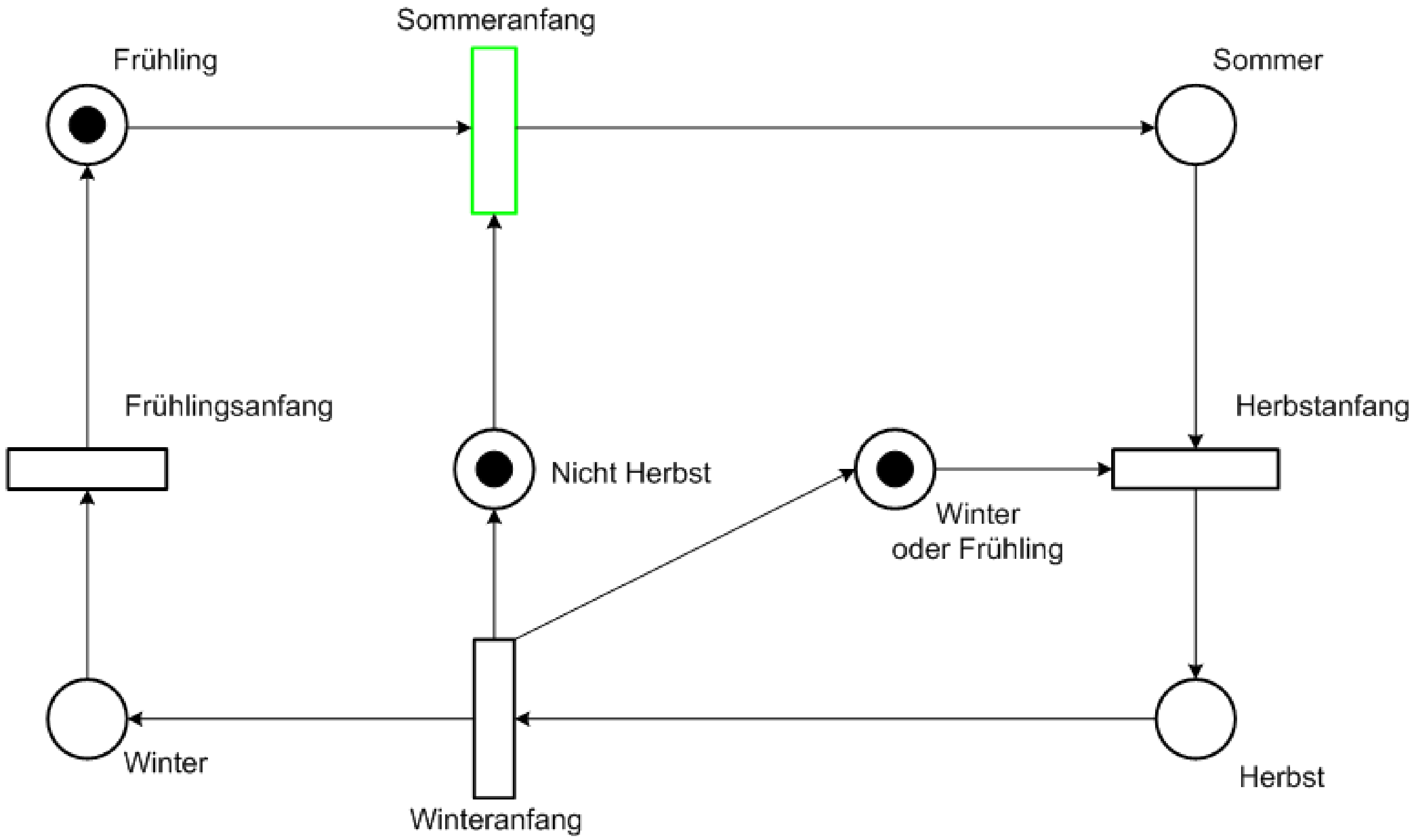


Neue Markierung $c_4 = c_0$

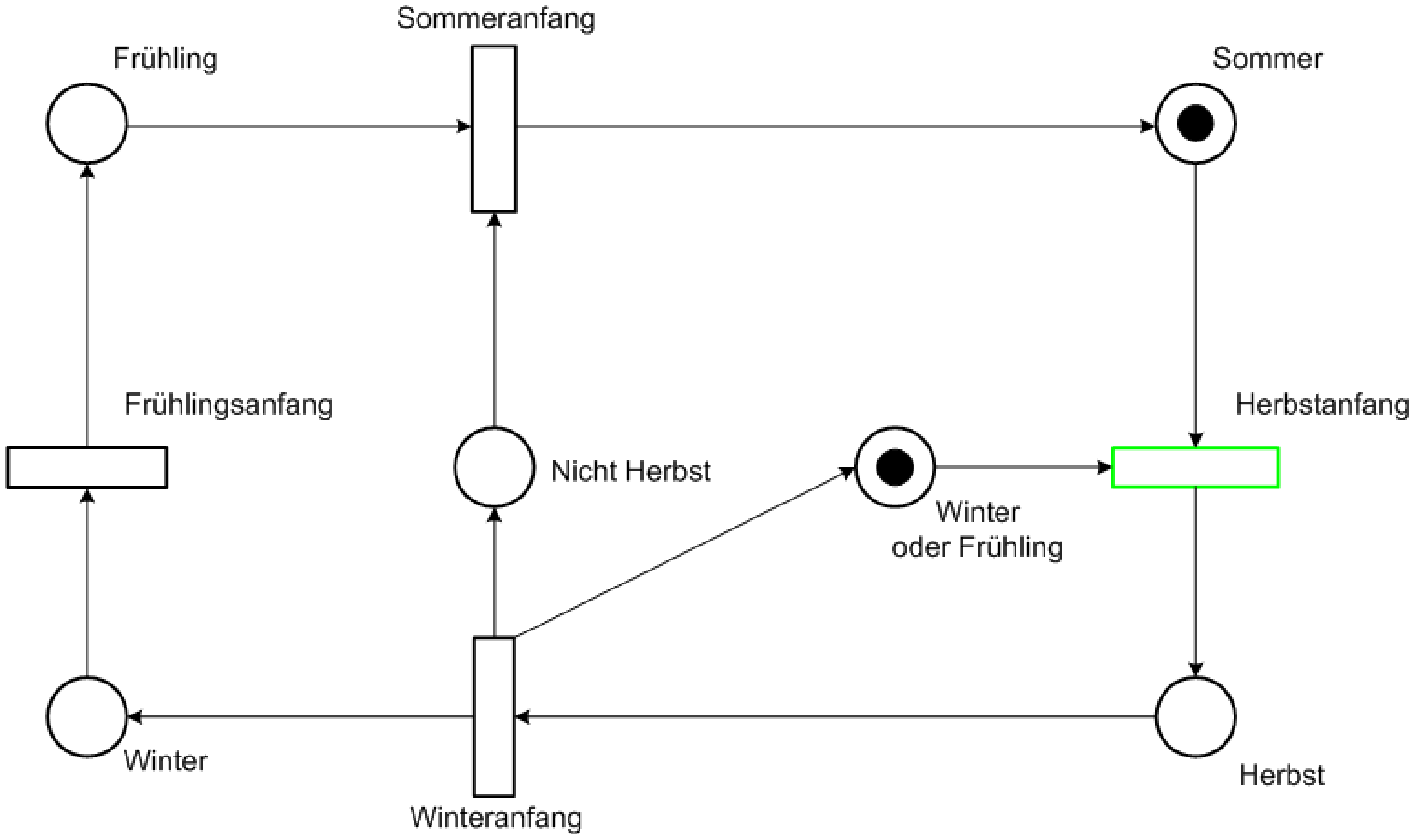
Beispiel – Jahreszeiten



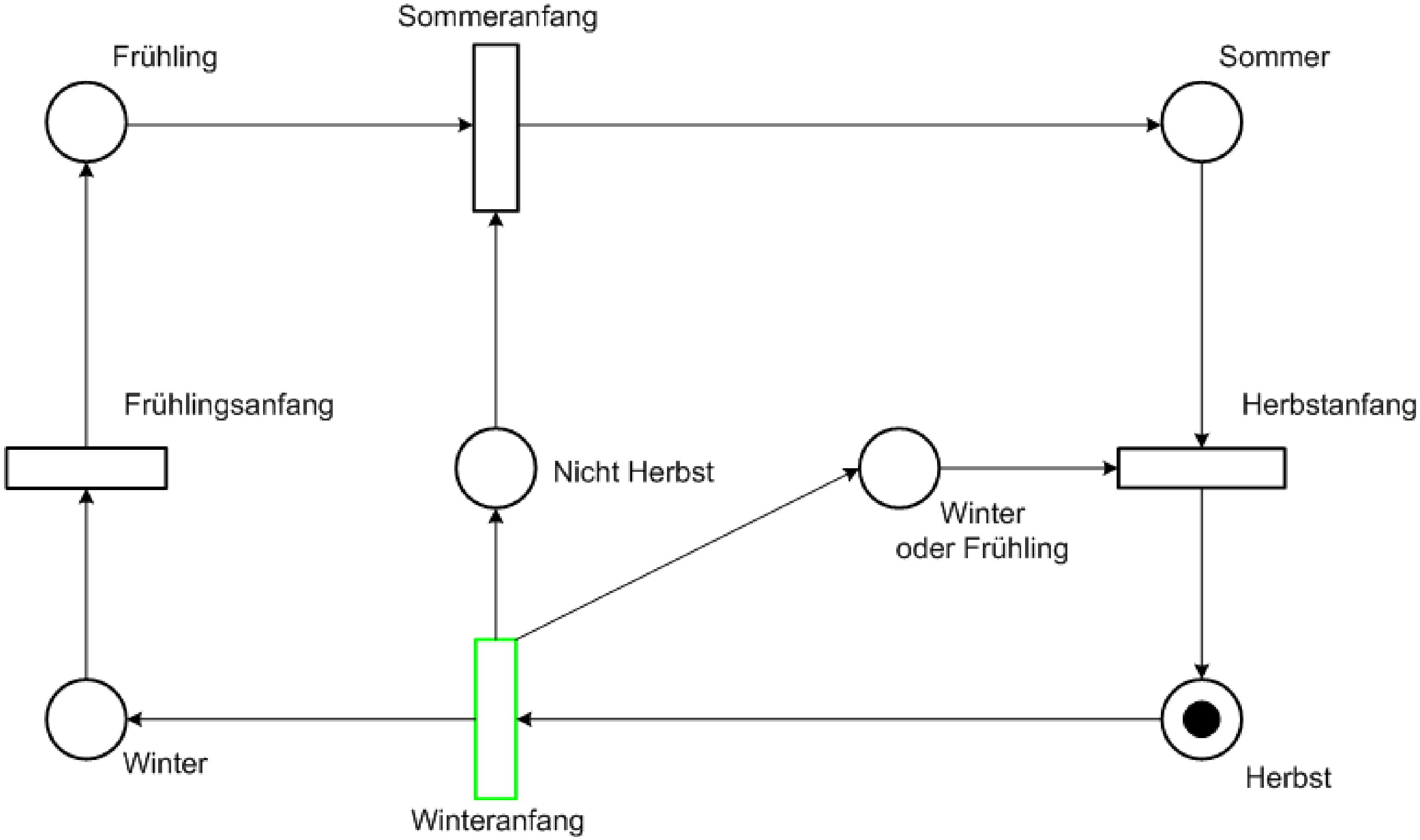
Beispiel – Jahreszeiten



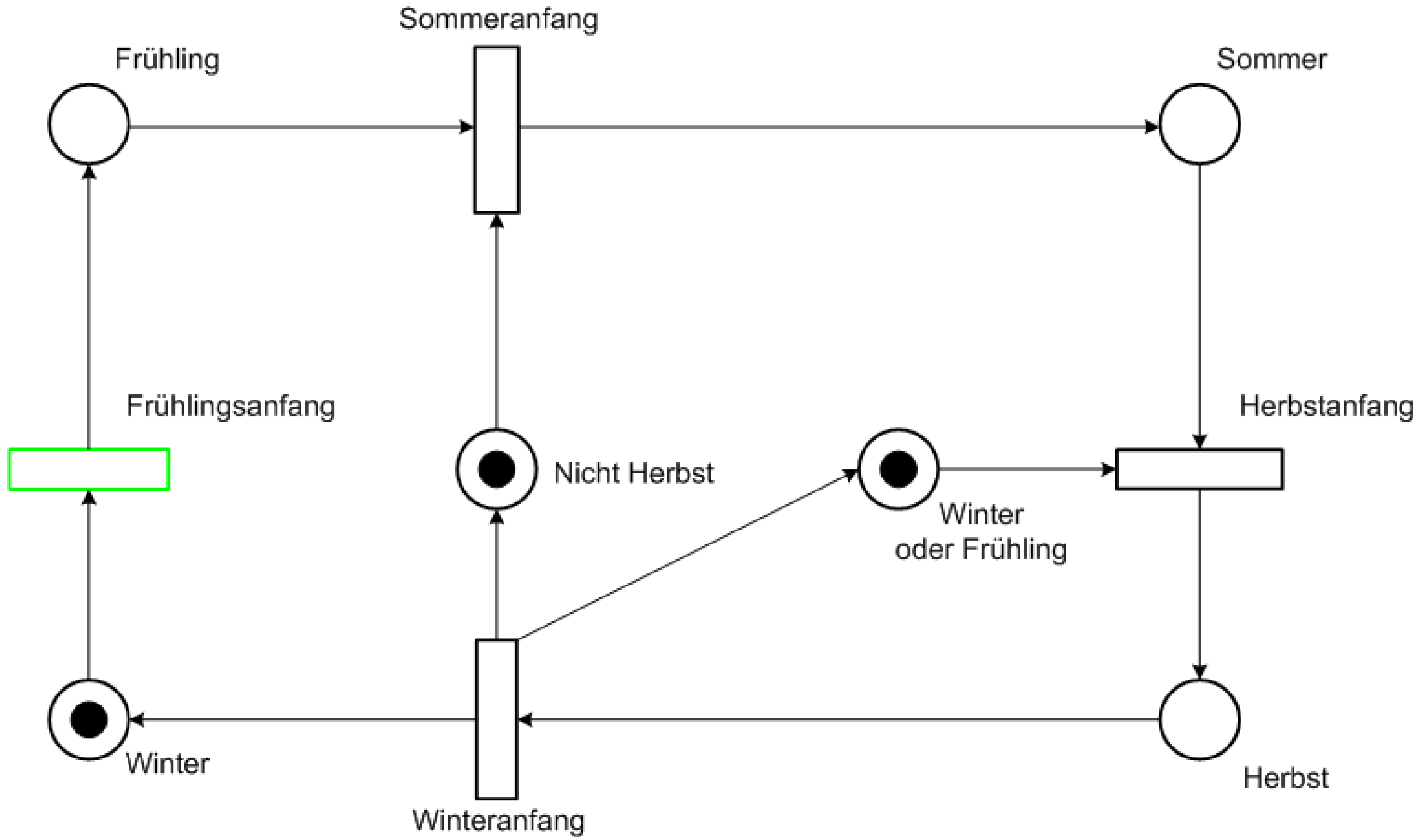
Beispiel – Jahreszeiten



Beispiel – Jahreszeiten



Beispiel – Jahreszeiten

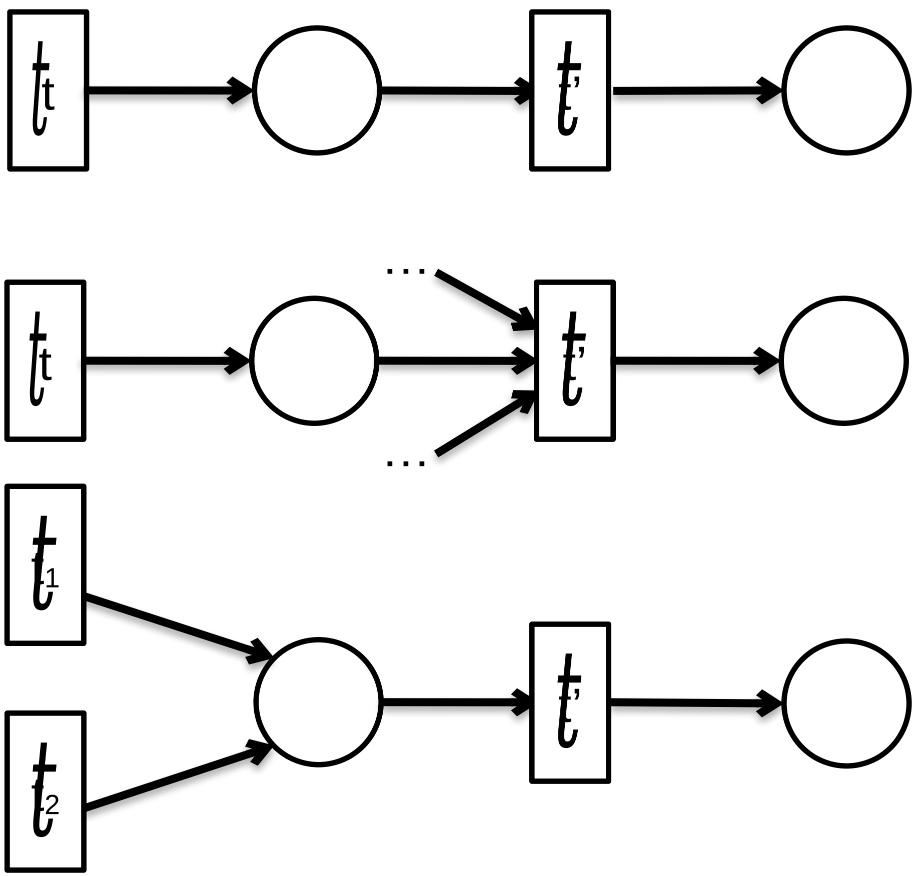


Zusammenfassung Stellen, Marken, Transitionen

- Stellen können markiert oder unmarkiert sein (repräsentierte Bedingung wahr oder falsch, Objekt vorhanden oder nicht vorhanden)
- Markierung gibt Systemzustand an
- Transitionen verändern den Systemzustand, also die Markierung
  - Können nur feuern, wenn Marken im Vorbereich vorhanden und Stellen im Nachbereich frei sind
  - Marken vor der Transition werden vernichtet, Stellen im Nachbereich gefüllt

Grundsituationen – Kausalität

- Kausalität wird durch die Verbindung von Stellen und Transitionen erreicht



Schalten von t führt genau zum Schalten von t'

Schalten von t notwendig für Schalten von t'

Schalten von t_1 oder t_2 hinreichend für Schalten von t'

Grundsituationen – Nebenläufigkeit

- Nebenläufige oder konkurrierende Prozesse können abgebildet werden
- Definition 4

  Sei $N = (S, T, F)$ ein Netz, M und M' Markierungen von N und $G \subseteq T$ eine Menge von Transitionen. Dann heißt G Schritt von M nach M' ($M >_G M'$), falls gilt

  (i) $\forall t \in G$ ist t M-aktiviert,

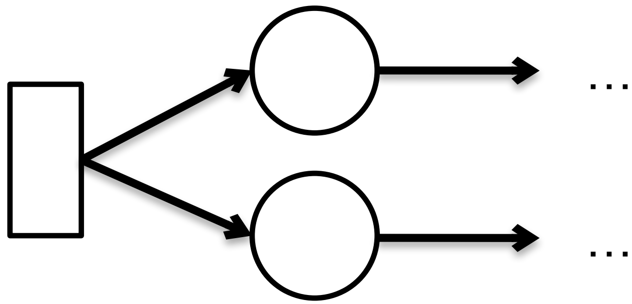
  (ii) $t_1, t_2 \in G \wedge t_1 \neq t_2 \Rightarrow \bullet t_1 \cap \bullet t_2 = \emptyset = t_1 \bullet \cap t_2 \bullet$ und

  (iii) $M' = (M \setminus \bullet G) \cup G\bullet$ mit $\bullet G = \bigcup_{t \in G} \bullet t$ und $G\bullet$ analog.

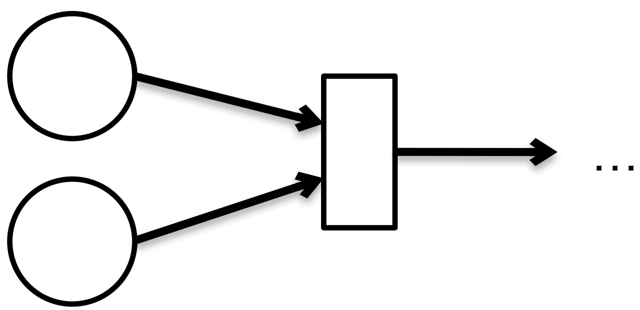
- Schaltreihenfolge der Transitionen aus G unerheblich, können also „gleichzeitig“ feuern => Parallelität
- G ist nebenläufig aktiviert

Grundsituationen – Nebenläufigkeit Fortsetzung

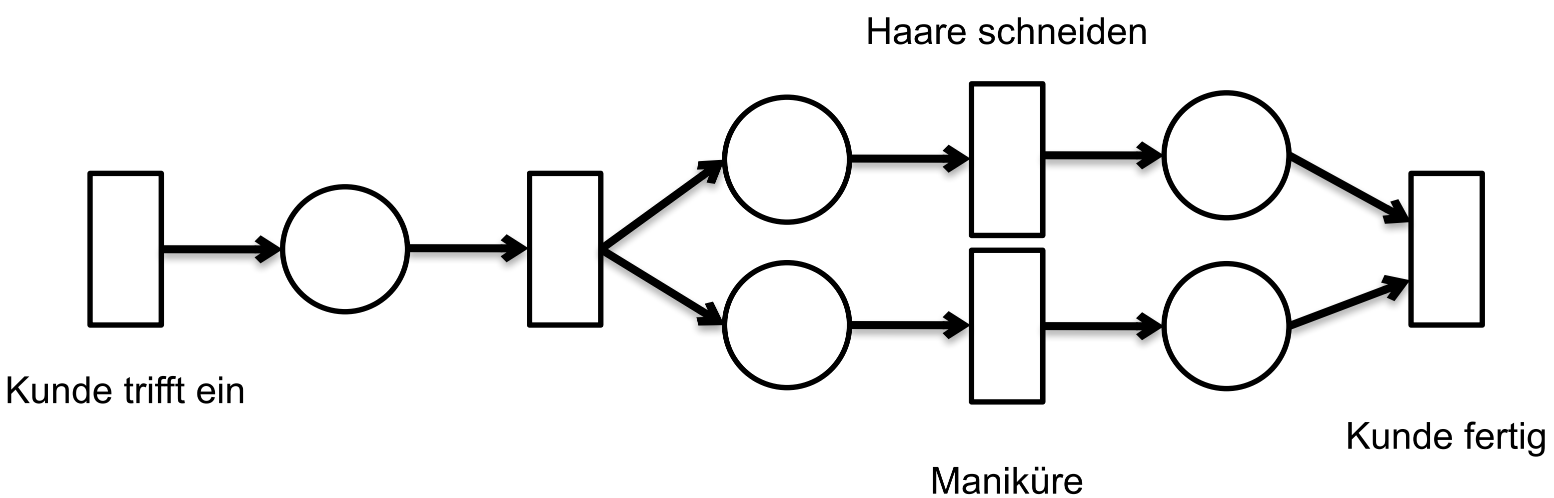
- Nebenläufige Prozesse müssen synchronisiert werden
- *Fork* – Aufspaltung von Prozessen



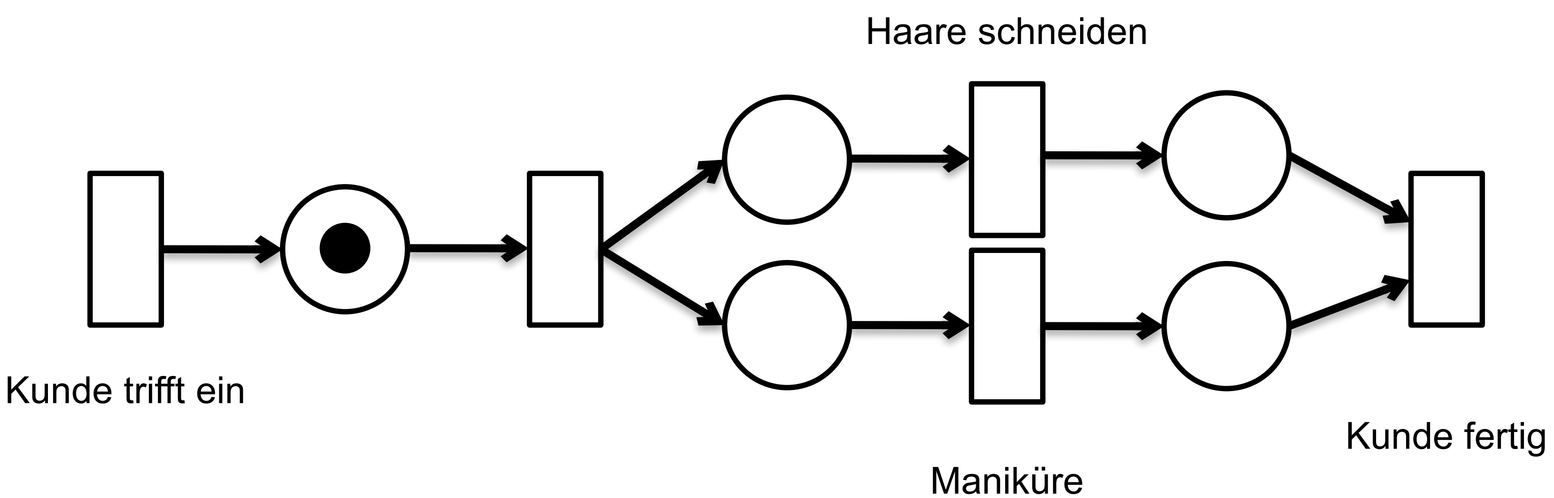
- *Join* – Zusammenführung von Prozessen



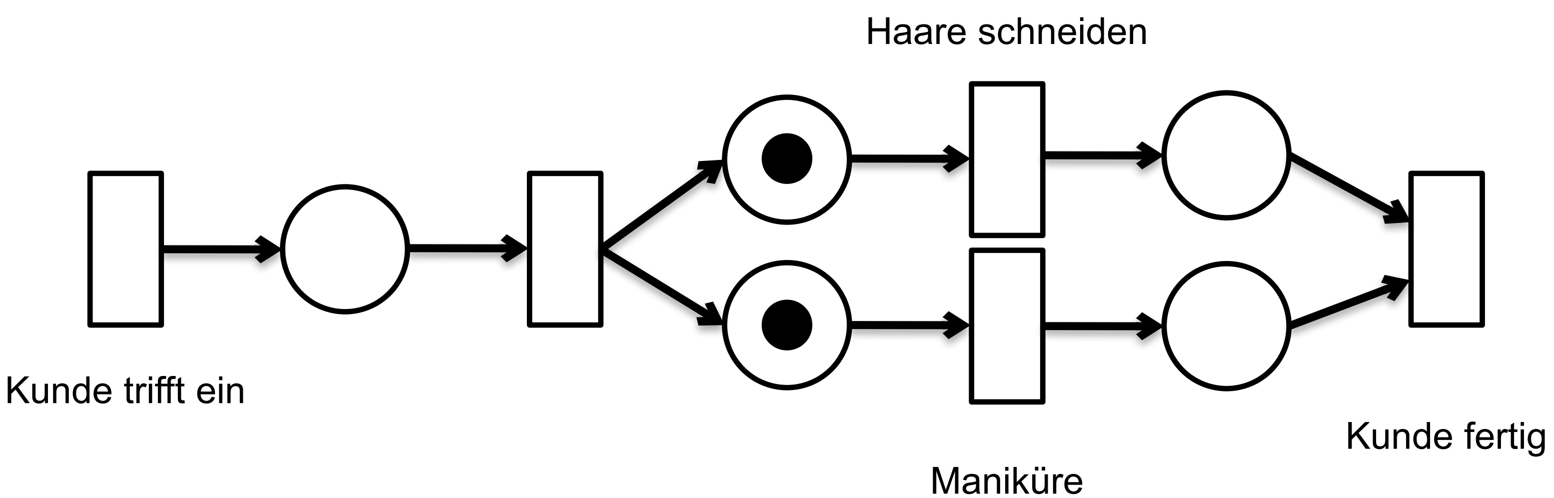
Nebenläufigkeit Beispiel – Der Friseursalon



Nebenläufigkeit Beispiel – Der Friseursalon

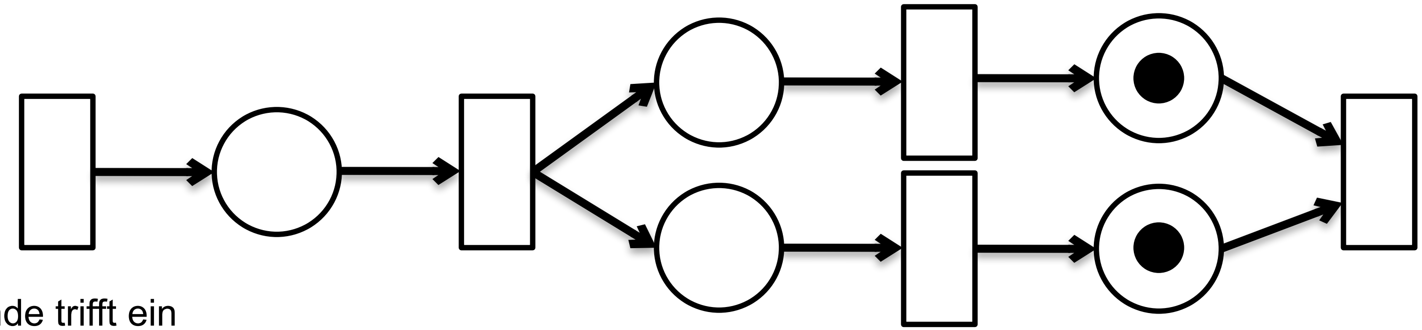


Nebenläufigkeit Beispiel – Der Friseursalon

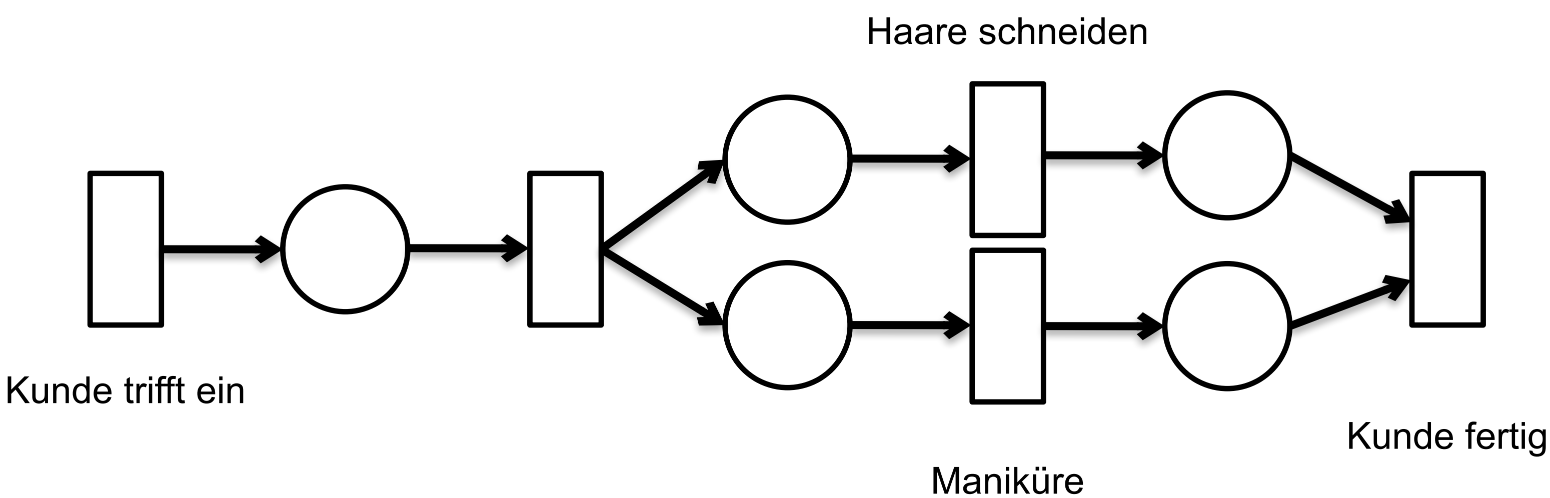


„Haare schneiden“ und „Maniküre“ sind nebenläufig aktiviert

Nebenläufigkeit Beispiel – Der Friseursalon



Nebenläufigkeit Beispiel – Der Friseursalon

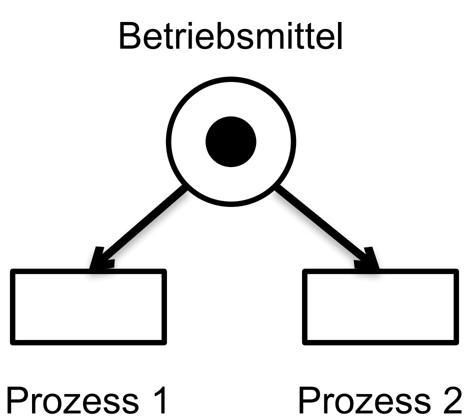


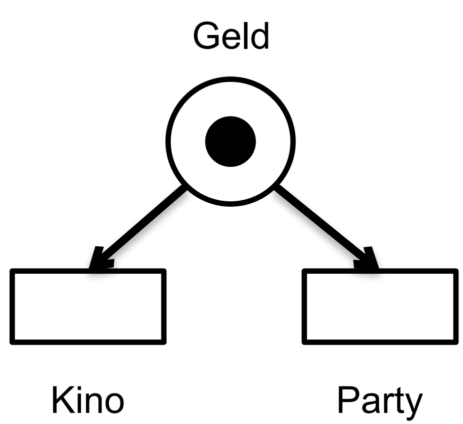
Grundsituationen – Konflikt

- Definition 5

Gilt für einen Fall c und zwei c-aktivierte Ereignisse e_1, e_2: $\bullet e_1 \cap \bullet e_2 \neq \emptyset$, dann liegt eine Konfliktsituation vor.

- Ereignisse benötigen dieselben Ressourcen/Bedingungen
- Können nur alternativ eintreten ⇒ Schaltung nicht deterministisch



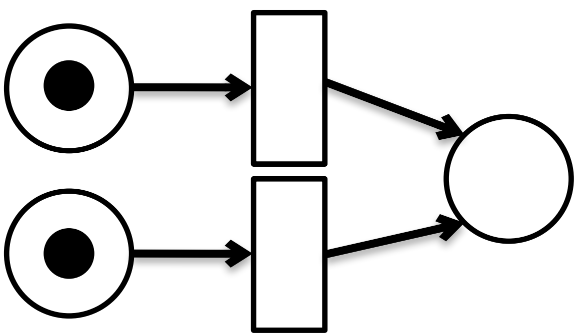


Grundsituationen – Kontakt

- Definition 6

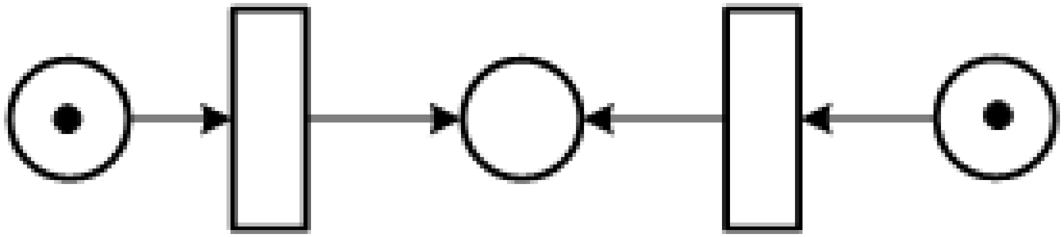
Gilt für einen Fall c: $\bullet e \subseteq c \wedge e \bullet \cap c \neq \emptyset$, dann liegt eine Kontaktsituation vor.

- Nachbereich eines Ereignisses ist nicht frei
- Schaltreihenfolge der aktivierten Transitionen nicht beliebig



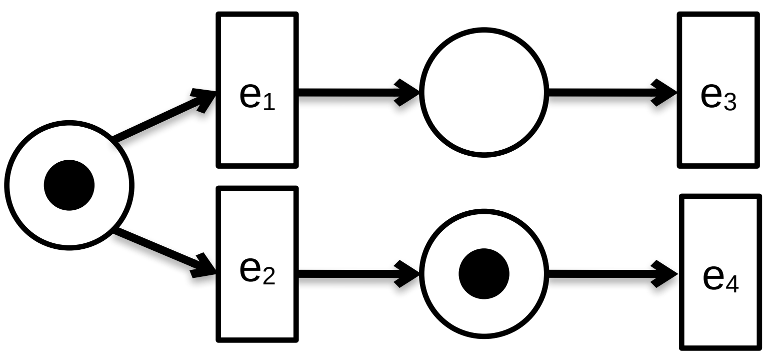
Konflikt und Kontakt

- Konflikt/Kontakt
  - Ereignisse, die nur alternativ zueinander eintreten können, weil sie gemeinsame Vor- oder Nachbedingungen besitzen, stehen in einem Konflikt/Kontakt zueinander
  - Kein Konflikt/Kontakt besteht, wenn zu einem bestimmten Zeitpunkt mehrere Transitionen schalten können, aber keine gemeinsamen Vor- und Nachbedingungen bestehen



Grundsituationen – Konfusion

- Entscheidet die Reihenfolge eintretender Ereignisse, ob ein Konflikt bzw. ein Kontakt entsteht oder nicht, liegt eine Konfusion vor
- Beispiel:



Bedingungs–Ereignis–Netze

- Stellen sind Bedingungen, Transitionen sind Ereignisse
- Definition 7

Ein Quadrupel $\Sigma = (B, E, F, C)$ heißt Bedingungs-Ereignis-Netz, falls gilt:

(i) (B, E, F) ist ein schlichtes Netz ohne isolierte Elemente.

(ii) $C \subseteq P(B)$ ist eine Äquivalenzklasse der Erreichbarkeitsrelation $R_\Sigma = (r_\Sigma \cup r_\Sigma^{-1})^*$. Dabei ist $r_\Sigma \subseteq P(B) \times P(B)$ und $c_1 \; r_\Sigma \; c_2 \Leftrightarrow \exists G \subseteq E$ mit $c_1 >_G c_2$. C heißt Fallklasse von Σ.

(iii) $\forall e \in E \; \exists c \in C$, so dass e c-aktiviert ist.

- C enthält also alle erreichbaren Markierungen
- Jedes Ereignis ist unter min. einer Markierung aktiviert

Lebendige B/E-Netze

- Definition 8

Sei $\Sigma = (B, E, F, C)$ ein B/E-Netz.

(i) Ein Fall $c \in C$ heißt tot in Σ, wenn kein Ereignis aus E c-aktiviert ist.

(ii) Ein Ereignis $e \in E$ heißt tot im Fall c, wenn es kein c' mit $c \, r_\Sigma \, c'$ gibt, bei dem e c'-aktiviert ist.

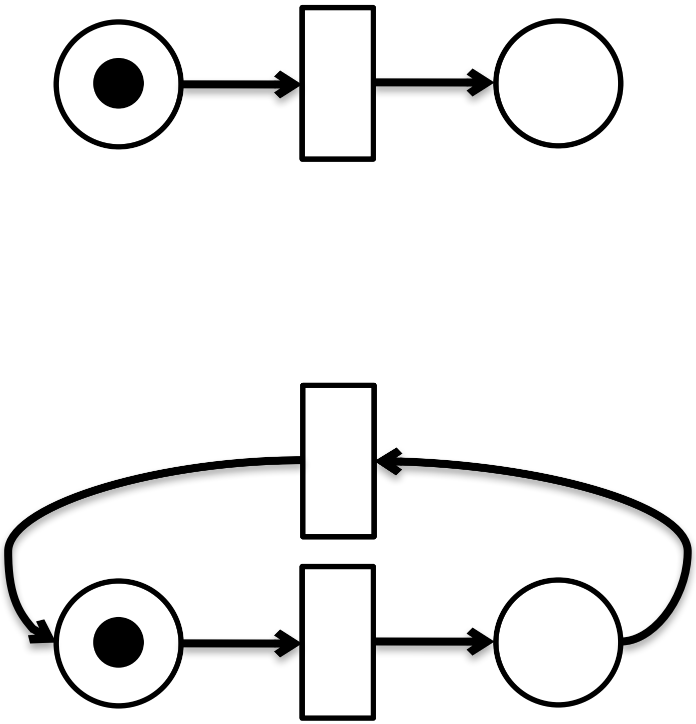
(iii) Ein Ereignis $e \in E$ ist lebendig im Fall c, wenn es in keinem Fall c' mit $c \, r_\Sigma \, c'$ tot ist.

(iv) Ein Fall $c \in C$ wird lebendig genannt, wenn alle Transitionen aus E in c lebendig sind.

(v) Das B/E-Netz Σ heißt lebendig, wenn alle Fälle in C lebendig sind.

- Dies lässt sich auf andere Arten von Petri-Netzen verallgemeinern

Lebendige B/E-Netze: Beispiel



Stellen/Transitionen-Netze

- **Definition 9**

Ein Sextupel $N = (S, T, F, K, W, m_0)$ heißt Stellen/Transitionen-Netz (S/T-Netz) bzw. Petri-Netz, wenn gilt:

(i) (S, T, F) ist ein Netz,
(ii) $K : S \to \mathbb{N}_+ \cup \{\infty\}$ ist die Kapazitätsfunktion,
(iii) $W; F \to \mathbb{N}_+$ ist die Gewichtsfunktion und
(iv) $m_0 : S \to \mathbb{N}_0 \cup \{\infty\}$ ist die Anfangsmarkierung, wobei $\forall s \in S : m_0(s) \leq K(s)$.

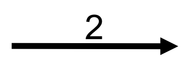
- **Folgerung**

Ein B/E-Netz (B, E, F, C) kann in ein Petri-Netz $N = (B, E, F, K, W, m_0)$ überführt werden mit

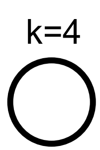
(i) $K : B \to 1$,
(ii) $W : F \to 1$ und
(iii) $m_0(s) = \begin{cases} 1, \text{ falls } s \in c \\ 0, \text{ sonst} \end{cases}$ für ein $c \in C$.

Dimensionen und Kapazitäten

- Dimensionen
  - Beschriftung von Verbindungen
  - Legen die Anzahl von erzeugten bzw. verbrauchten Marken beim Schalten einer Transition fest
  - Wenn keine Angabe dann 1

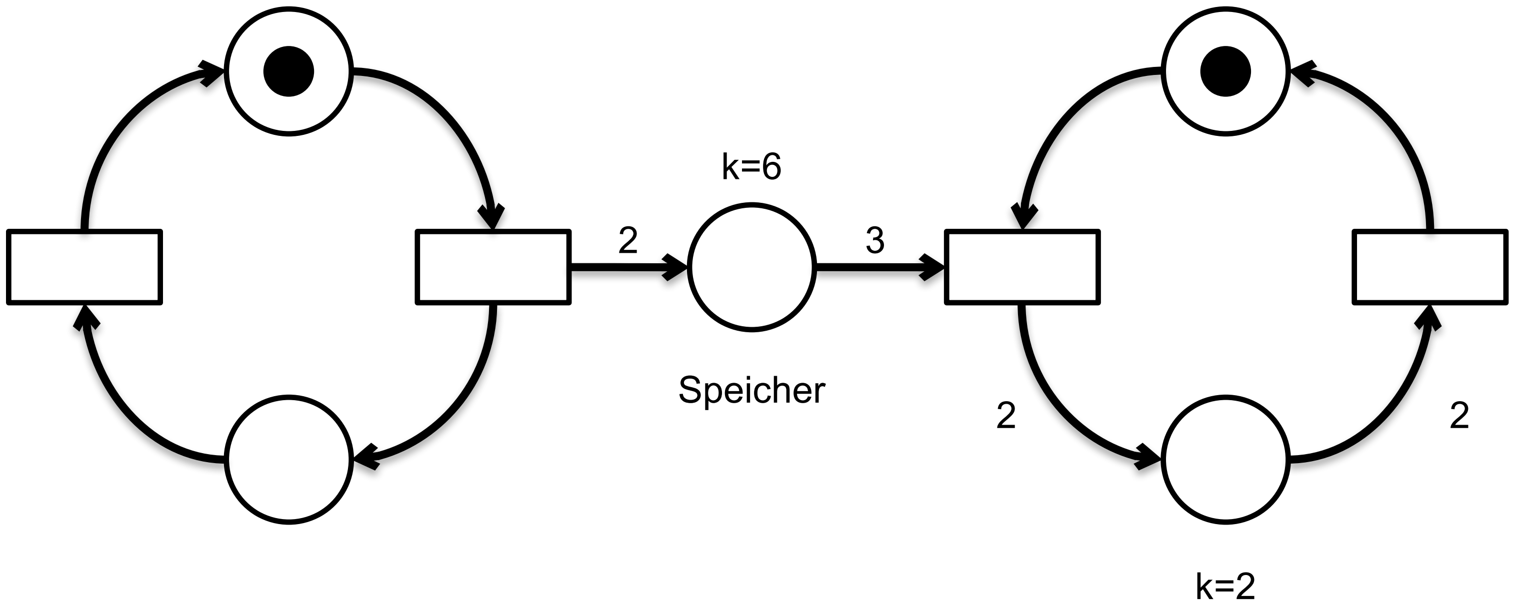


- Kapazitäten
  - Beschriftung von Stellen
  - Legen die Höchstgrenze für die Anzahl von Token in einer Stelle fest
  - Wenn keine Beschriftung, dann k=1



Stellen/Transitionen-Netze: Beispiel

Erzeuger-Verbraucher-System



Stellen/Transitionen-Netze: Schaltregel

- Definition 10

Sei $N = (S, T, F, K, W, m_0)$ ein Petri Netz, m eine Markierung von N und t eine Transition aus T. Dann gilt:

(i) t ist m-aktiviert, wenn für alle $s \in \bullet t$ gilt: $m(s) \geq W(s,t)$ und für alle $s \in t\bullet$ gilt: $m(s) + W(t,s) \leq K(s)$.

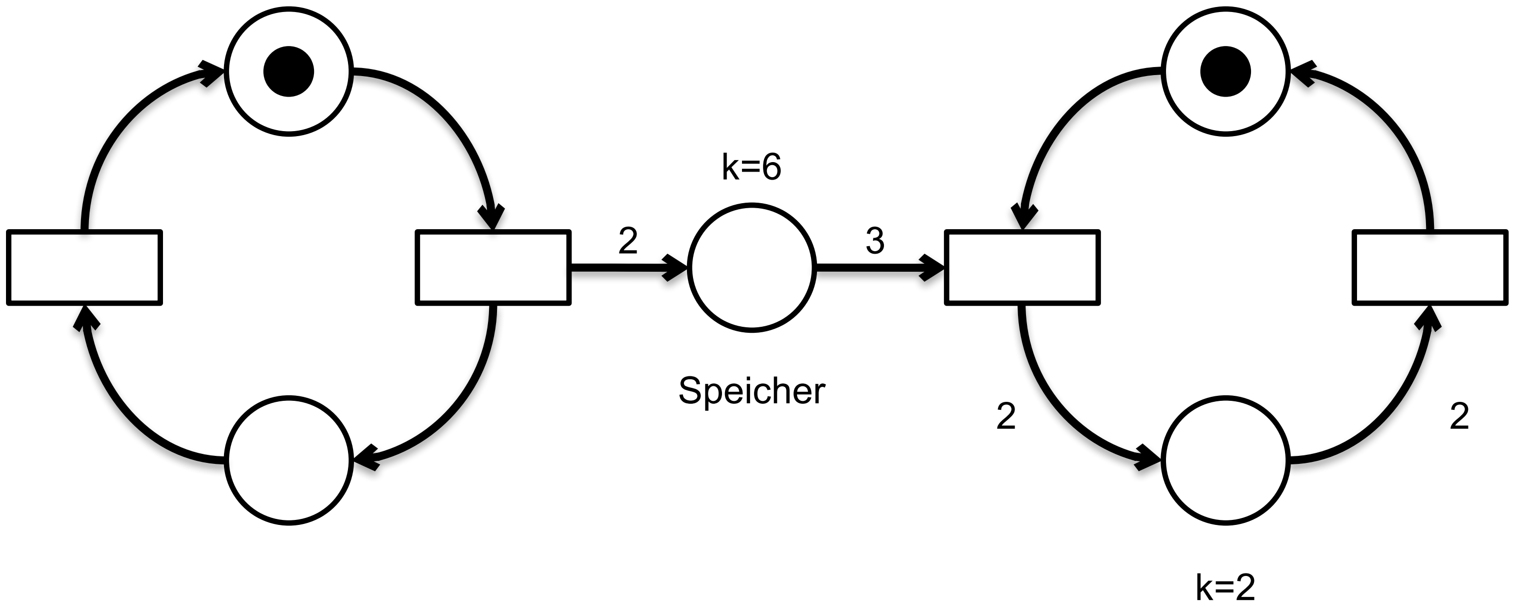
(ii) Ist t m-aktiviert, kann t schalten, wodurch die Markierung m' entsteht, die durch

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

definiert ist.

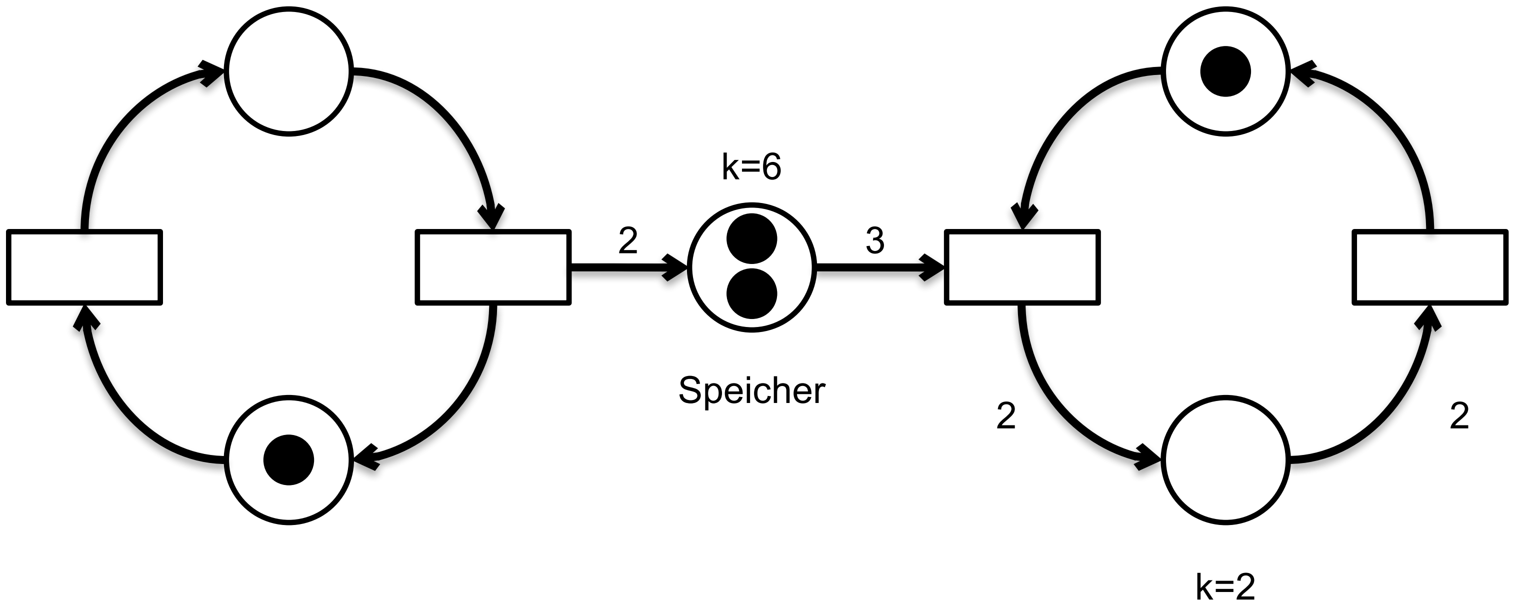
Stellen/Transitionen-Netze: Beispiel

Erzeuger-Verbraucher-System



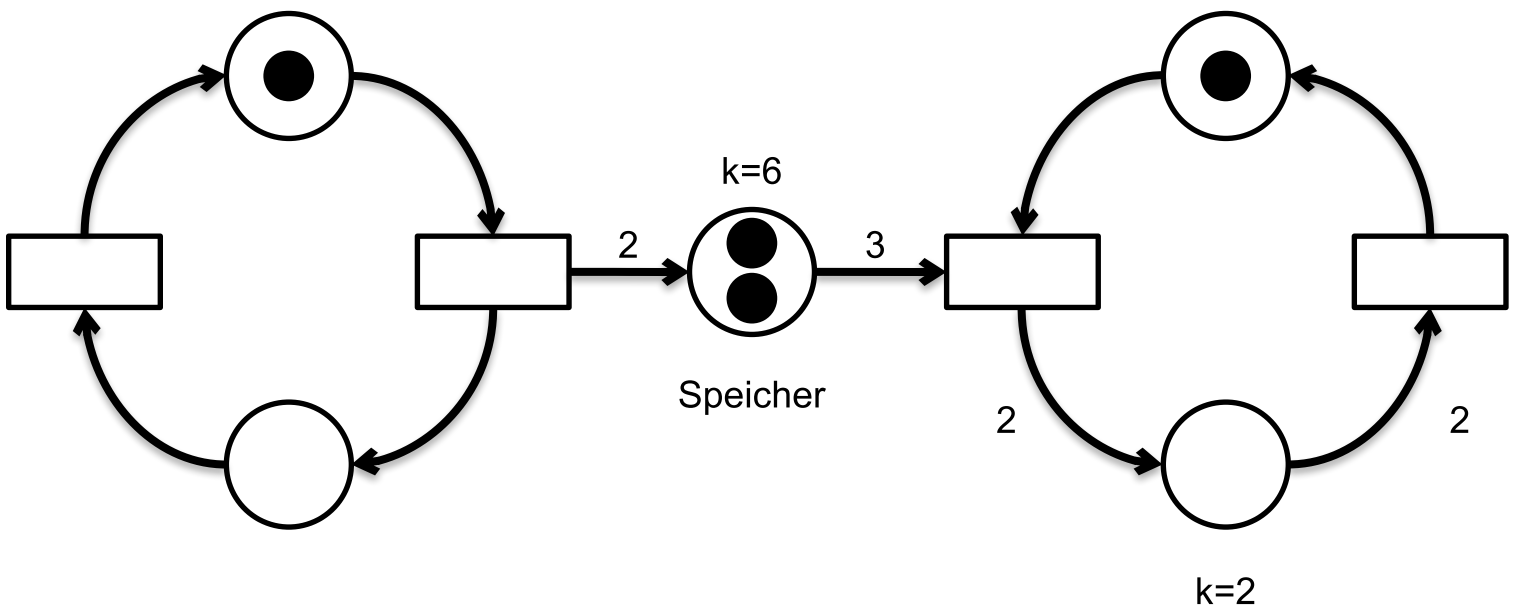
Stellen/Transitionen-Netze: Beispiel

Erzeuger-Verbraucher-System



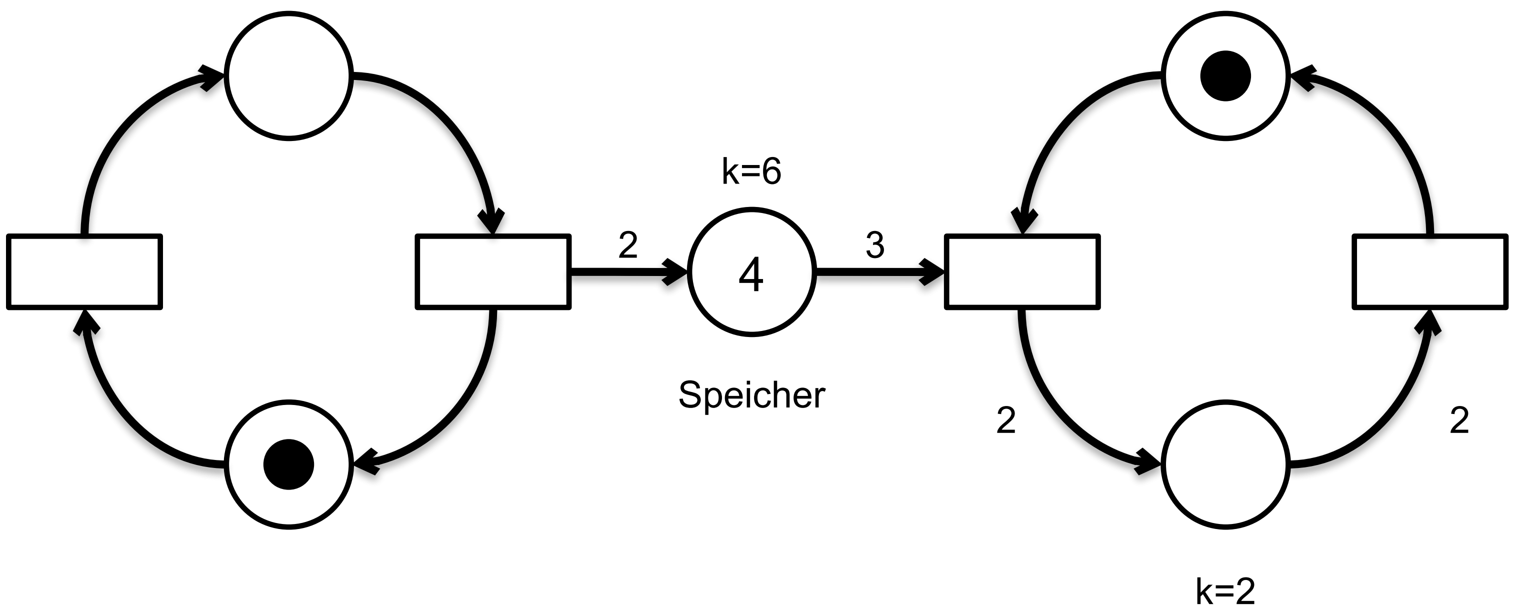
Stellen/Transitionen-Netze: Beispiel

Erzeuger-Verbraucher-System



Stellen/Transitionen-Netze: Beispiel

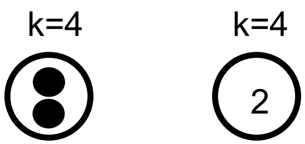
Erzeuger-Verbraucher-System



Für die Übersichtlichkeit können Marken als Zahlen dargestellt werden

Markierung

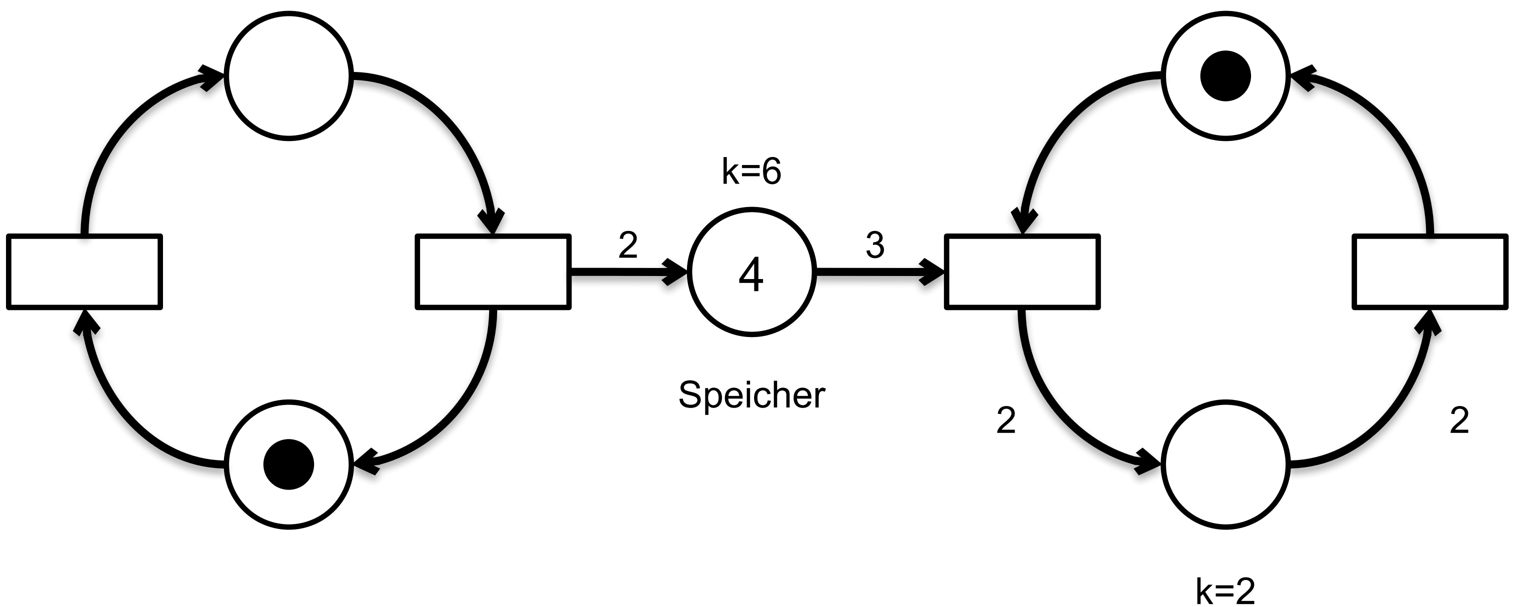
- Markierung von Stellen mit Kapazität
  - Bei wenigen Marken zeichnet man oft einen Punkt je Marke, bei mehreren meist nur die Zahl



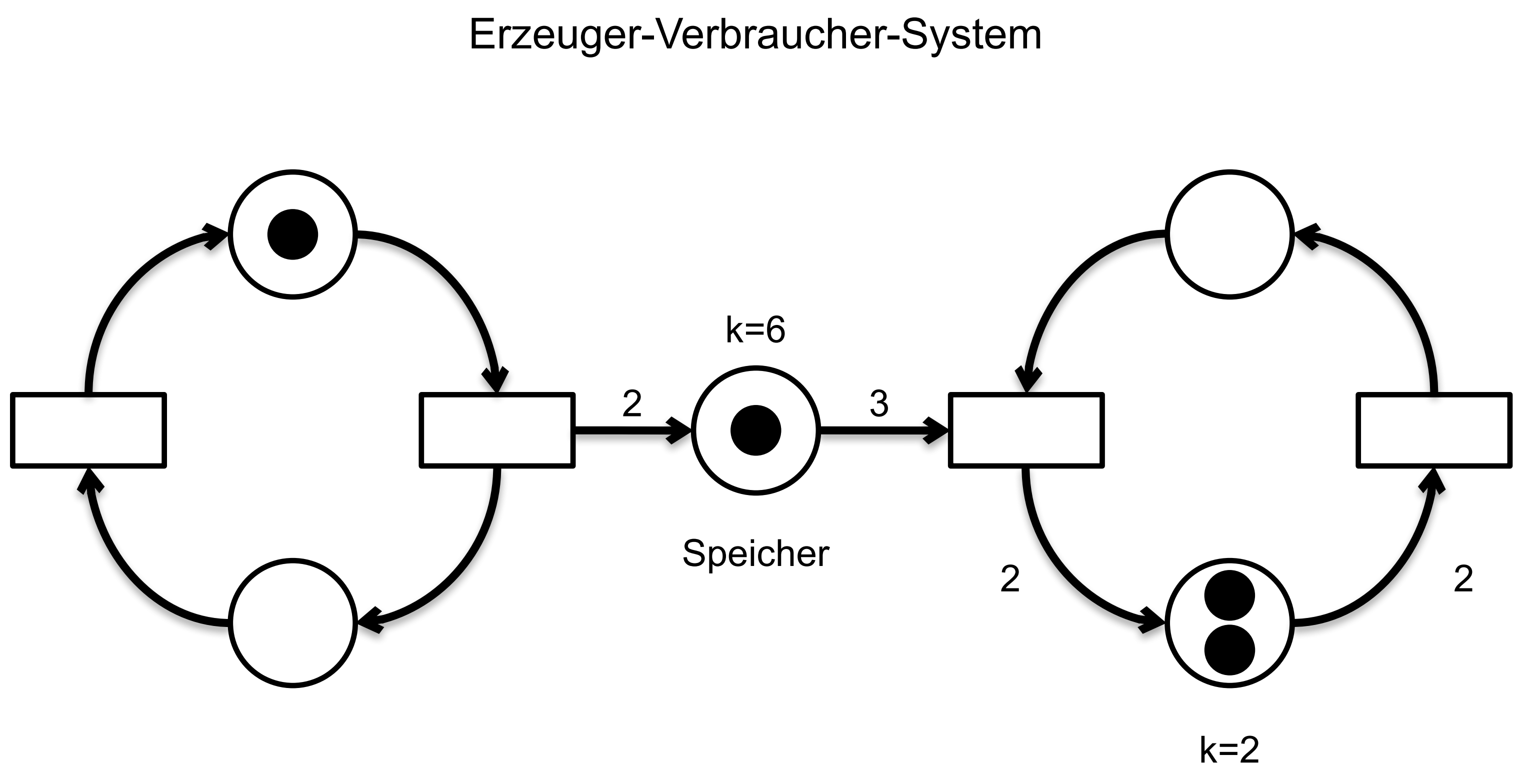
- Zahlreiche andere Erweiterungen existieren
- Notationselemente bei verschiedenen Quellen unterschiedlich dargestellt

Stellen/Transitionen-Netze: Beispiel

Erzeuger-Verbraucher-System



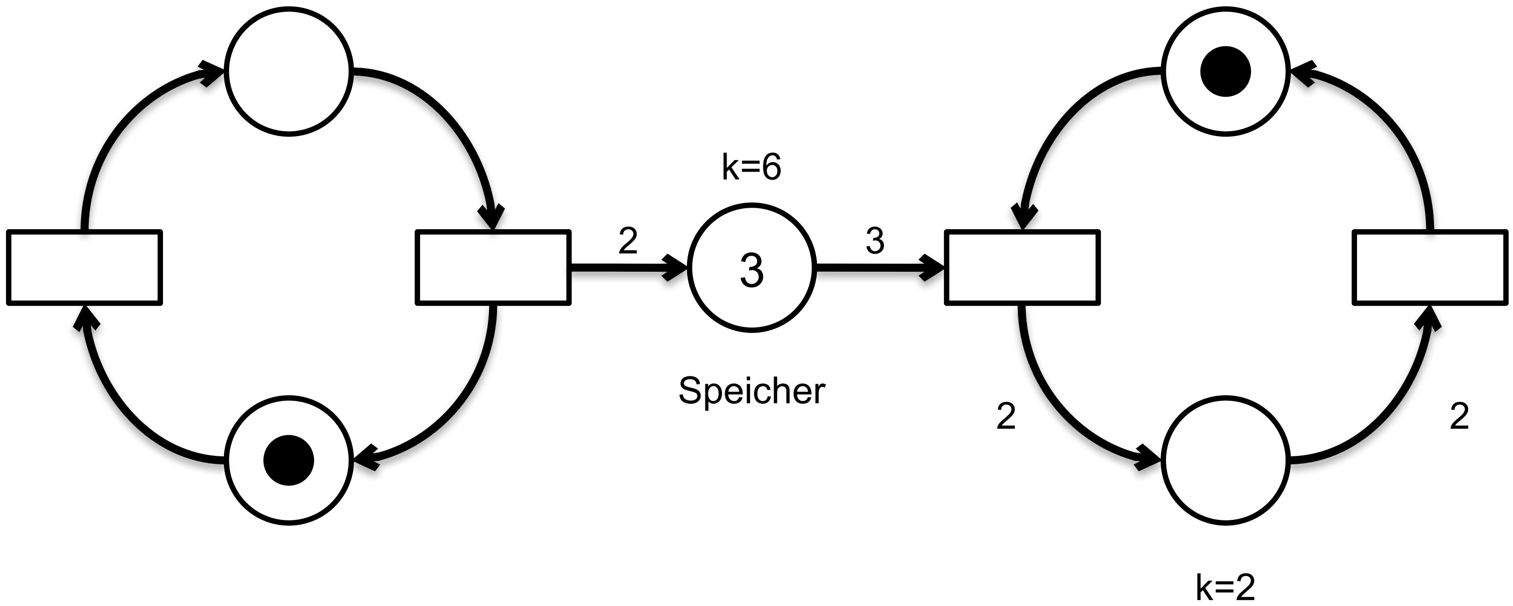
Stellen/Transitionen-Netze: Beispiel



Beide Transitionen waren nebenläufig aktiviert

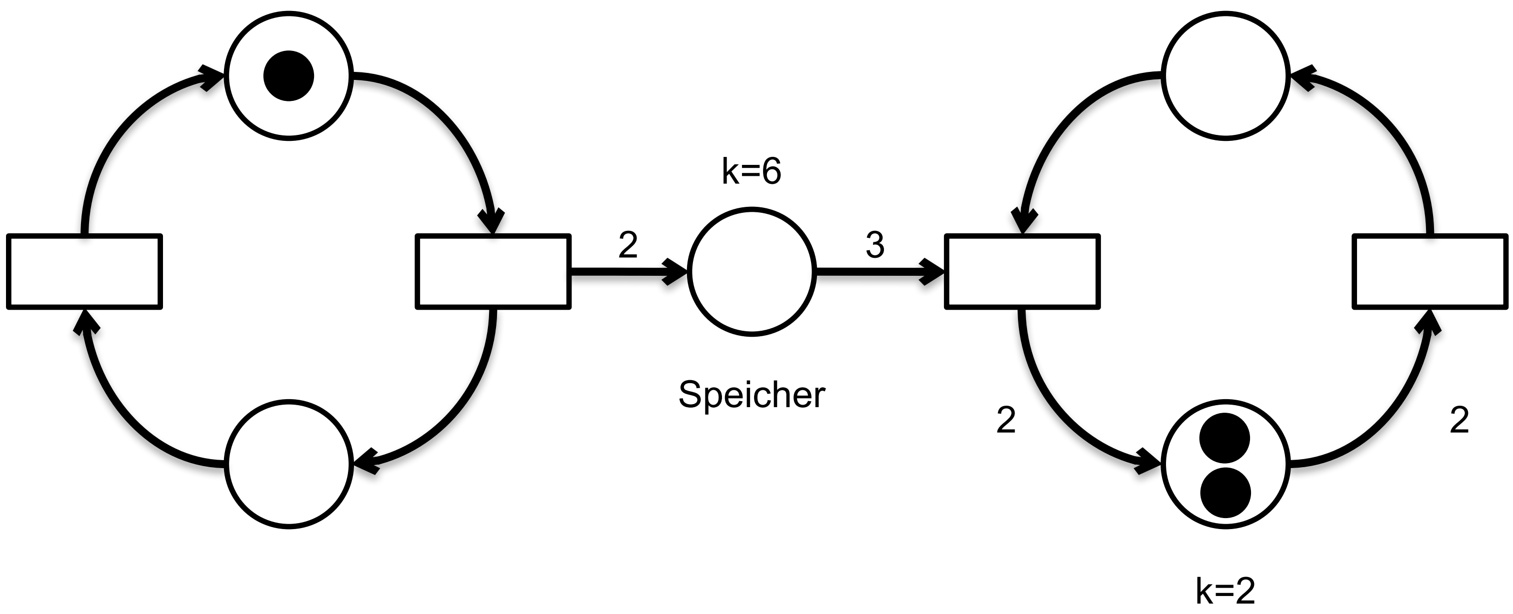
Stellen/Transitionen-Netze: Beispiel

Erzeuger-Verbraucher-System



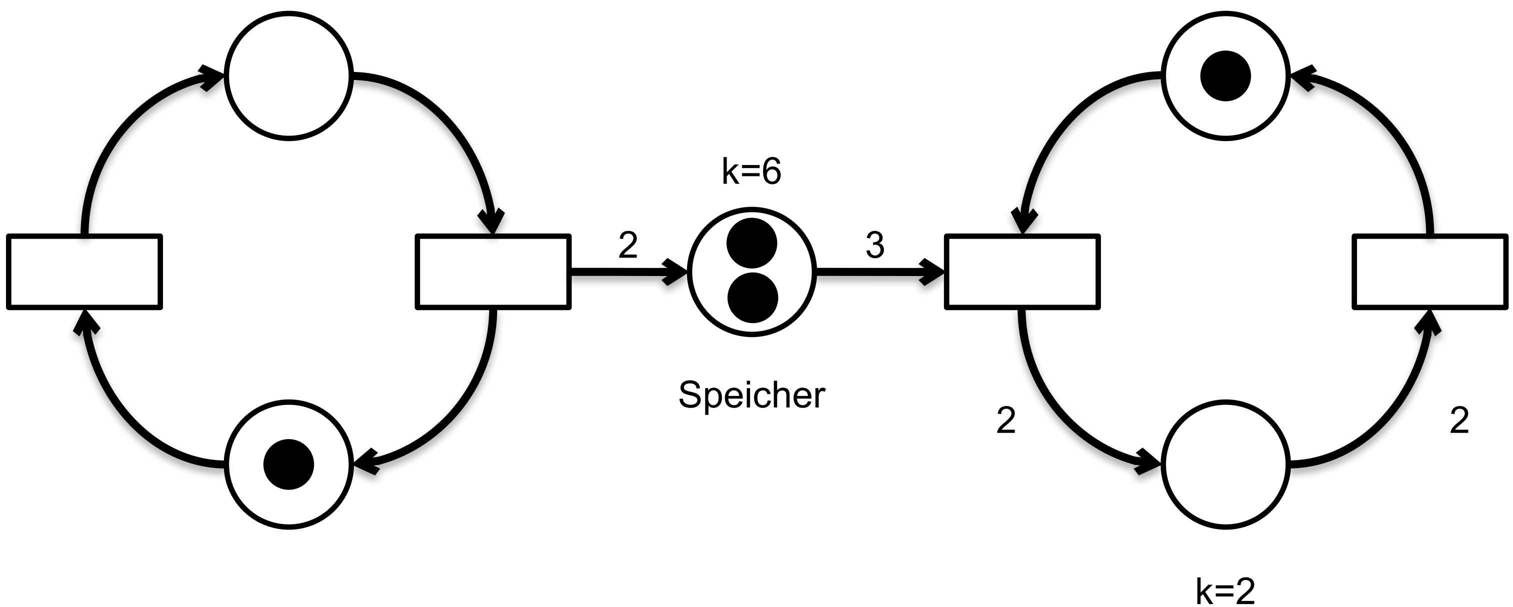
Stellen/Transitionen-Netze: Beispiel

Erzeuger-Verbraucher-System



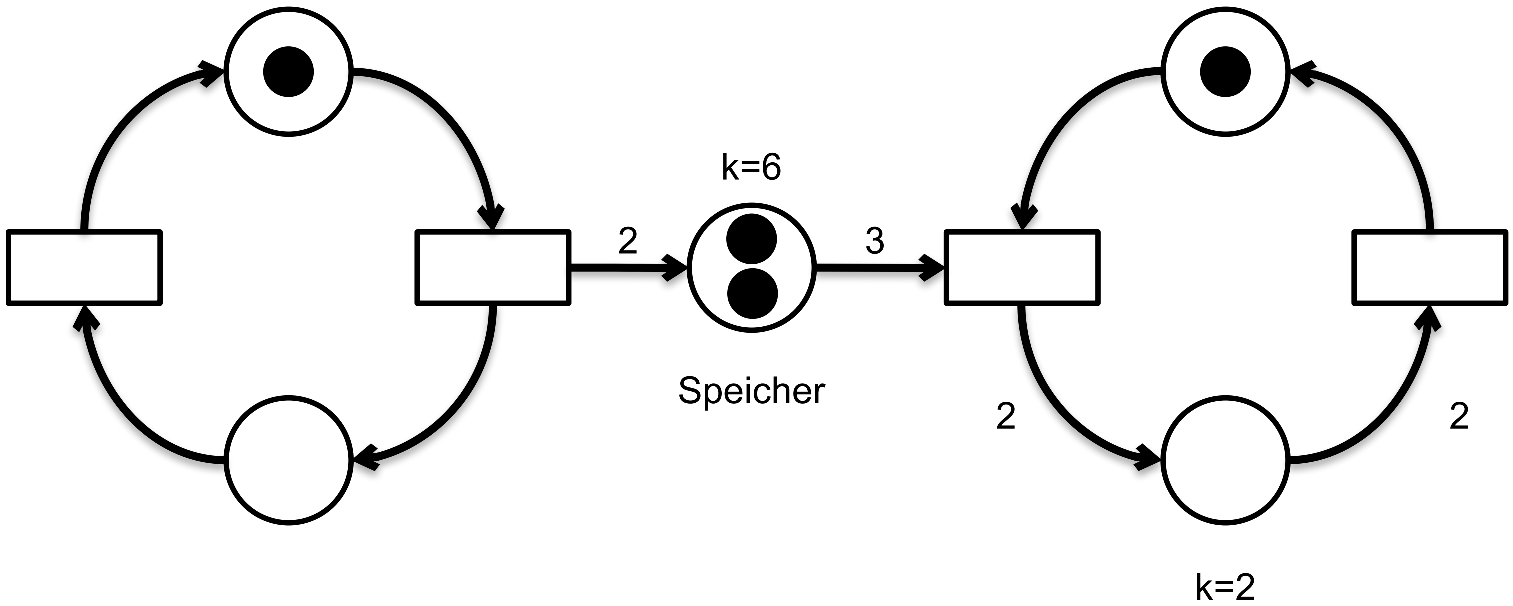
Stellen/Transitionen-Netze: Beispiel

Erzeuger-Verbraucher-System



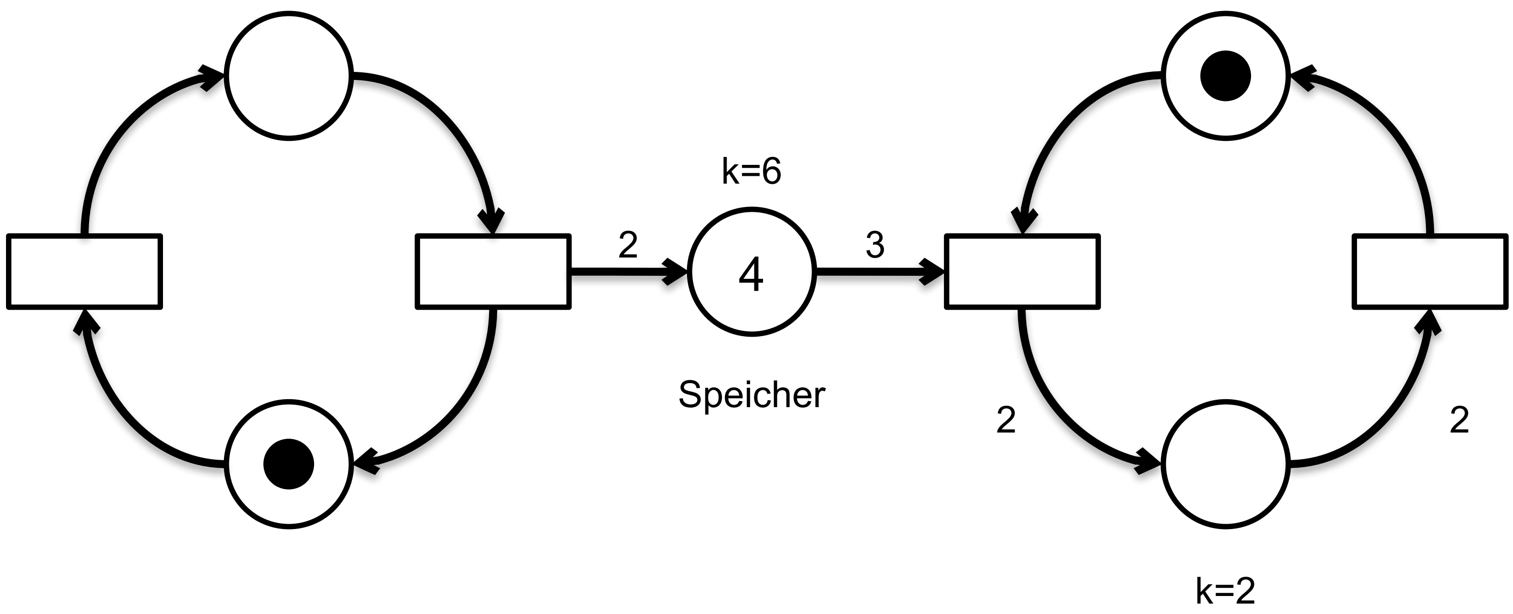
Stellen/Transitionen-Netze: Beispiel

Erzeuger-Verbraucher-System



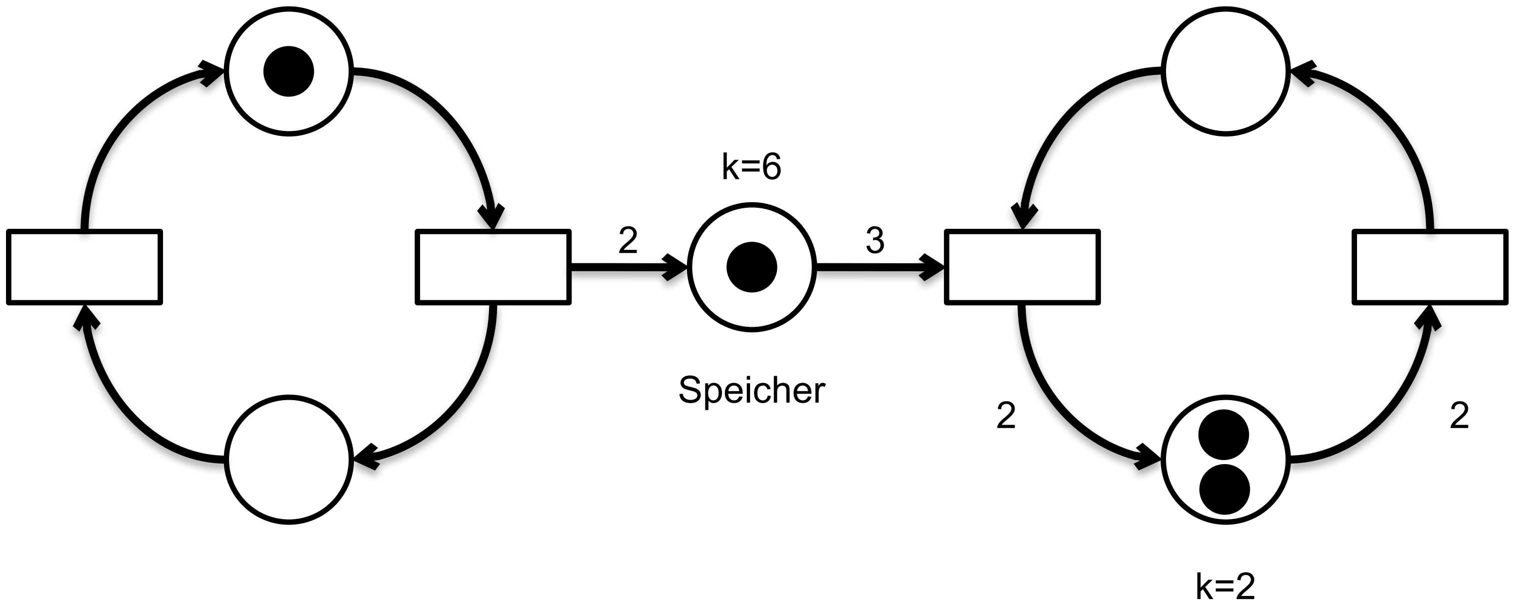
Stellen/Transitionen-Netze: Beispiel

Erzeuger-Verbraucher-System

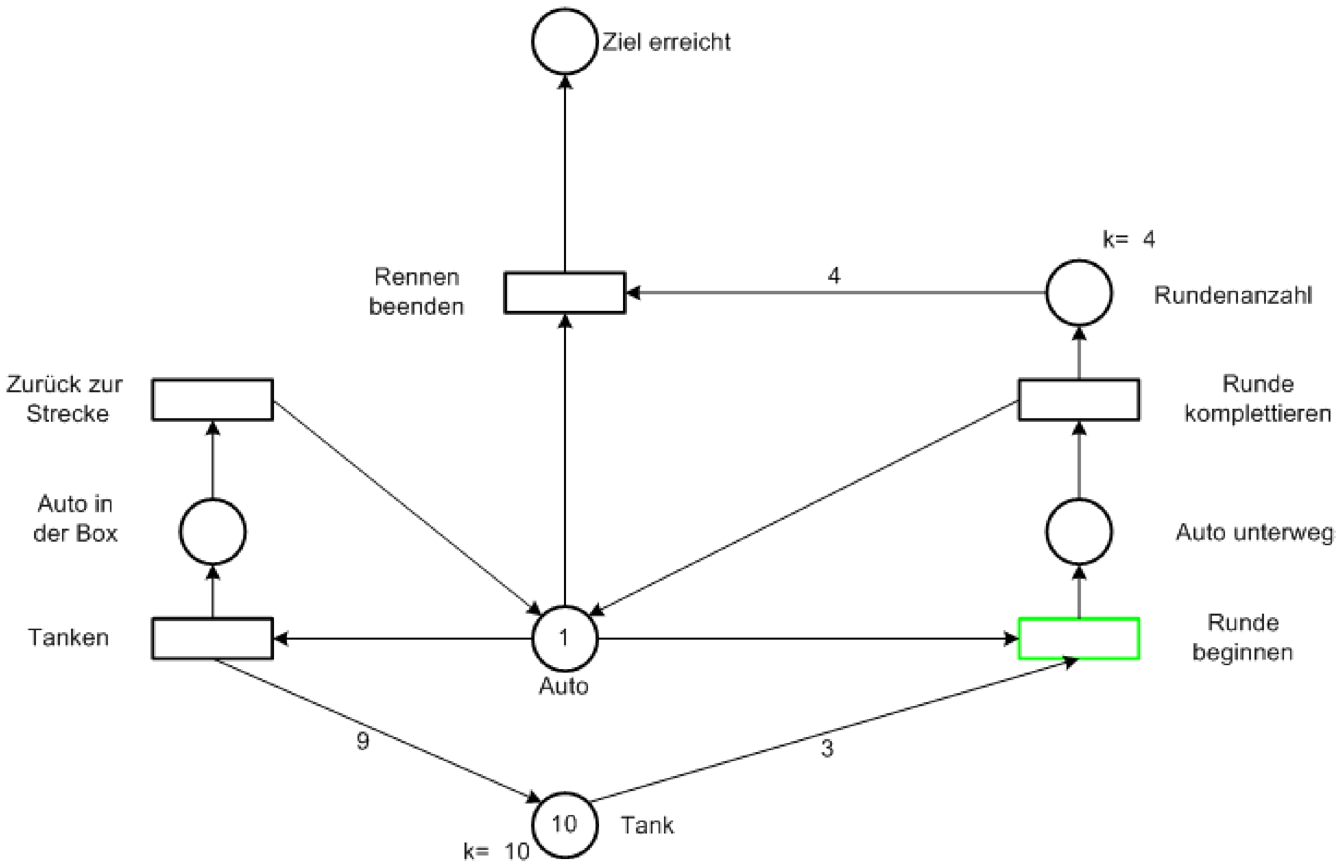


Stellen/Transitionen-Netze: Beispiel

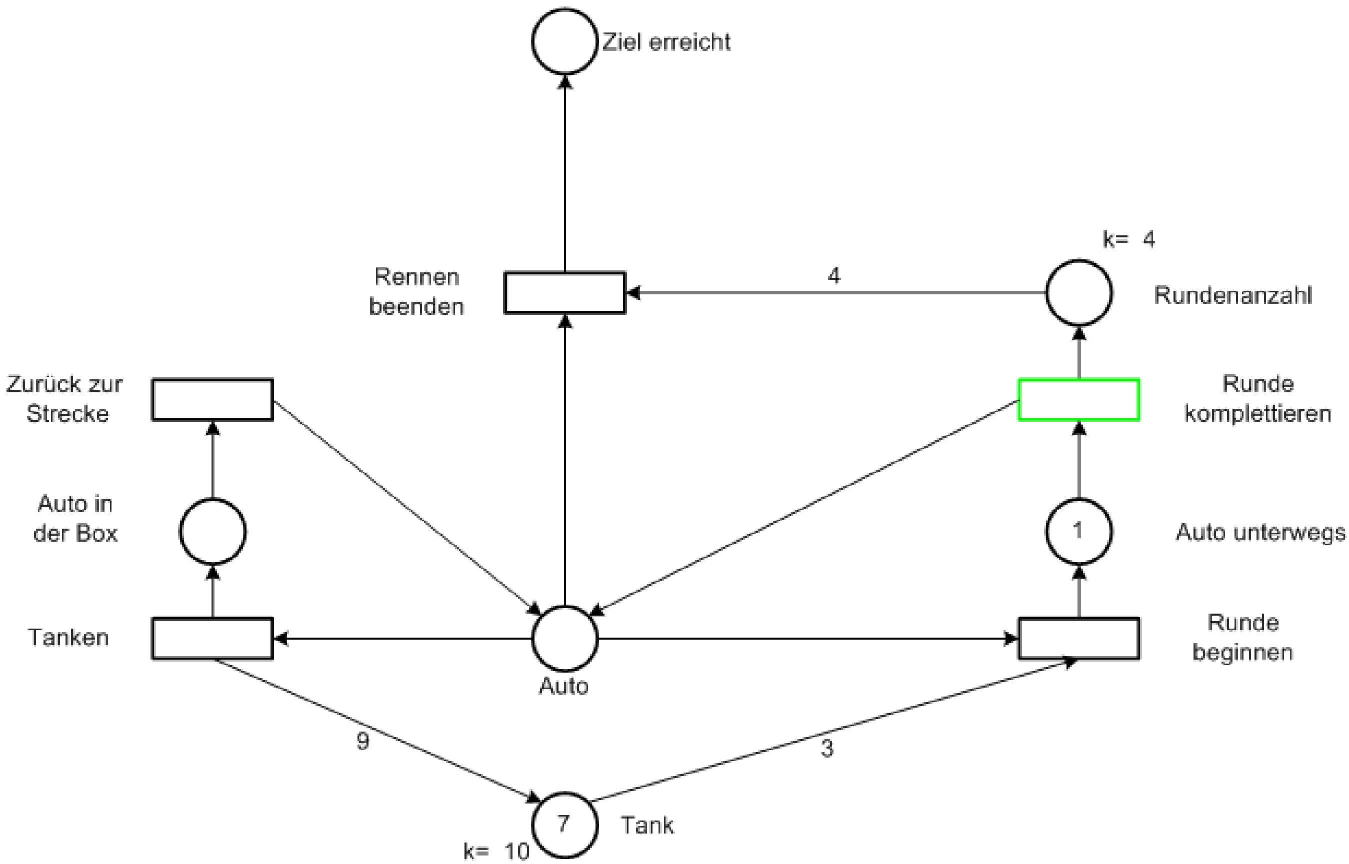
Erzeuger-Verbraucher-System



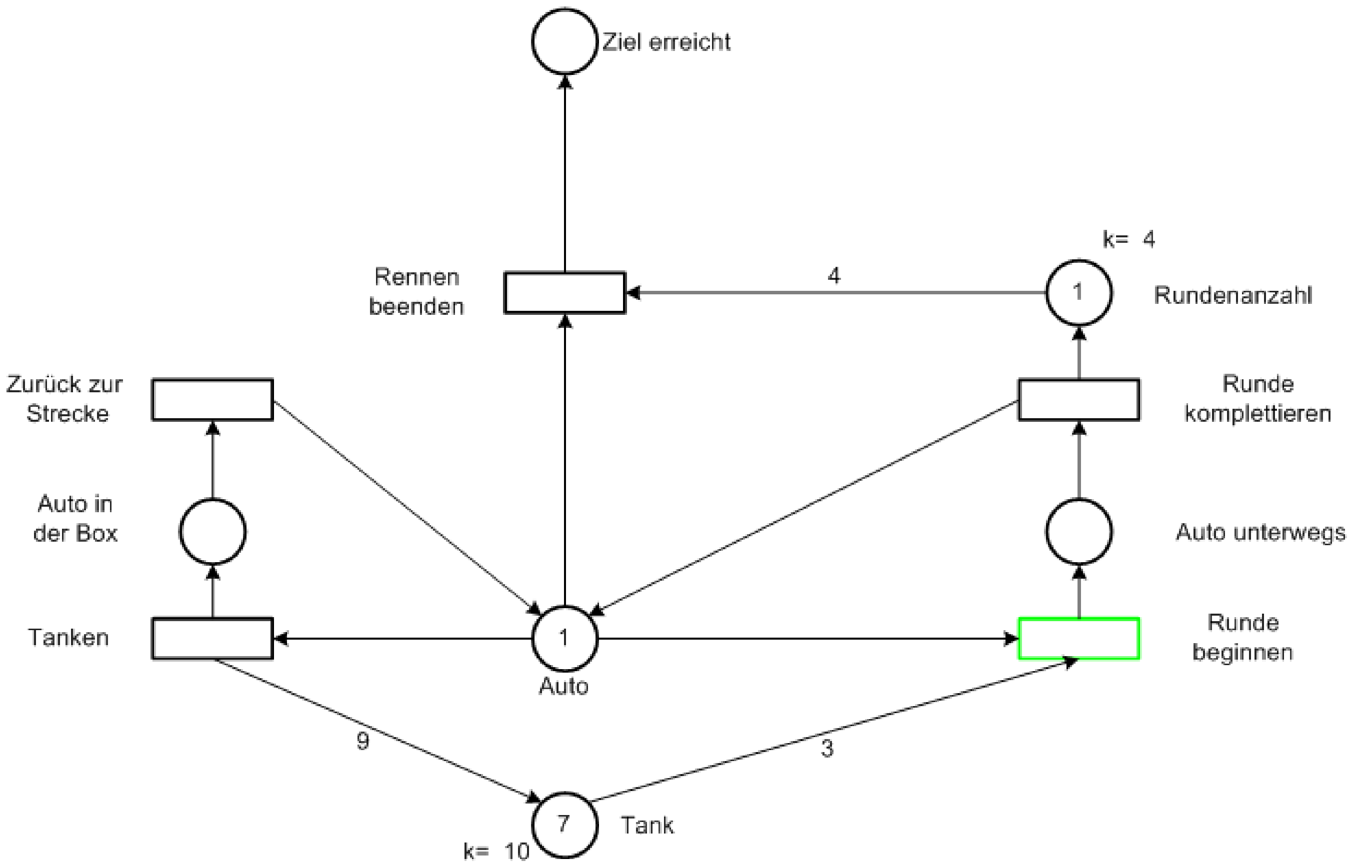
Beispiel – Formel 1



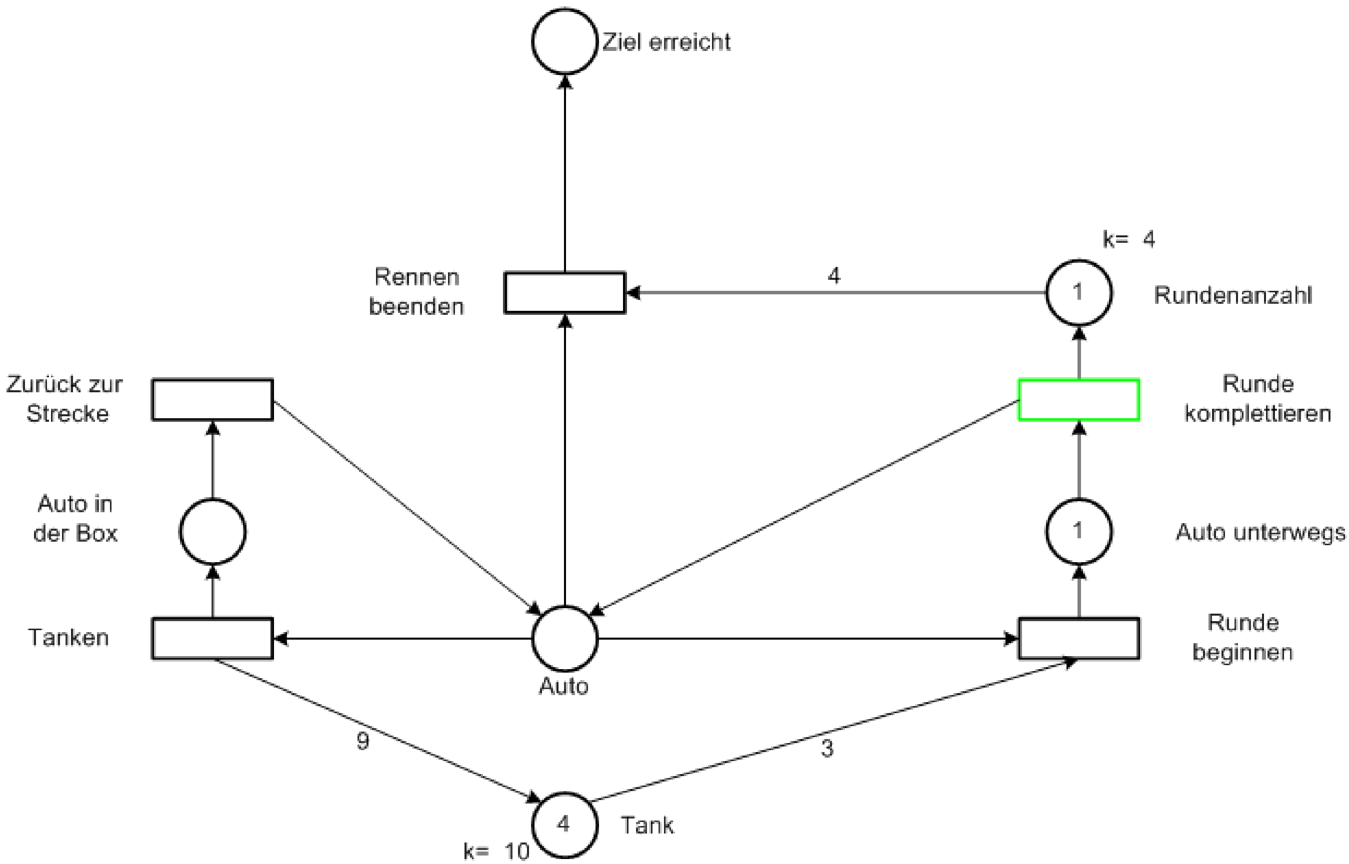
Beispiel – Formel 1



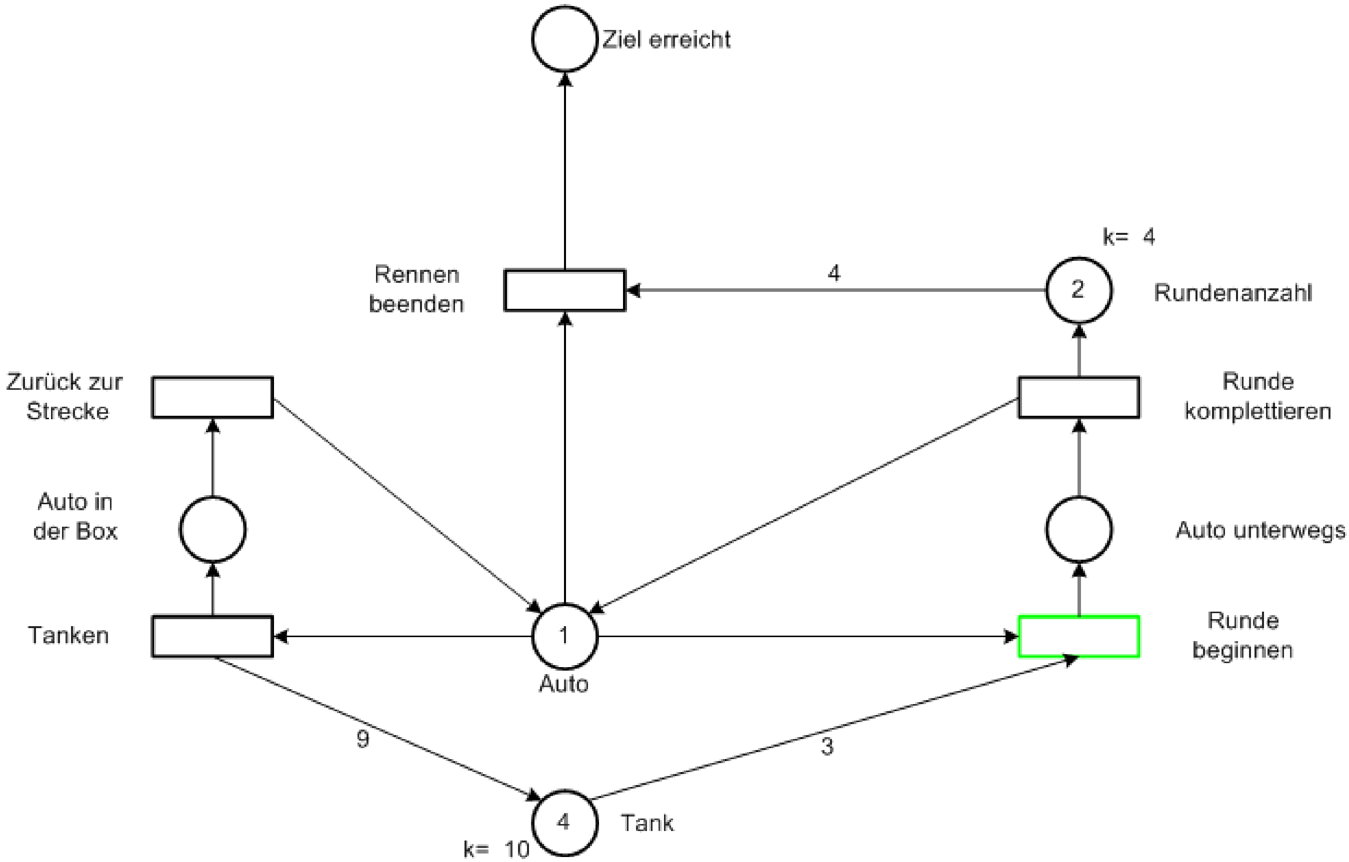
Beispiel – Formel 1



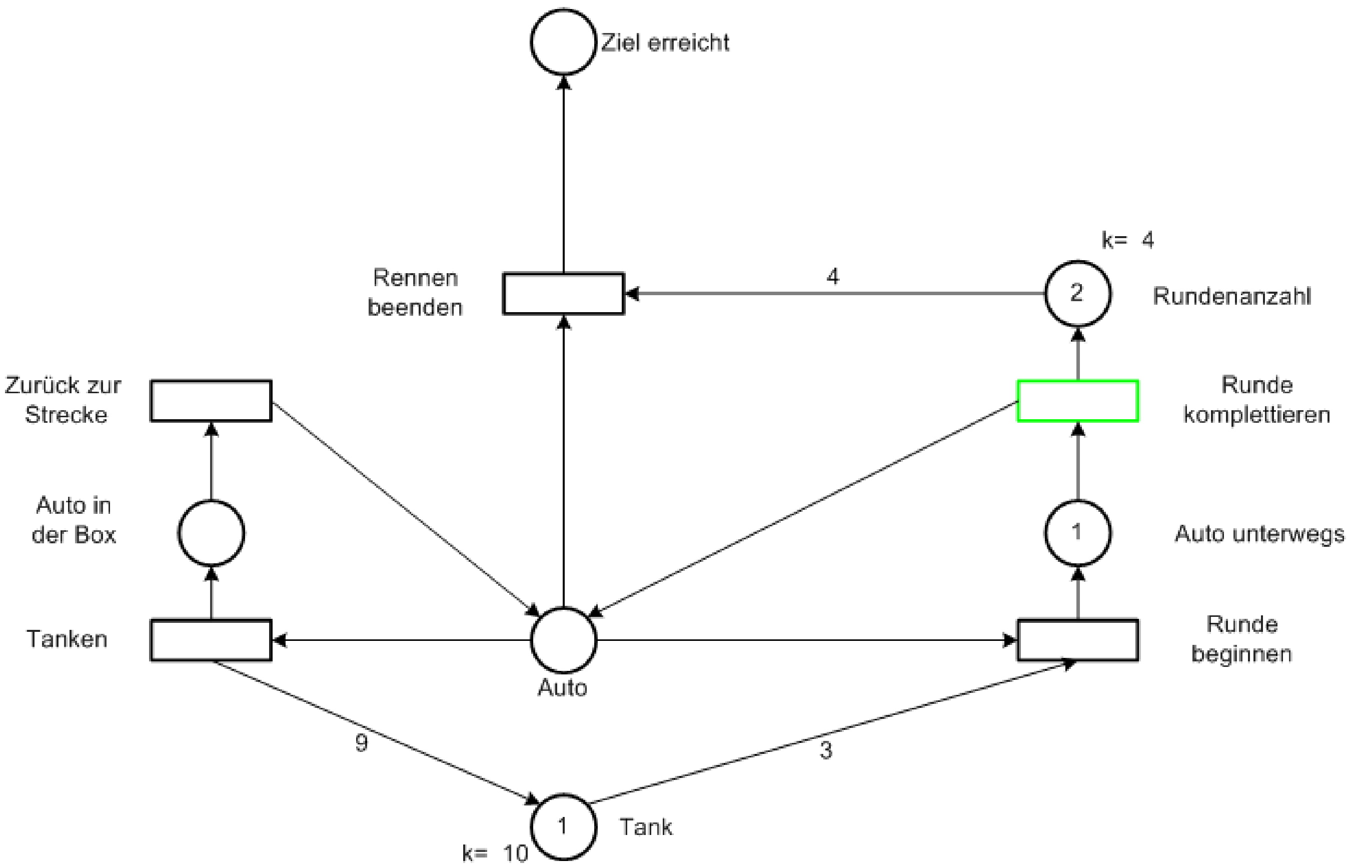
Beispiel – Formel 1



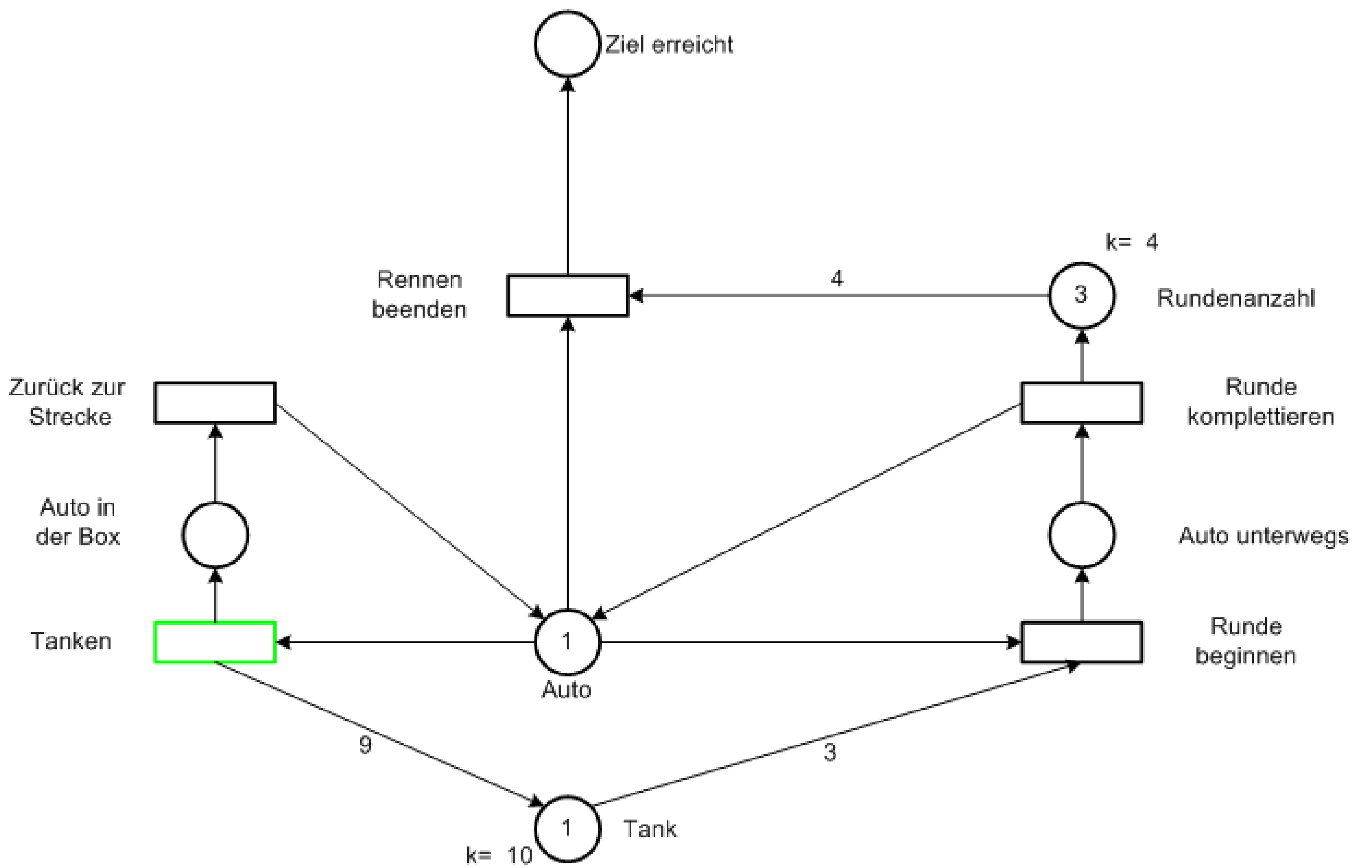
Beispiel – Formel 1



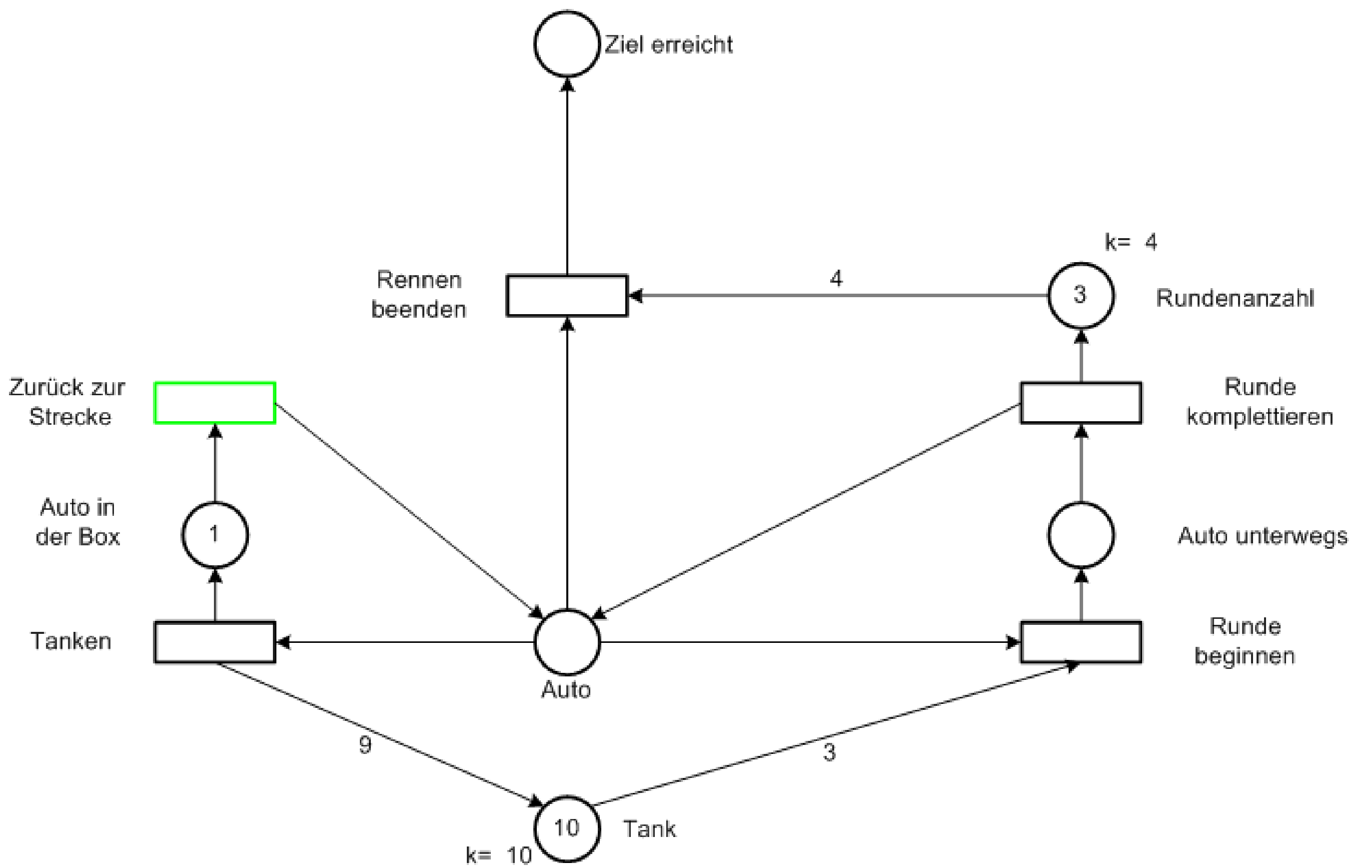
Beispiel – Formel 1



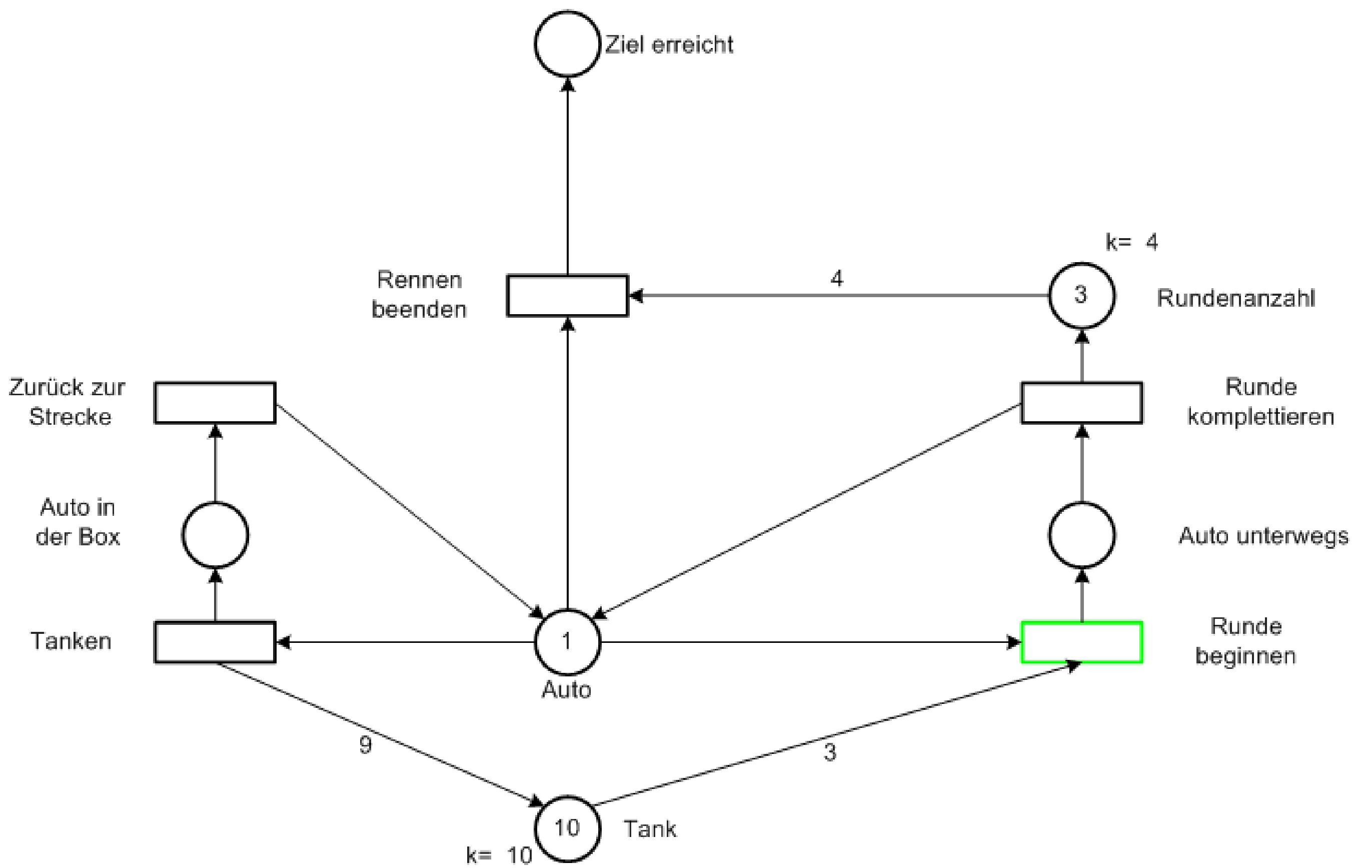
Beispiel – Formel 1



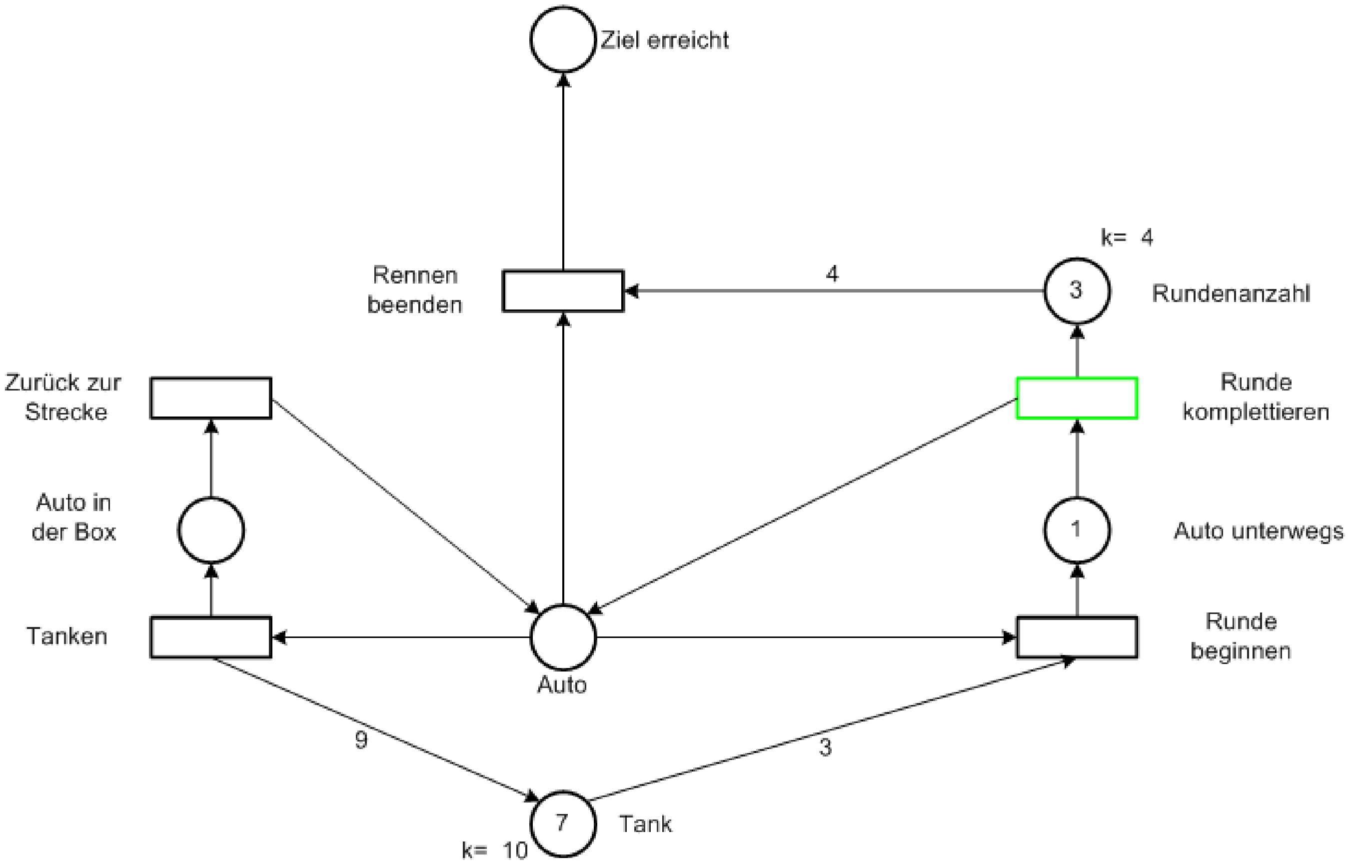
Beispiel – Formel 1



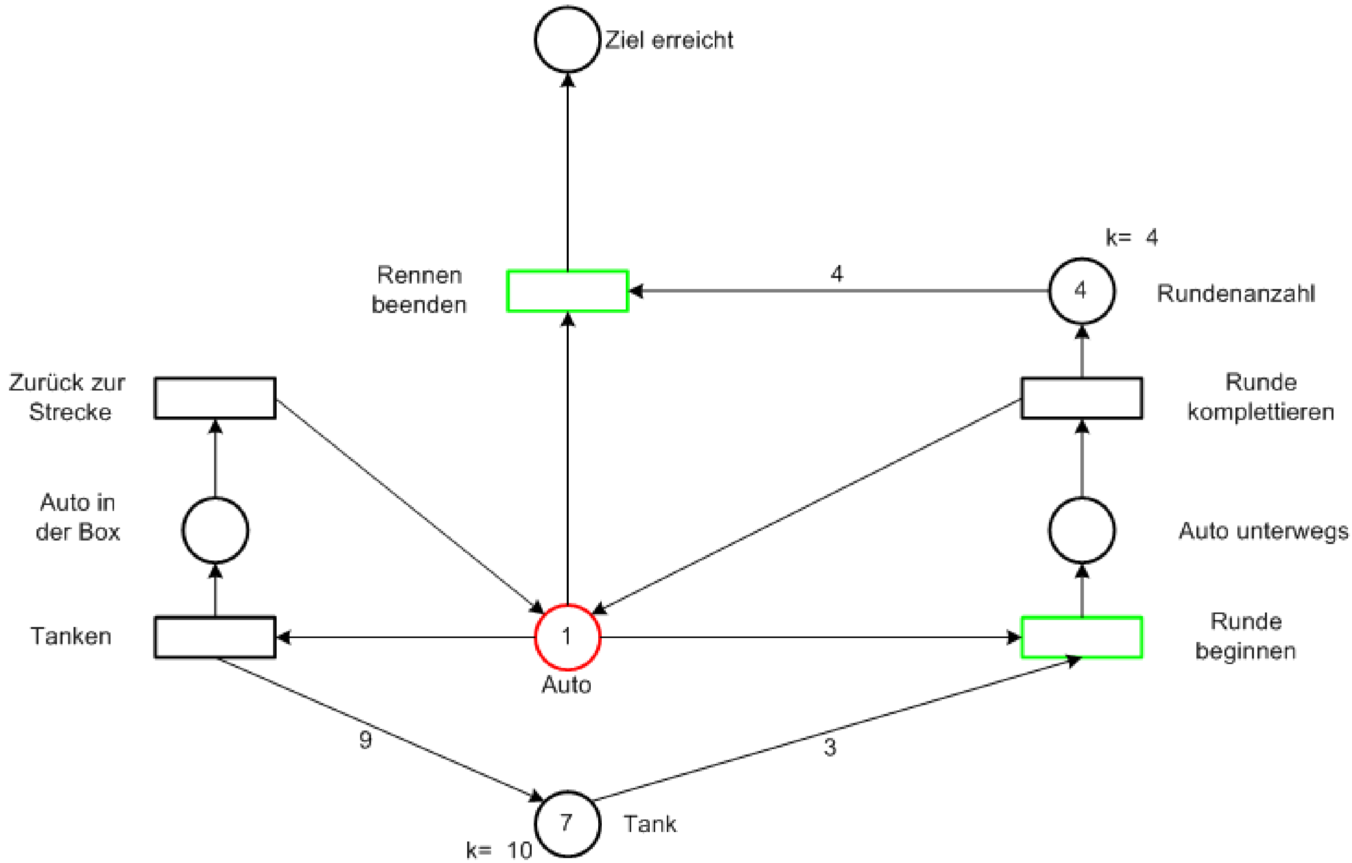
Beispiel – Formel 1



Beispiel – Formel 1



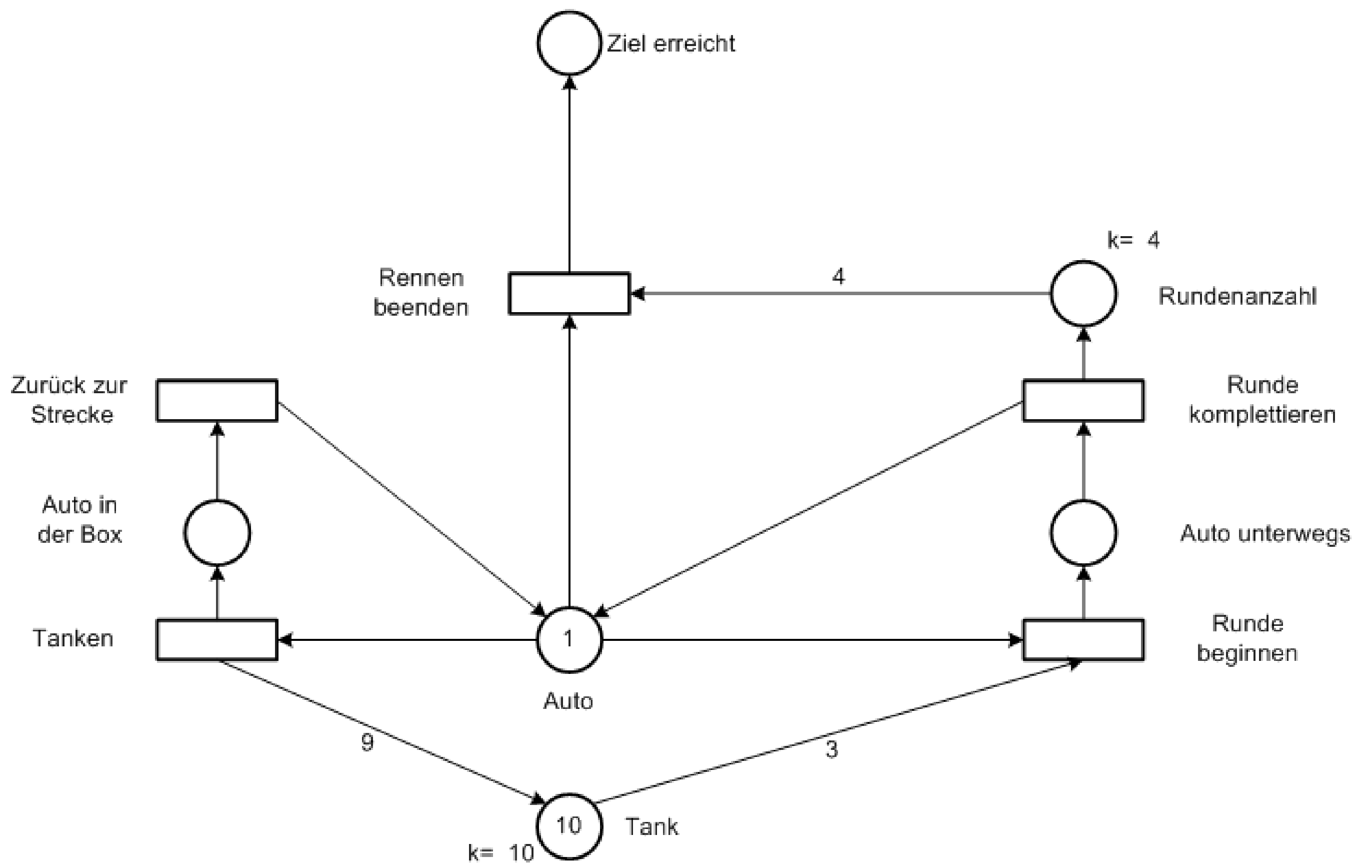
Beispiel – Formel 1



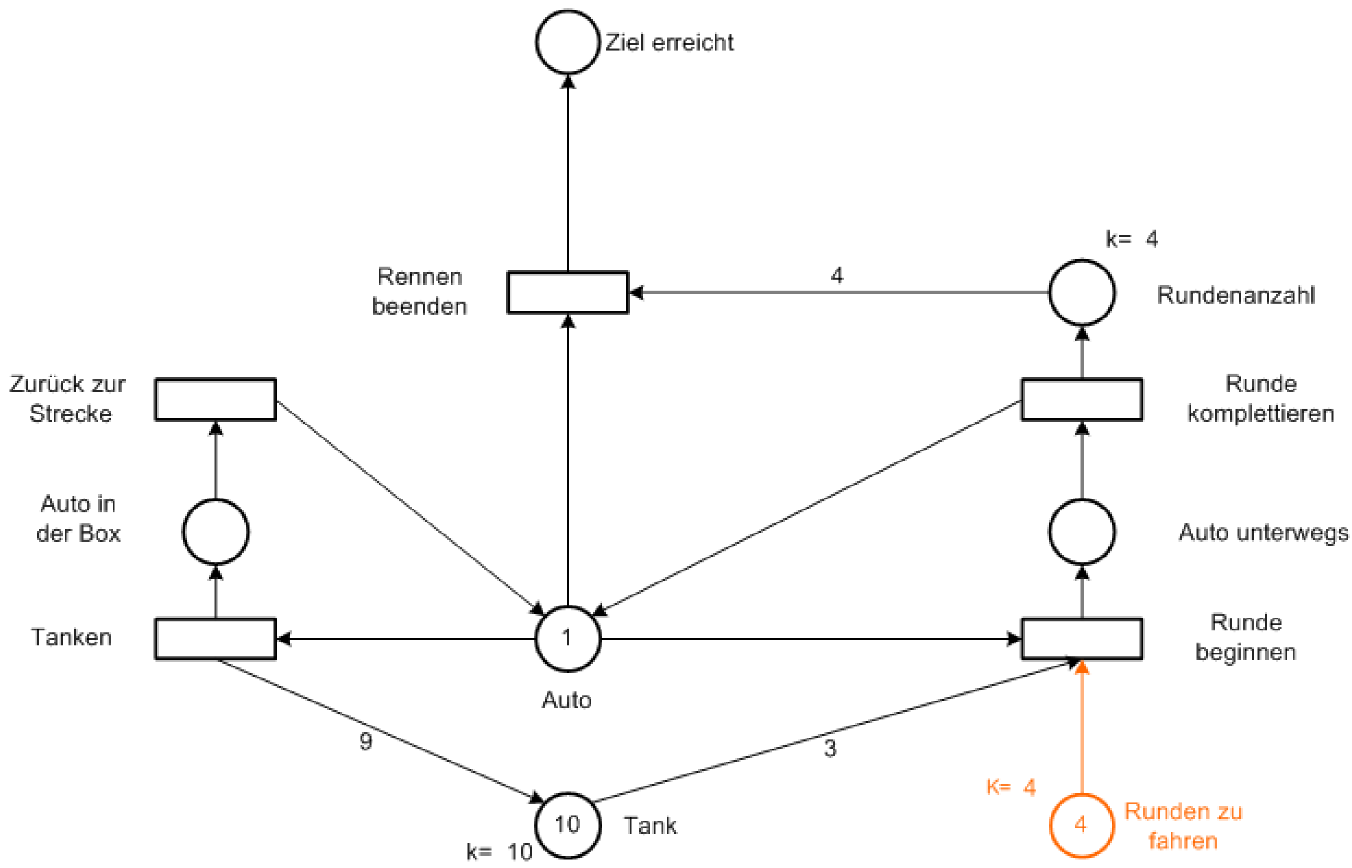
Konflikt

Wie könnte man ihn beheben?

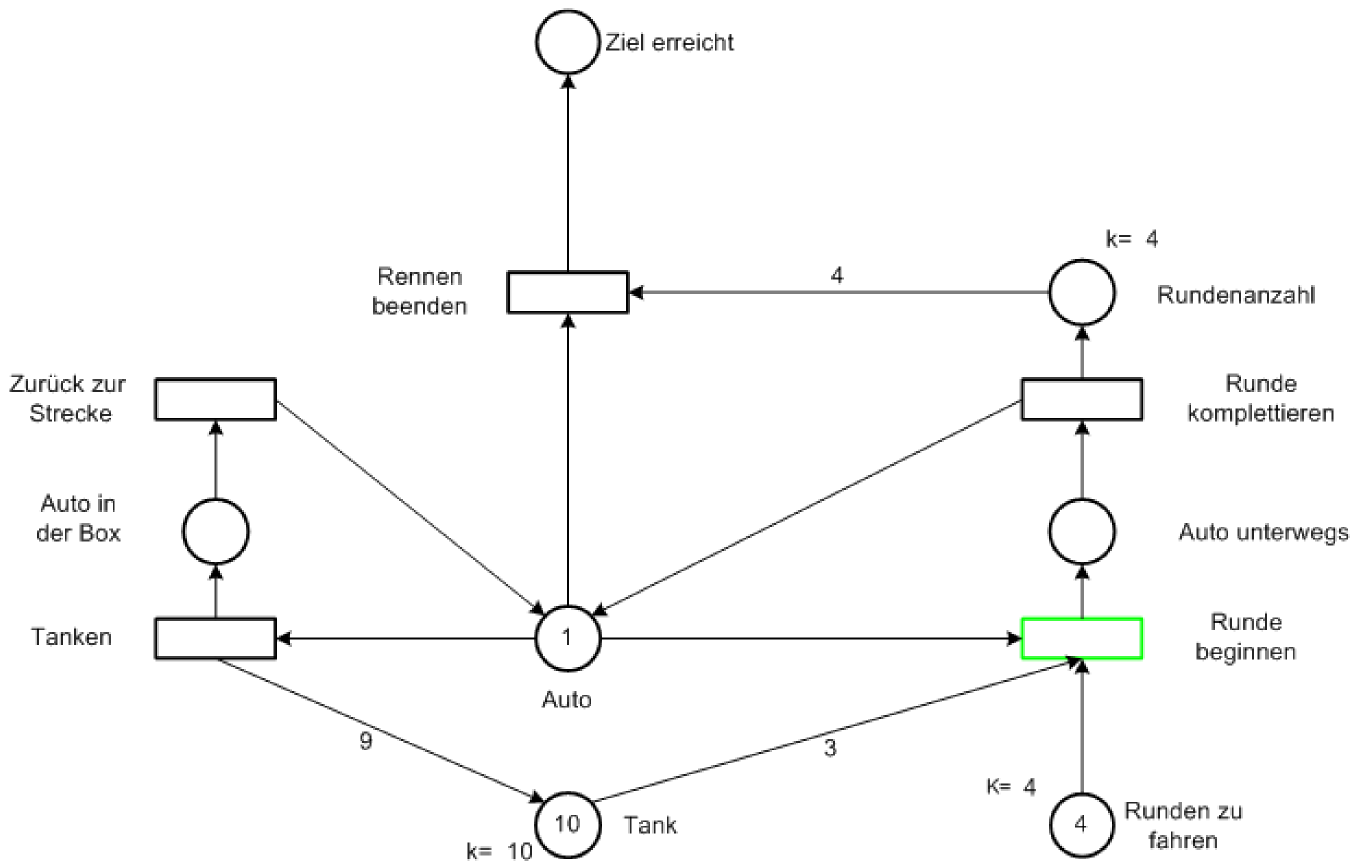
Beispiel – Formel 1



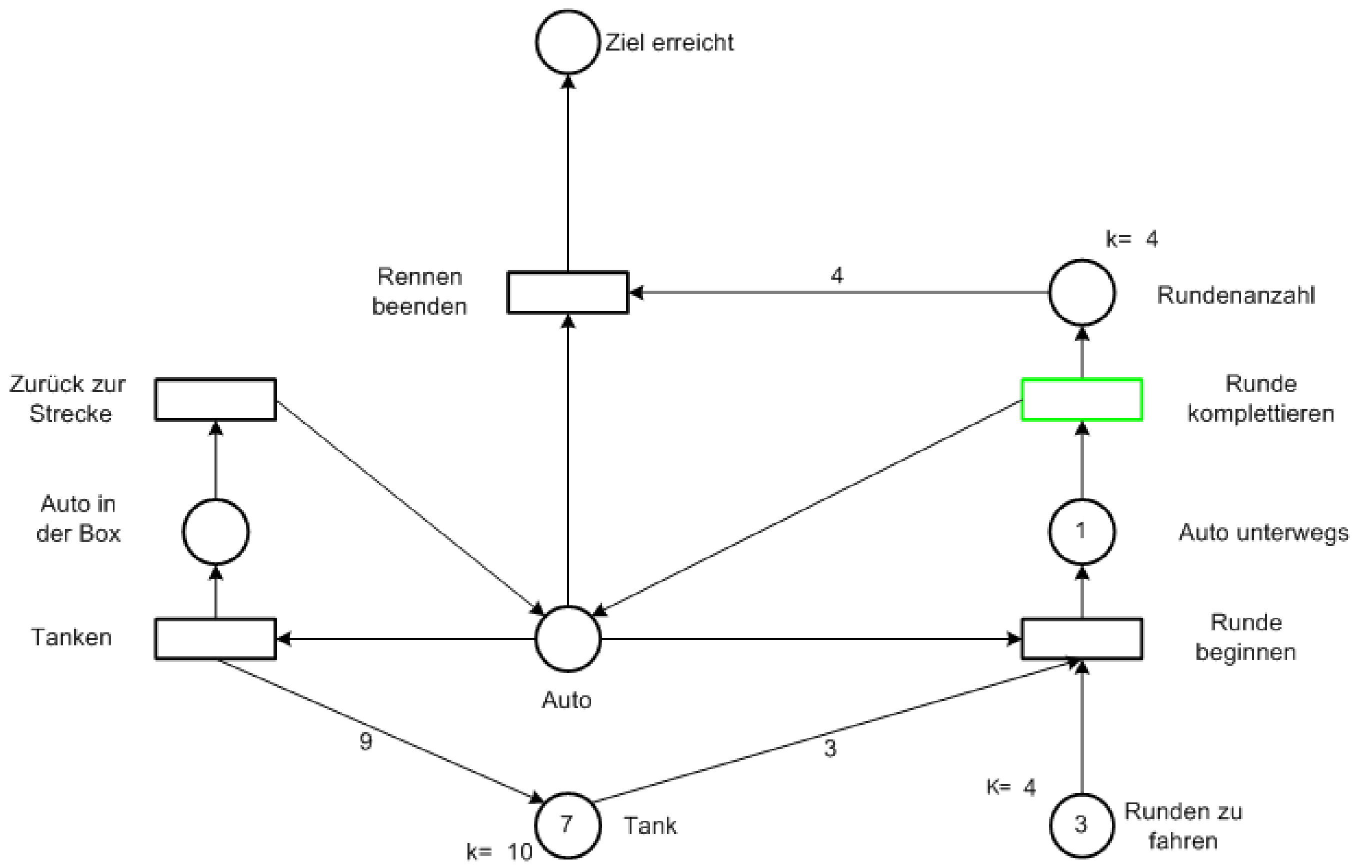
Beispiel – Formel 1



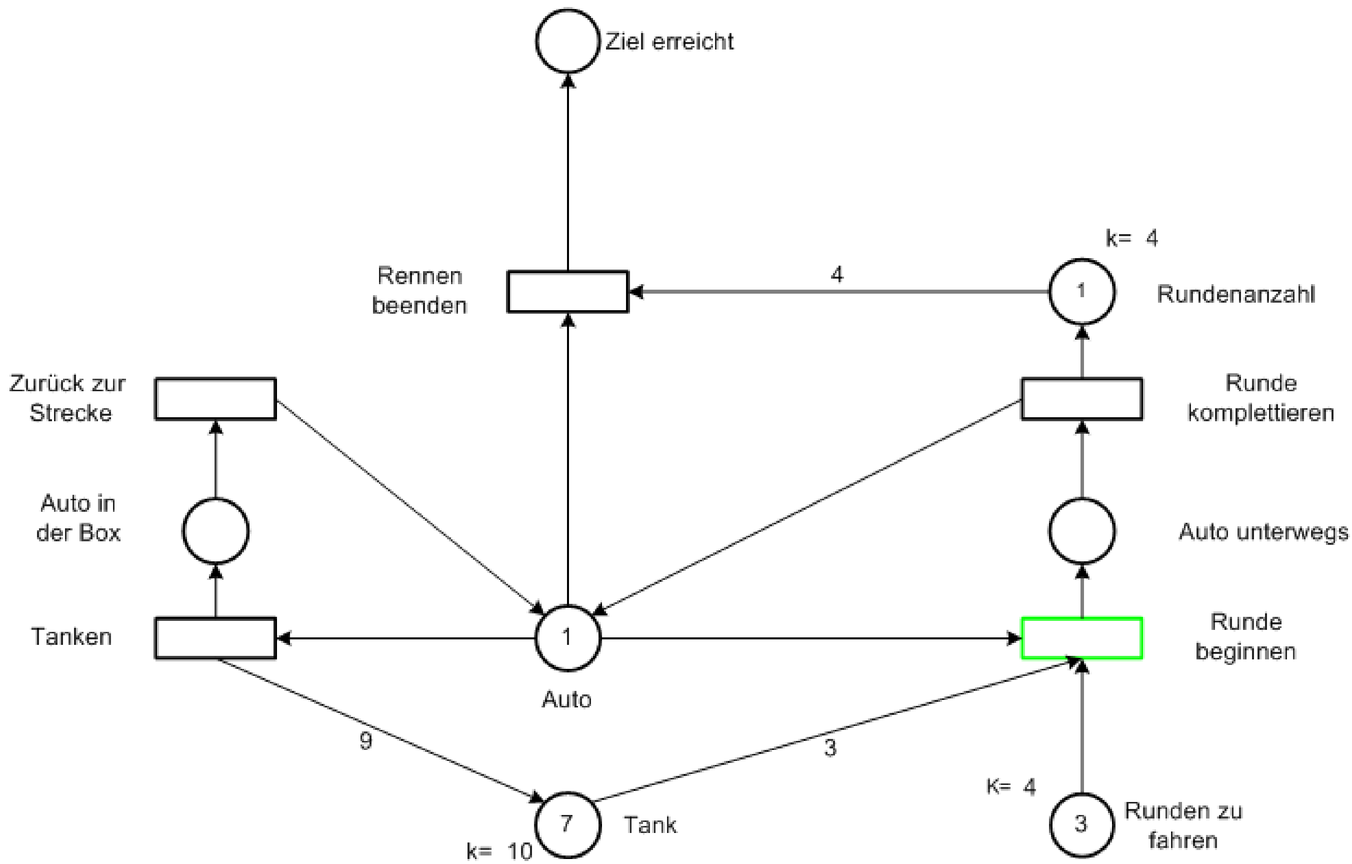
Beispiel – Formel 1



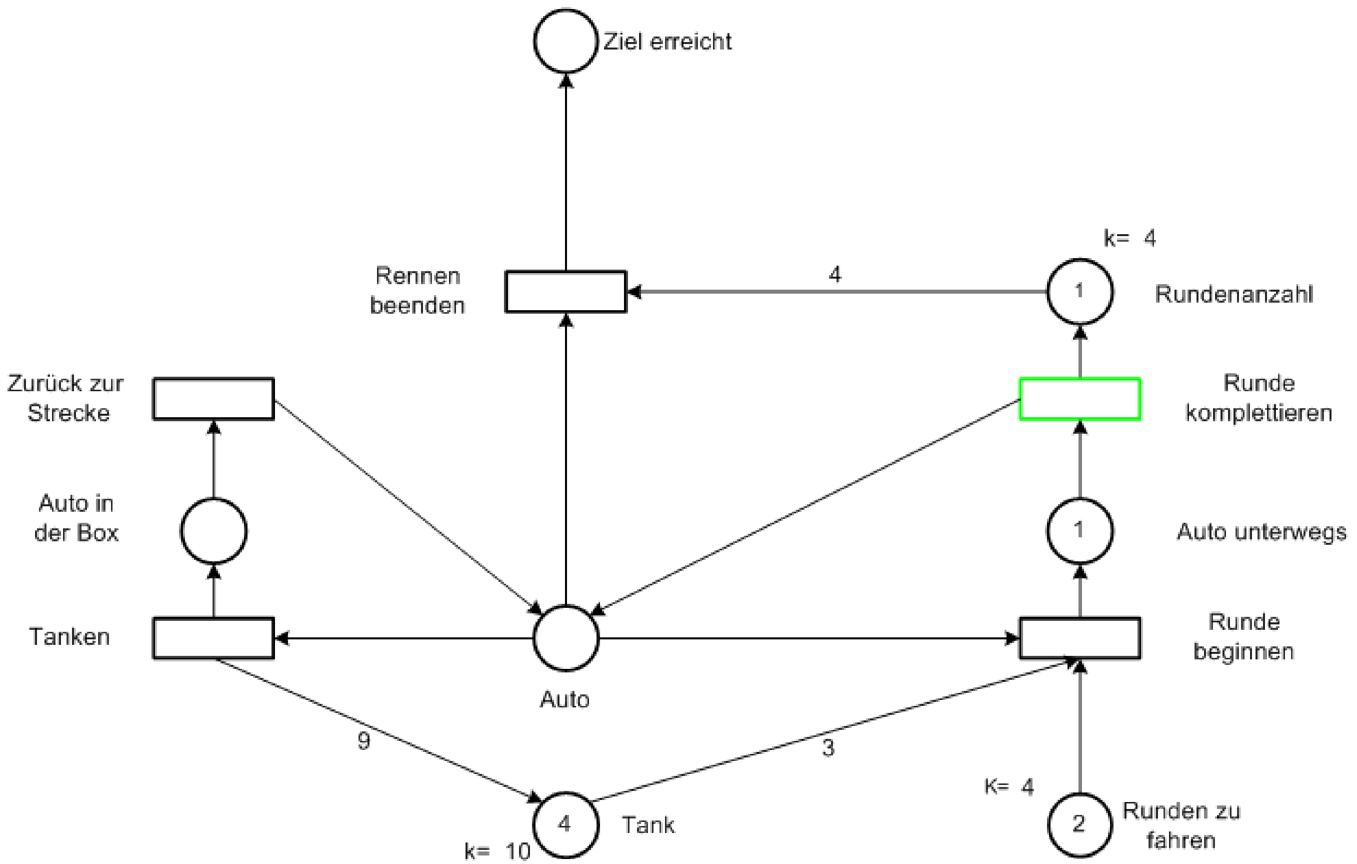
Beispiel – Formel 1



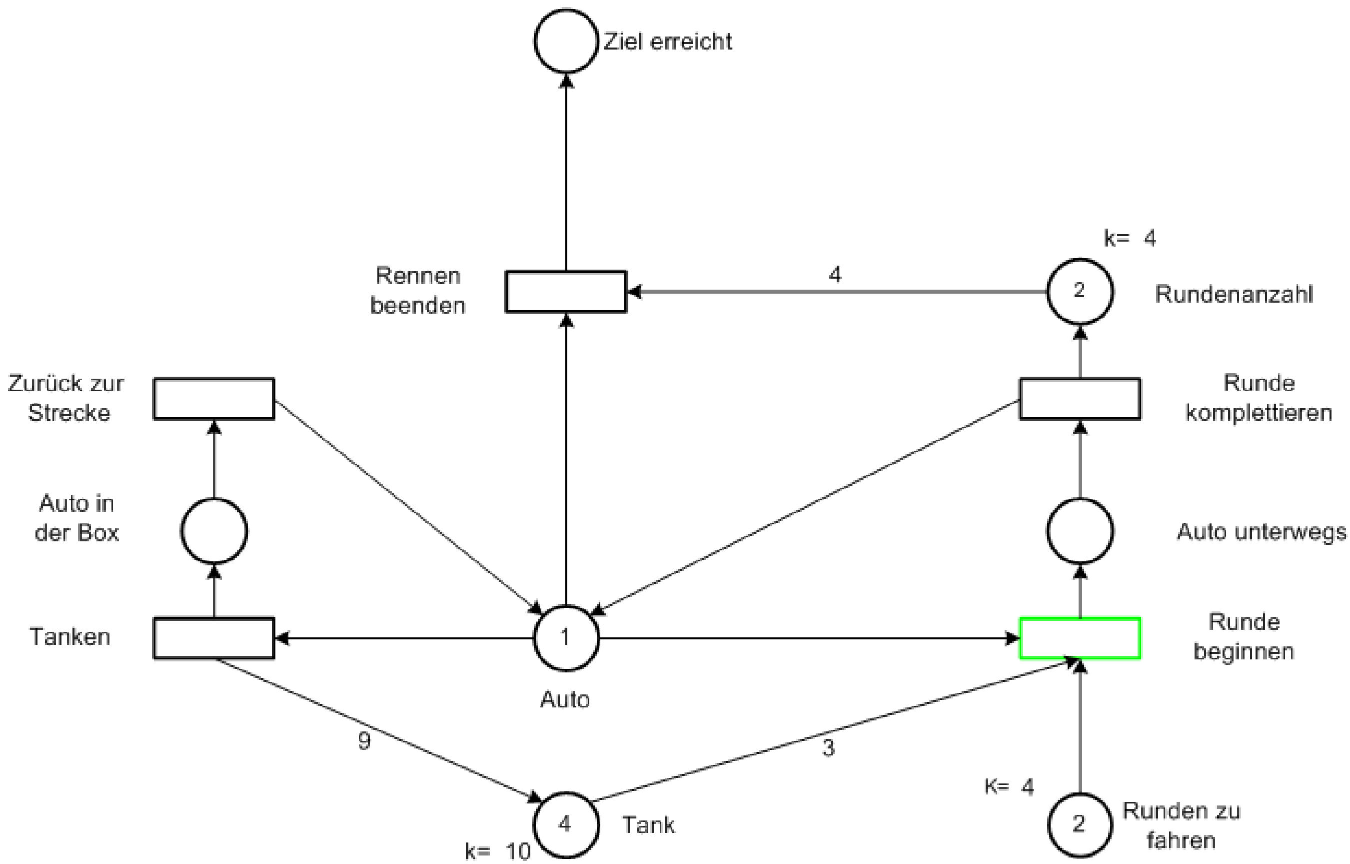
Beispiel – Formel 1



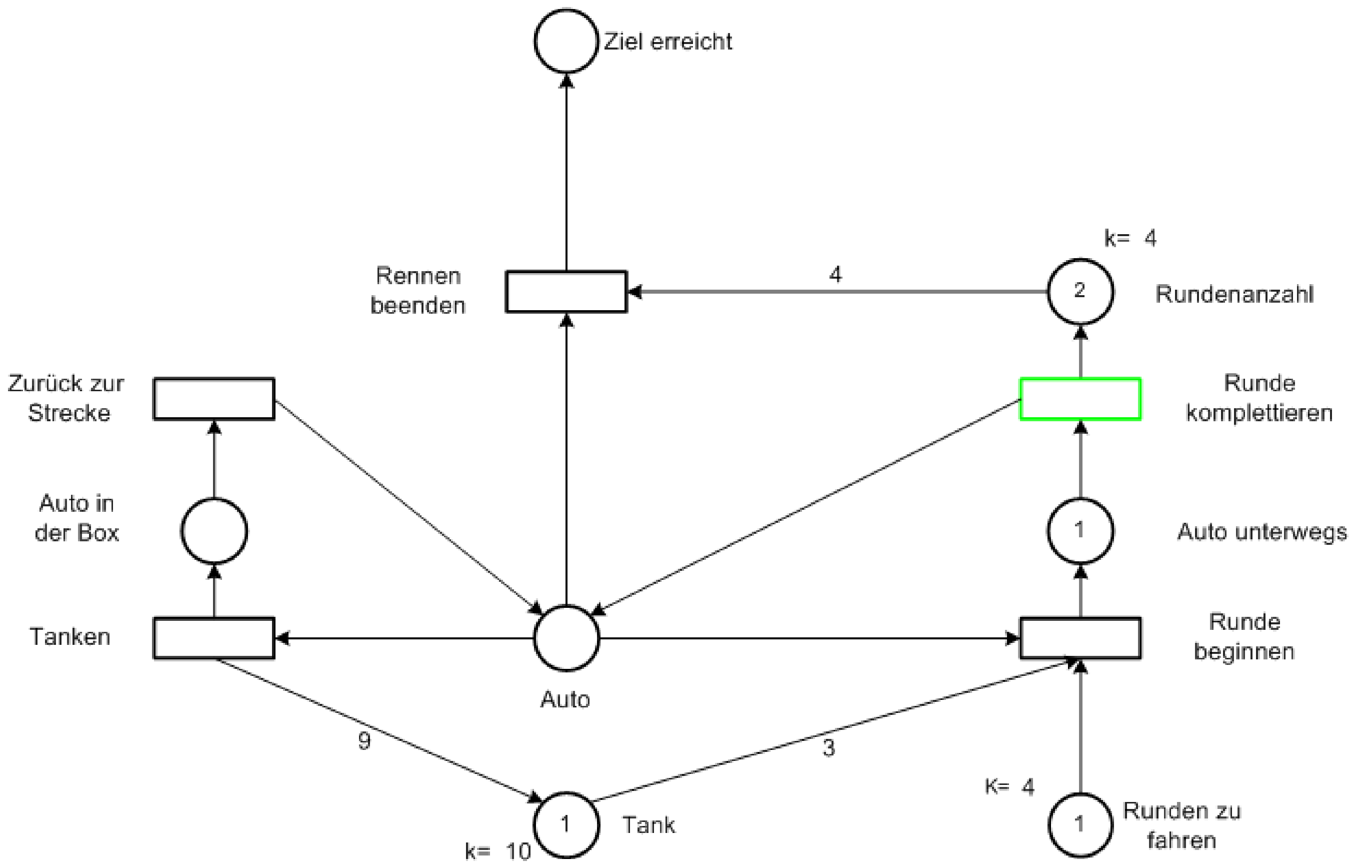
Beispiel – Formel 1



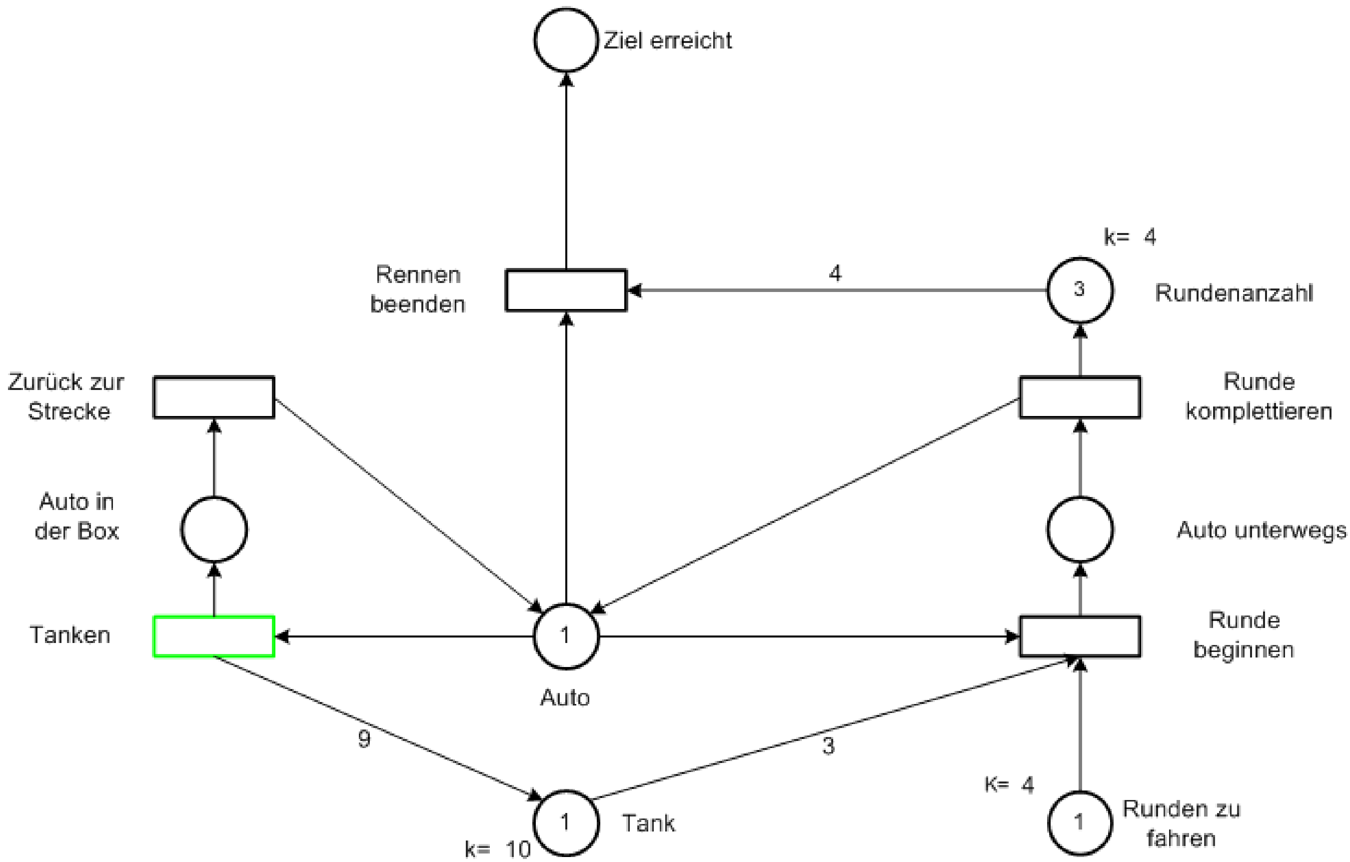
Beispiel – Formel 1



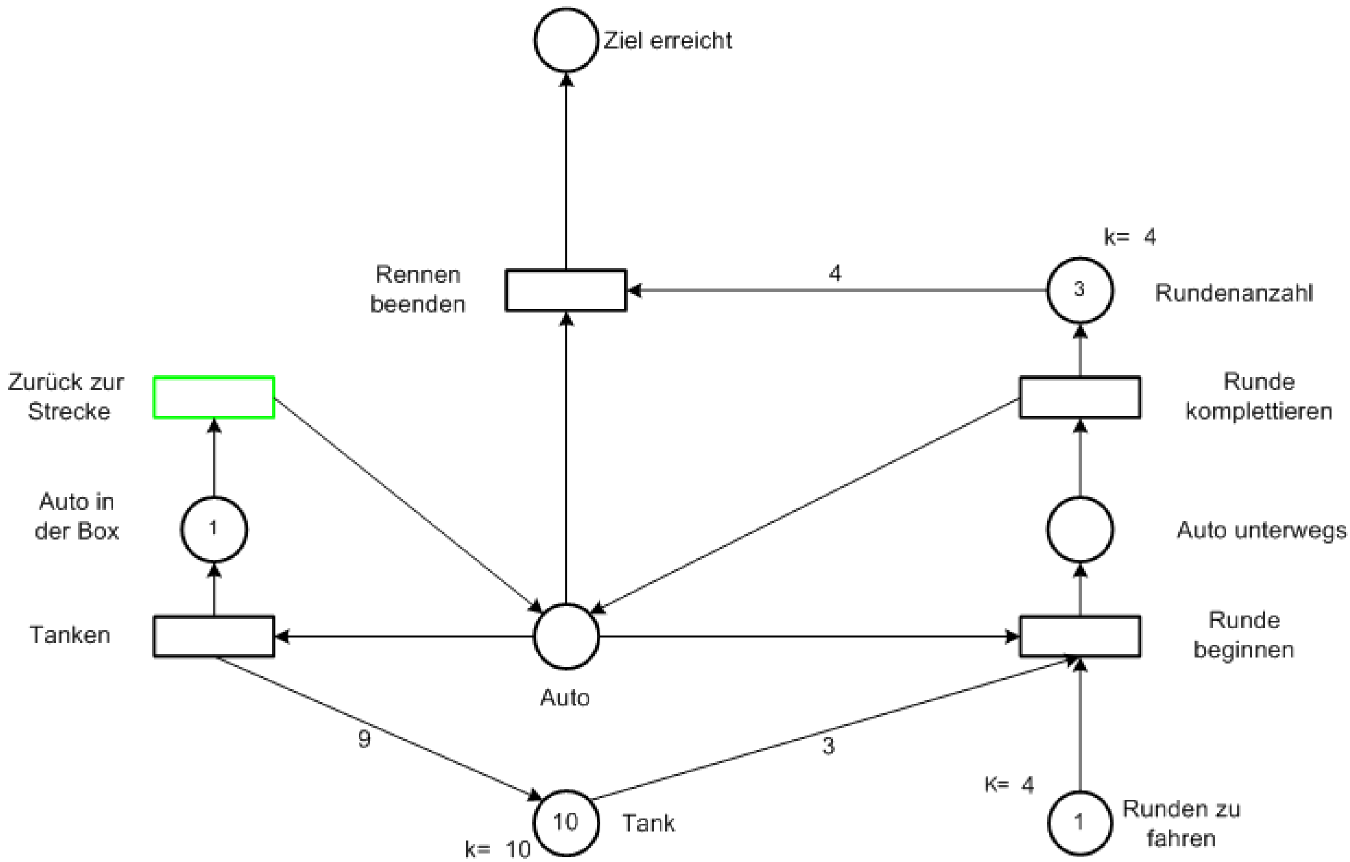
Beispiel – Formel 1



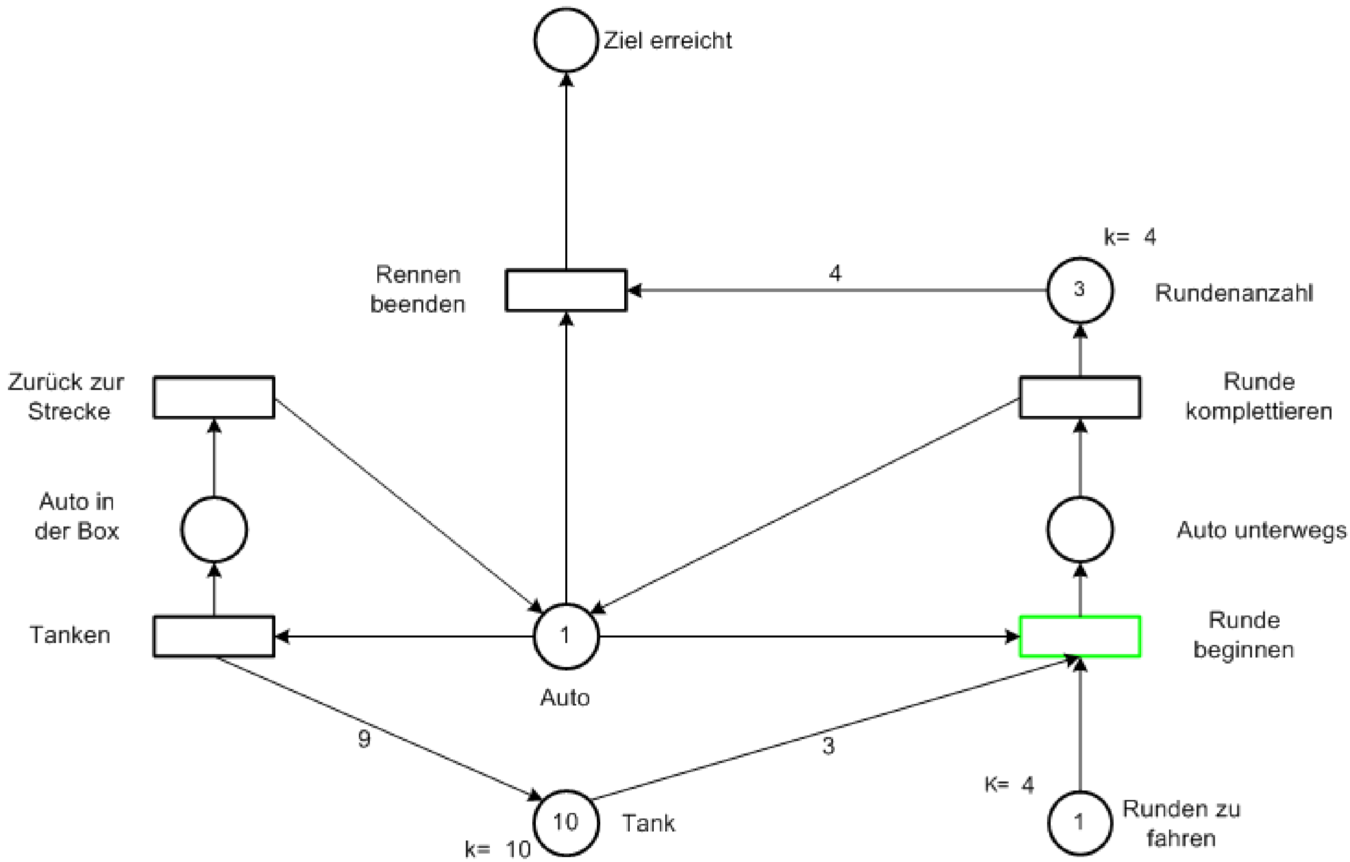
Beispiel – Formel 1



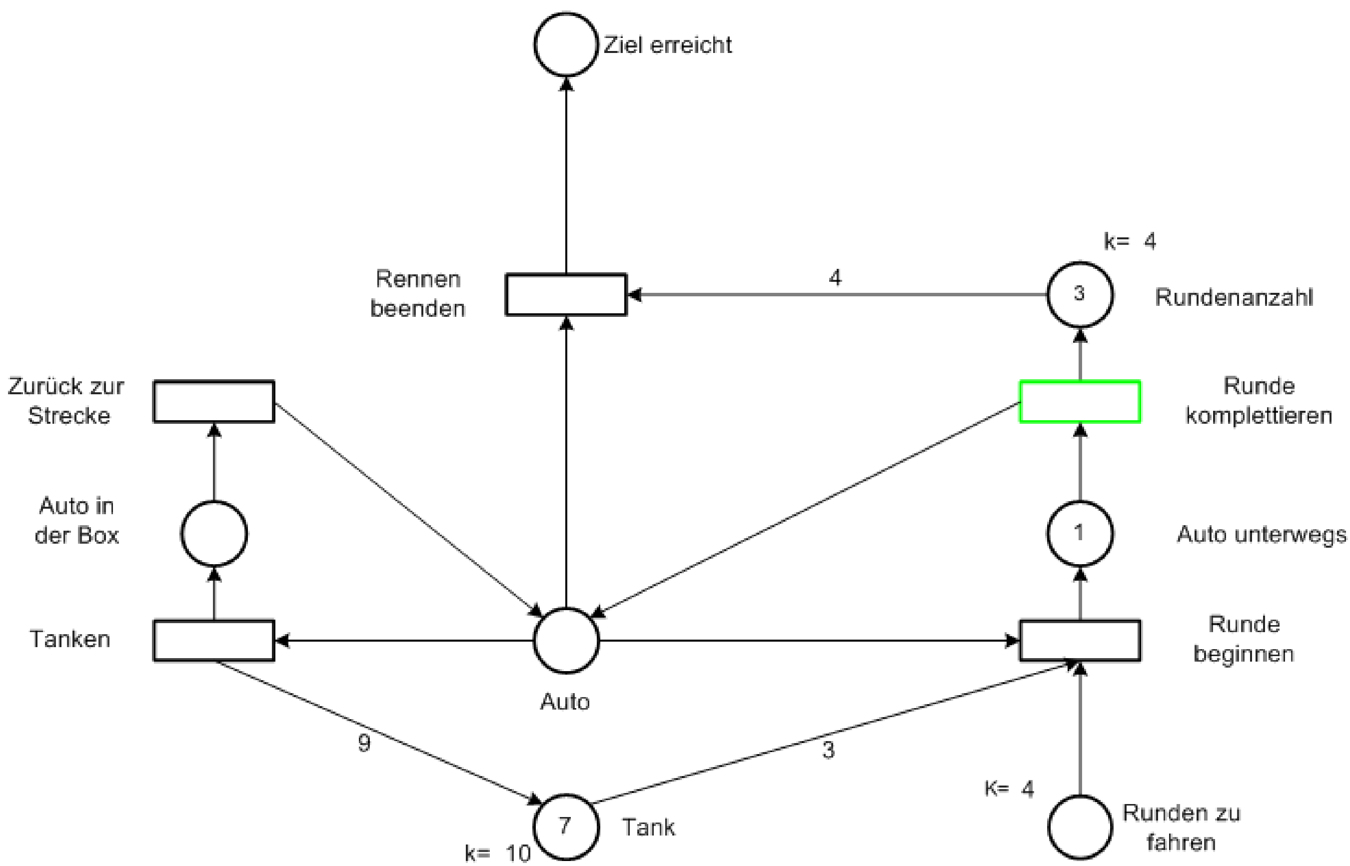
Beispiel – Formel 1



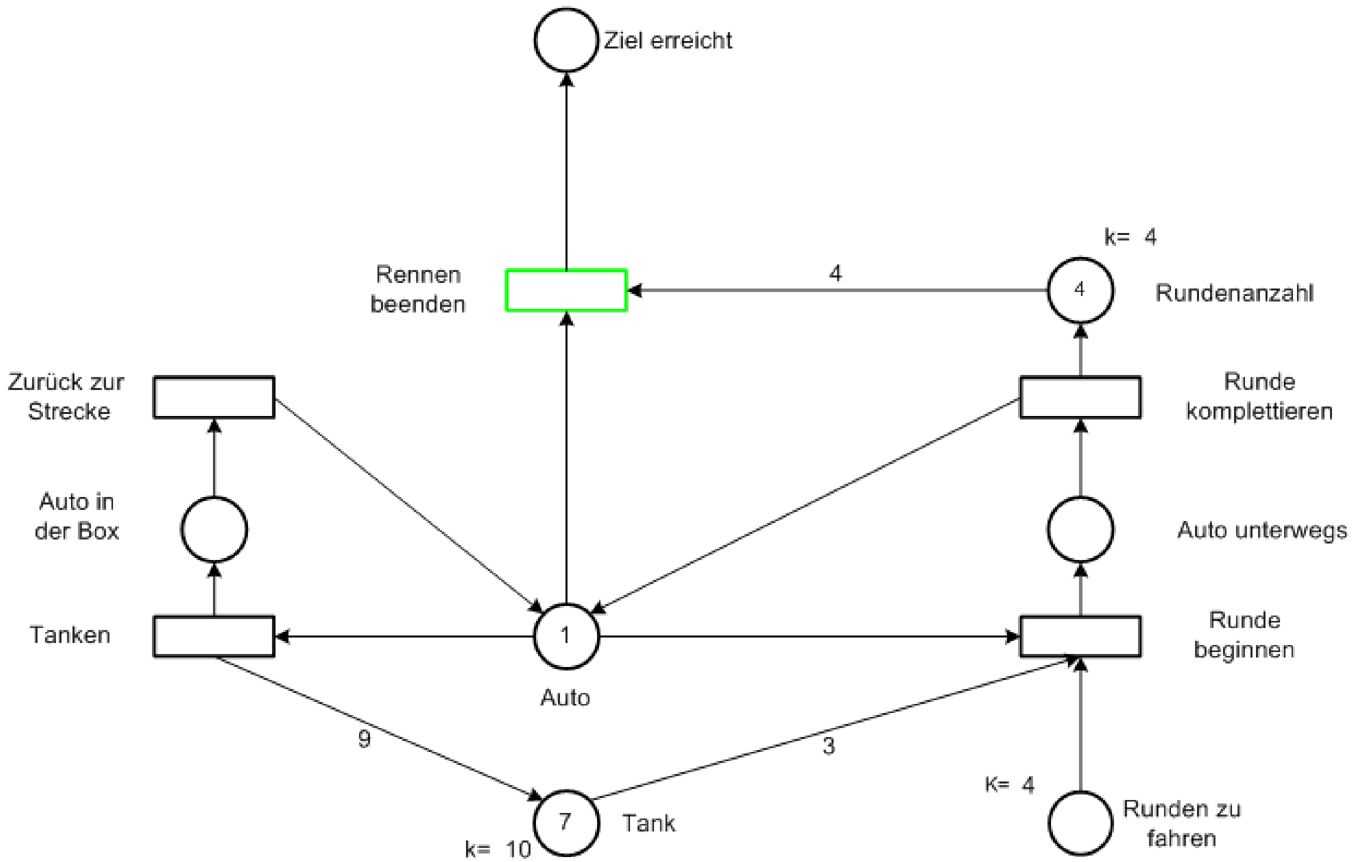
Beispiel – Formel 1



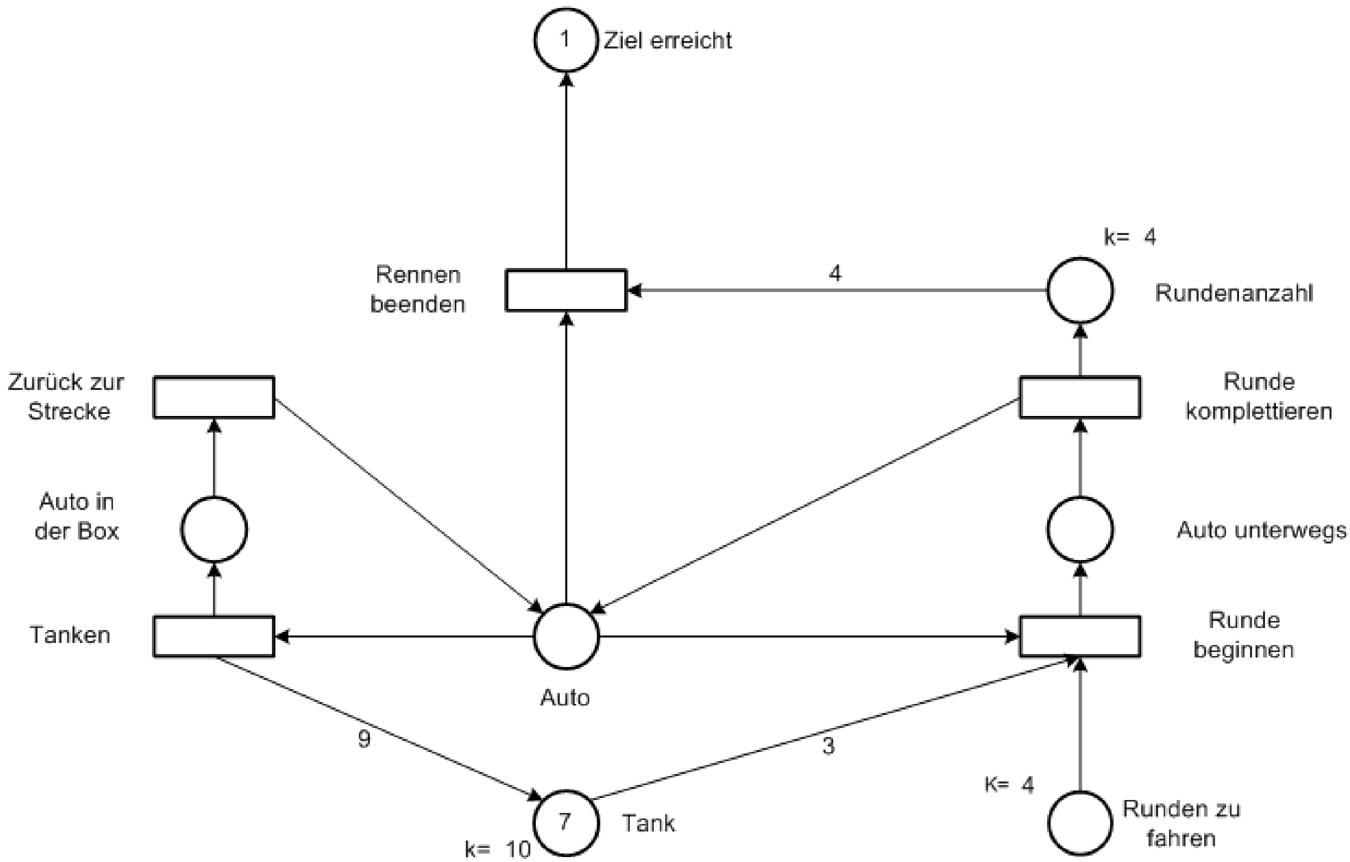
Beispiel – Formel 1



Beispiel – Formel 1

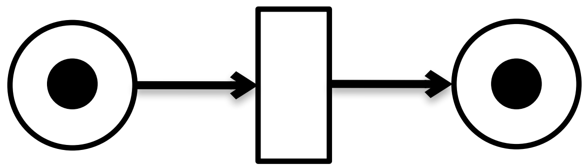


Beispiel – Formel 1

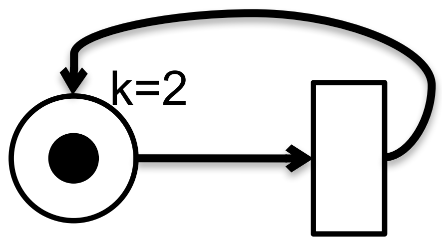


Lebendigkeit in Petri-Netzen

- *Deadlock* – Eine Markierung, unter der keine Transition lebendig ist (tote Markierung)

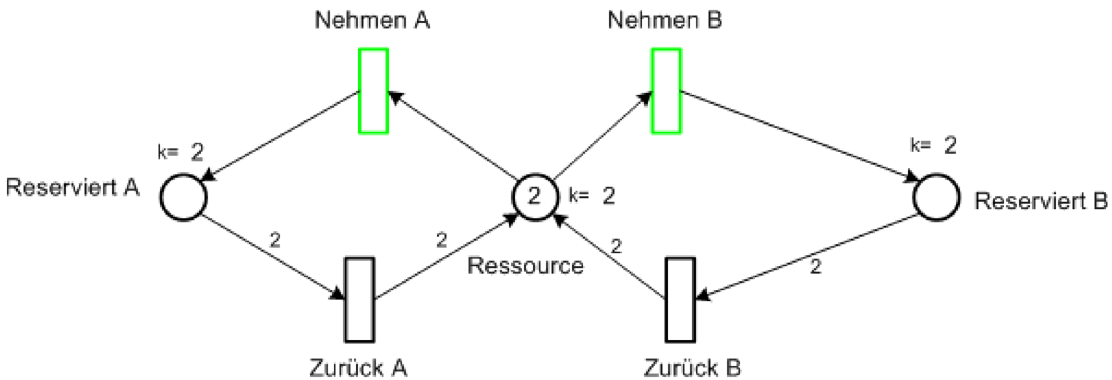


- *Trap* – Eine Stelle, die einmal markiert, unter keiner Folgemarkierung unmarkiert ist



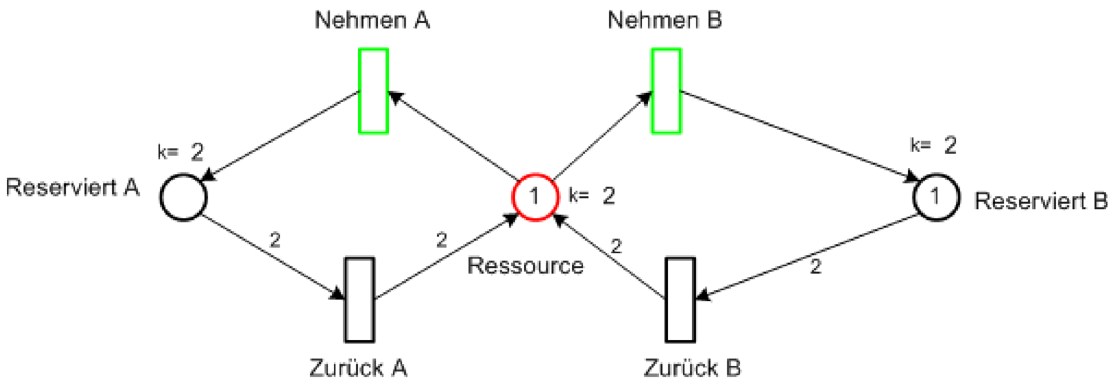
Lebendigkeit in Petri-Netzen (Deadlocks)

- **Deadlock**
  - Formal: Ein Deadlock ist eine Stellenmenge S, die – wenn sie einmal unmarkiert ist – genau dann nie wieder markiert werden kann, wenn keine Transition mehr schalten kann, die in ihrem Nachbereich eine Stelle aus S hat
  - Praktisch: Ein Deadlock liegt vor, wenn ein Teilnetz durch eine (ungünstige) Schaltfolge lahm gelegt werden kann



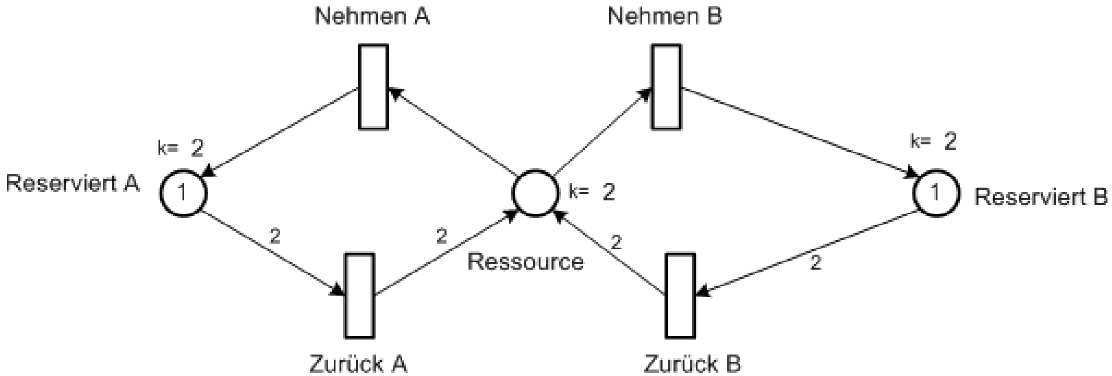
Lebendigkeit in Petri-Netzen (Deadlocks)

- **Deadlock**
  - Formal: Ein Deadlock ist eine Stellenmenge S, die – wenn sie einmal unmarkiert ist – genau dann nie wieder markiert werden kann, wenn keine Transition mehr schalten kann, die in ihrem Nachbereich eine Stelle aus S hat
  - Praktisch: Ein Deadlock liegt vor, wenn ein Teilnetz durch eine (ungünstige) Schaltfolge lahm gelegt werden kann



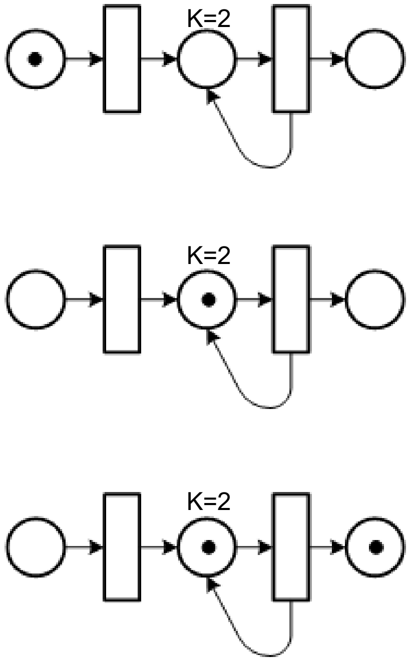
Lebendigkeit in Petri-Netzen (Deadlocks)

- **Deadlock**
  - Formal: Ein Deadlock ist eine Stellenmenge S, die – wenn sie einmal unmarkiert ist – genau dann nie wieder markiert werden kann, wenn keine Transition mehr schalten kann, die in ihrem Nachbereich eine Stelle aus S hat
  - Praktisch: Ein Deadlock liegt vor, wenn ein Teilnetz durch eine (ungünstige) Schaltfolge lahm gelegt werden kann

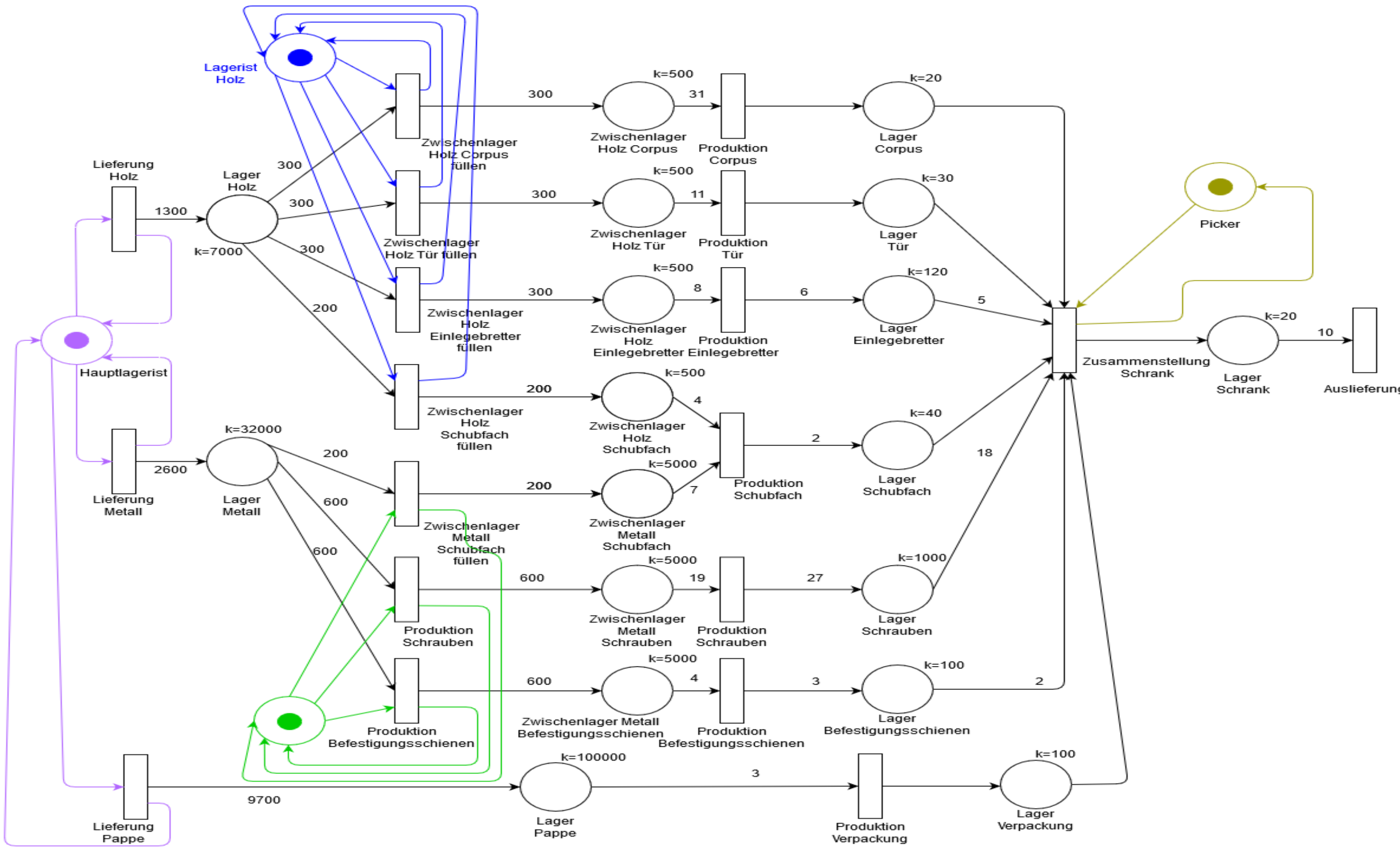


Lebendigkeit in Petri-Netzen (Traps)

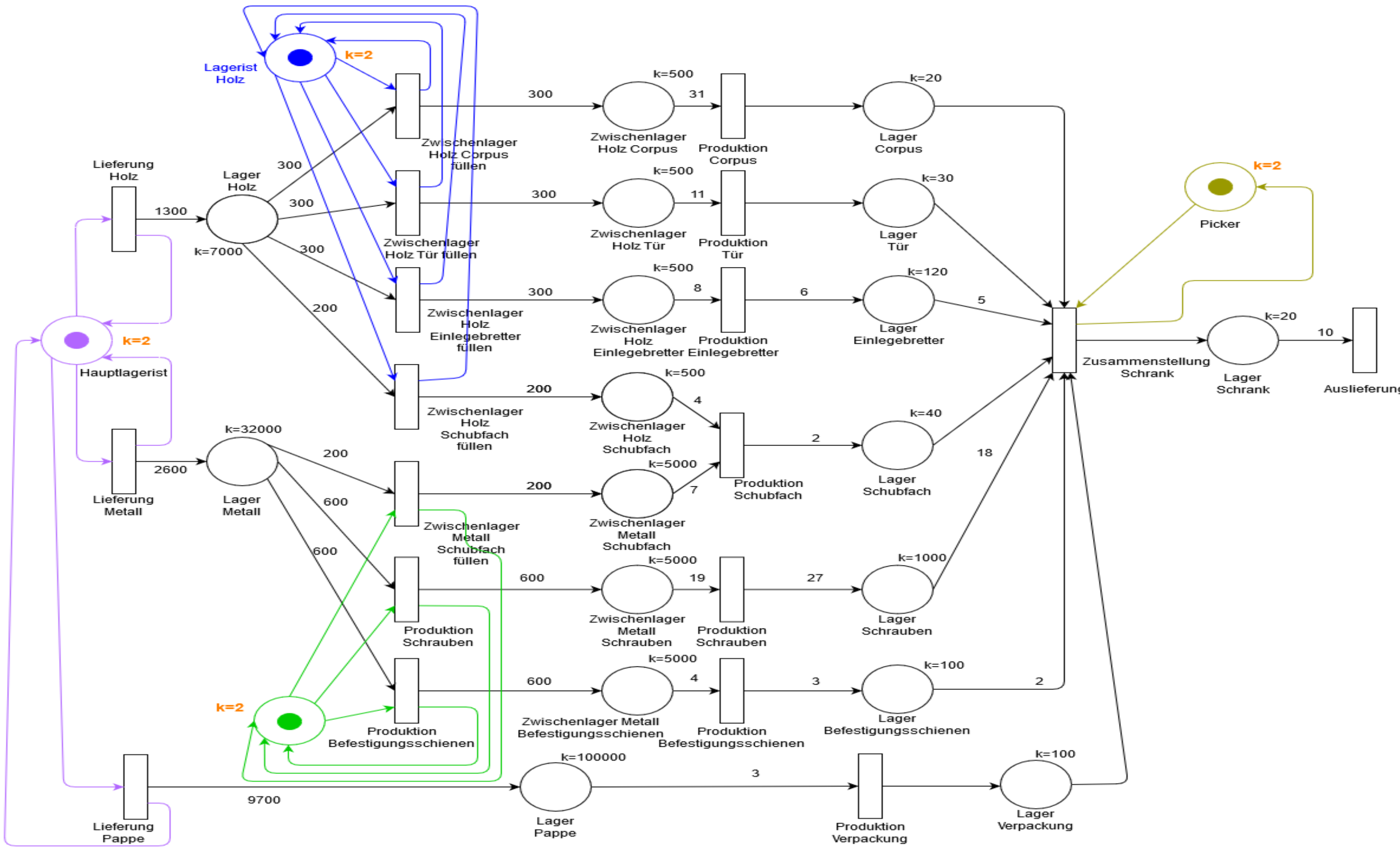
- **Trap**
  - Formal: Ein Trap ist eine Stellenmenge S, die – wenn sie einmal markiert wurde – nie mehr alle Marken verlieren kann
  - Praktisch: Die Stelle wird nicht leer



Beispiel – Schrank



Beispiel – Schrank



Lebendigkeit in Petri-Netzen

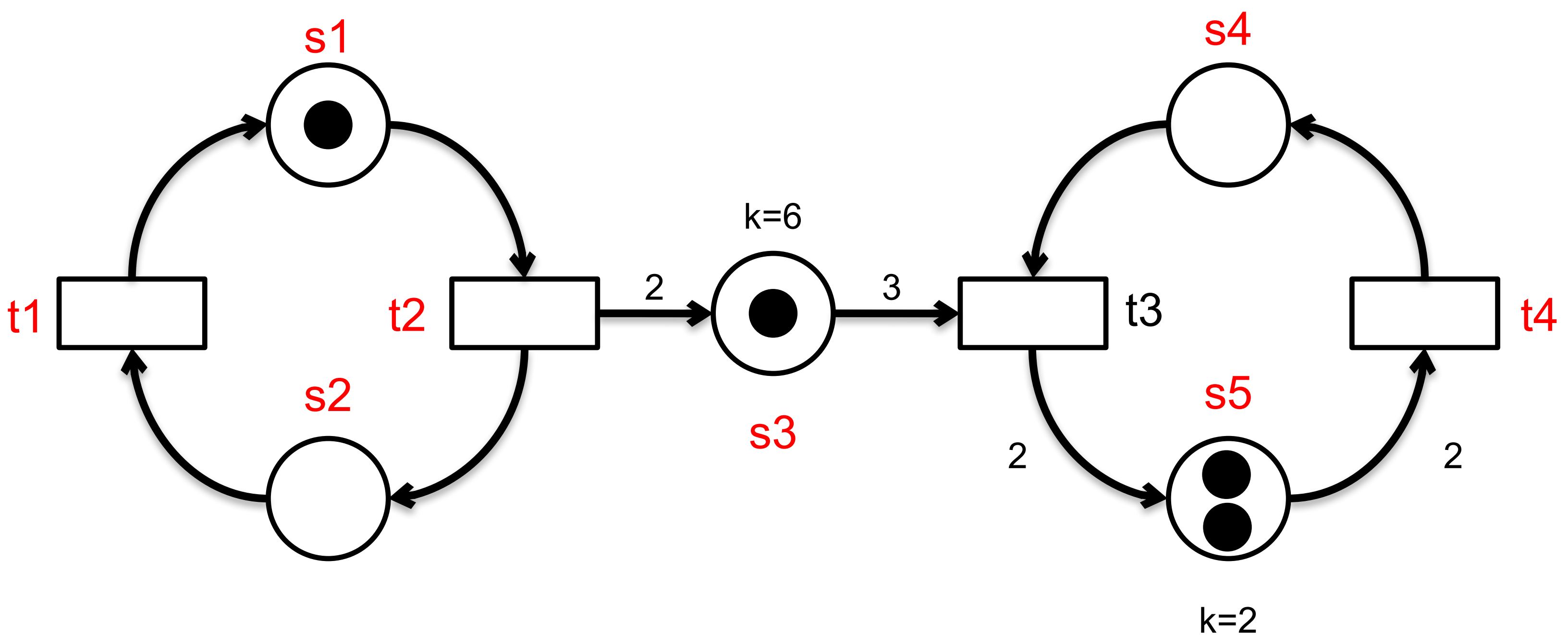
- **Lebendigkeit**
  - Eine Transition heißt
    - tot, wenn sie bei keinem der möglichen Folgezustände (Folgemarkierung) aktiviert ist, d. h. nie wieder schalten kann
    - aktivierbar, wenn sie unter mindestens einer Folgemarkierung aktiviert ist
    - lebendig, wenn sie unter allen Folgemarkierungen aktivierbar ist
  - Ein Petri-Netz heißt
    - tot, wenn alle Transitionen tot sind
    - deadlockfrei oder schwach lebendig falls es unter keiner Folgemarkierung tot ist, d. h. bei jeder Folgemarkierung mindestens eine Transition aktivierbar ist.
    - (stark) lebendig, wenn alle Transitionen lebendig sind

Weitere Eigenschaften

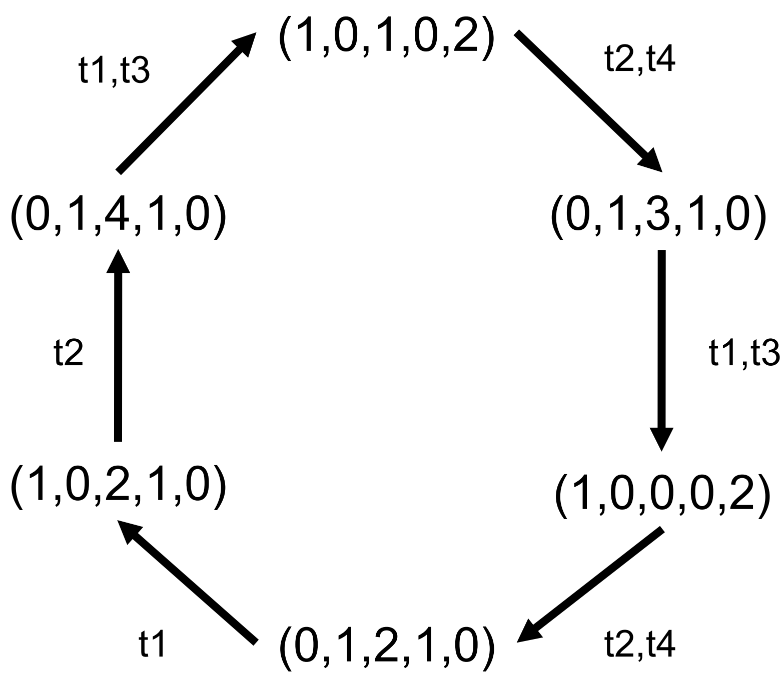
- *Terminierung* – Ein Petri–Netz terminiert unter einer bestimmten Anfangsmarkierung, wenn jede Schaltsequenz zu einem Deadlock führt
- *Reversibilität* – Ein Petri–Netz, das aus jedem seiner Zustände in seinen Anfangszustand zurückkehren kann, wird rücksetzbar genannt
- –Beschränktheit – Jede Stelle enthält unter jeder Folgemarkierung maximal Marken
- In einem Petri–Netz können diese Eigenschaften nachgewiesen werden, jedoch sind die nötigen Algorithmen oft zu komplex für die Praxis

Erreichbarkeitsgraph

- Der Erreichbarkeitsgraph enthält alle von der Anfangsmarkierung erreichbaren Markierungen
- Beispiel Erzeuger-Verbraucher-System



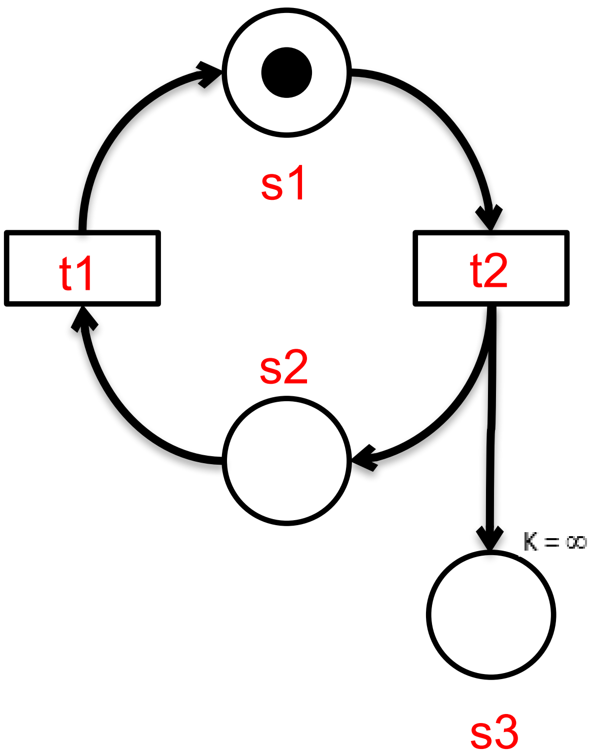
Erreichbarkeitsgraph cont.

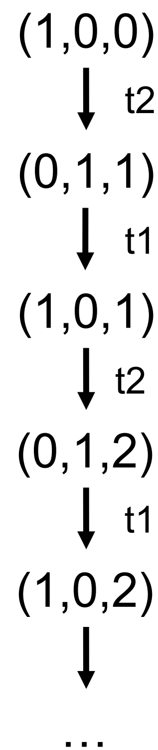


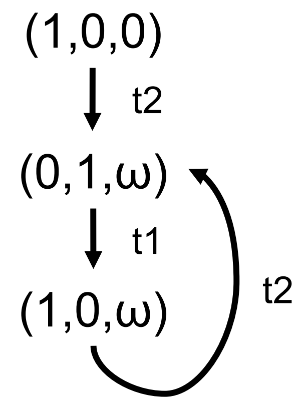
Lebendig? Reversibel? Deadlockfrei? Trapfrei? Beschränkt?

Überdeckungsgraph

- Erreichbarkeitsgraph kann unendlich groß sein
- Überdeckungsgraph kann Platzhalter für unendliche Werte enthalten (ω)







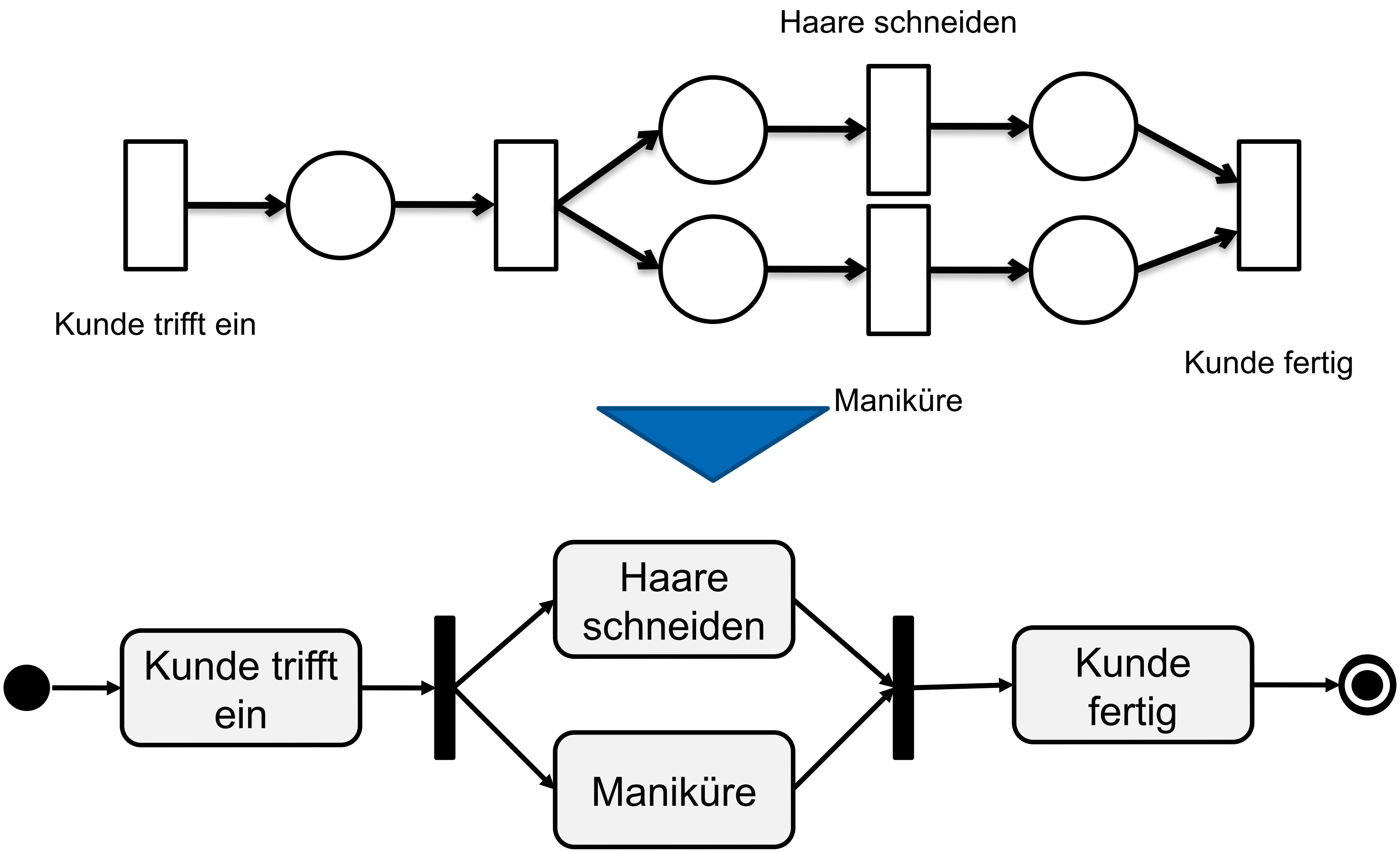
Strukturelle Analyse von Petri-Netzen

- Verhalten eines Petri-Netzes (z.B. Beschränktheit, Lebendigkeit) kann u.U. durch Analyse der Struktur bestimmt werden
- Entspricht die Struktur eines Petri-Netzes bestimmten Anforderungen, können Verhaltenseigenschaften abgeleitet werden
- z.B. (extended) Free Choice Nets, Marked Graphs, State Machines

Unterarten von Petri-Netzen: Marked Graph

- „Markierter Graph“
- Ein Petri-Netz heißt Markierter Graph, wenn jede Stelle genau eine ausgehende und eine eingehende Kante hat
  - Konflikte sind nicht möglich
  - Nebenläufigkeit kann abgebildet werden
  - Bei einer Marke lebendig und reversibel
- Kann ohne Informationsverlust als Aktivitätsdiagramm abgebildet werden (Stellen als Pfeile)
- Zur Abbildung nicht-deterministischer Prozesse aufgrund von Nebenläufigkeit

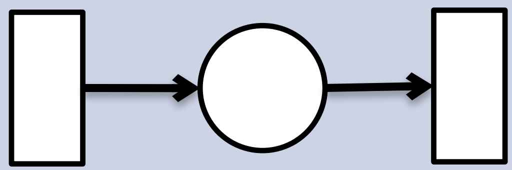
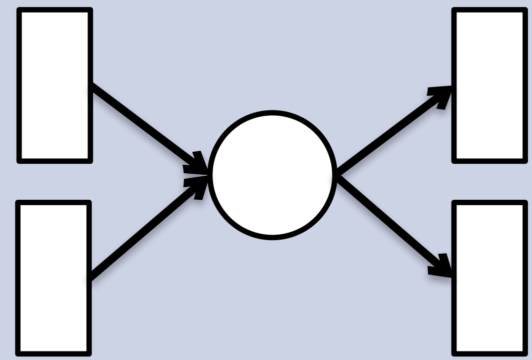
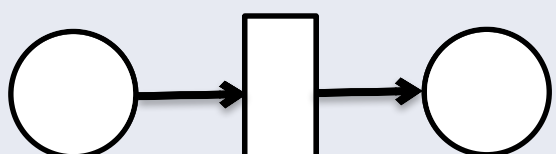
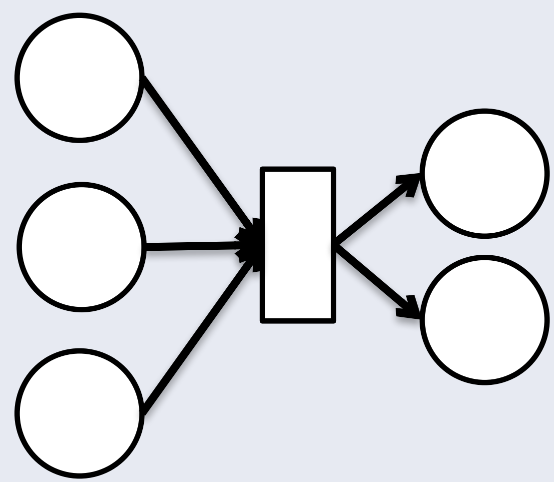
Beispiel Markierter Graph



Unterarten von Petri-Netzen: State Machine

- (Finiter) „Zustandsautomat"
- Ein Petri-Netz heißt State Machine, wenn jede Transition genau eine eingehende und eine ausgehende Kante besitzt
  - Keine Nebenläufigkeit
  - Konflikte sind möglich
  - Kein Erzeugen/Verbrauchen von Marken => 1-beschränkt
  - Bei einer Marke lebendig und reversibel
- Kann ohne Informationsverlust als Zustandsdiagramm abgebildet werden (Transitionen als Pfeile)
- Zur Abbildung nicht-deterministischer Prozesse aufgrund von geteilten Ressourcen

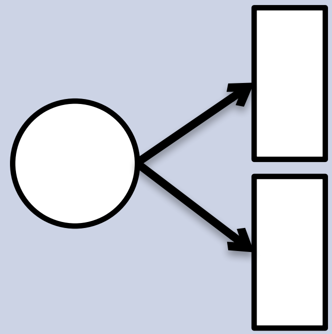
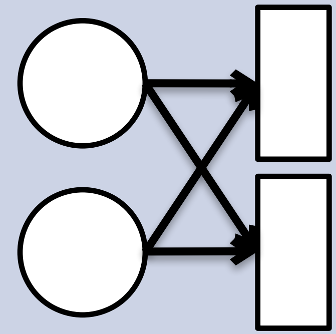
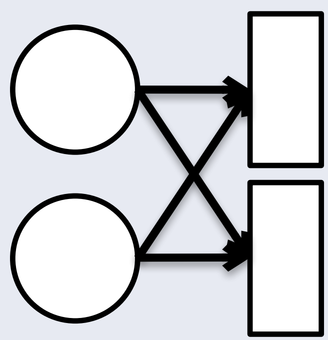
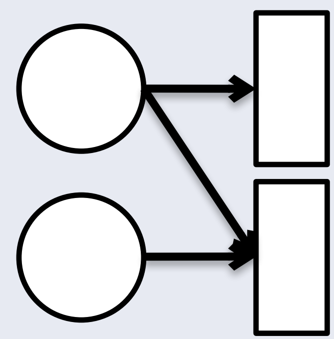
Marked Graph/State Machine Visualisierung

	Erlaubt	Nicht Erlaubt
Marked Graph		
State Machine		

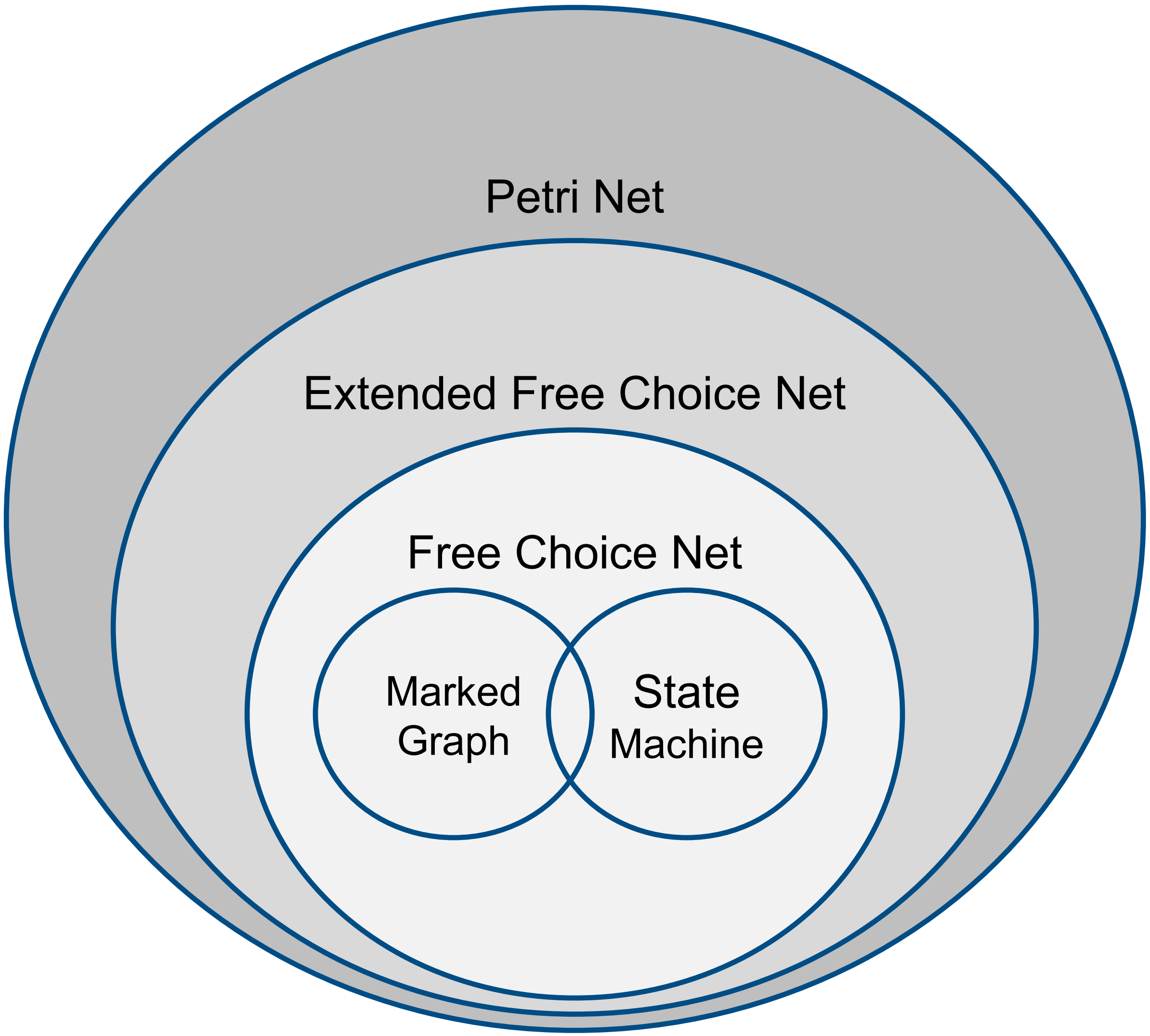
Unterarten von Petri-Netzen: Free Choice Net

- „Freie-Wahl-Netz“
- Jede Kante von einer Stelle zu einer Transition ist entweder
  - Die einzige Kante, die in dieser Stelle beginnt, oder
  - Die einzige Kante, die an dieser Transition endet
  - ➢ Nebenläufigkeit und Konflikte sind möglich, aber nicht gleichzeitig
- Die Wahl zwischen dem Feuern zweier Transitionen ist niemals abhängig vom Rest des Netzes
  - Wenn eine Transition im Nachbereich einer Stelle aktiviert ist, sind alle Transitionen im Nachbereich dieser Stelle aktiviert
- Extended Free Choice Net: kann in ein FCN überführt werden
  - Existiert eine Kante von einer Stelle s zu einer Transition, dann existiert eine Kante von allen Stellen vor dieser Transition zu jeder Transition im Nachbereich von s
- Ist ein (e)FCN frei von Deadlocks und Traps, ist es lebendig

Free-Choice Netze Visualisierung

	Erlaubt	Nicht Erlaubt
Free Choice Net		
Extended Free Choice Net		

Unterarten von Petri-Netzen: Zusammenfassung



Hörsaalübung Münzautomat

- Das Verhalten eines Münzautomaten soll modelliert werden
  - Der Automat steht bereit, wenn keine Münzen im Automaten sind
  - Eine Münze kann in den bereitstehenden Automaten eingeworfen werden
  - Ist eine Münze im Automaten, kann entweder zum Zustand der Bereitschaft abgebrochen werden oder ein Artikel ausgewählt werden
  - Ist ein Artikel ausgewählt, wird dieser ausgegeben und der Automat steht erneut bereit

- Erweiterungen
  - Es befinden sich fünf Artikel im Automaten. Ist der Automat leer, werden alle Artikel wieder aufgefüllt.
  - Ein Artikel kostet 50 ct, es werden 10 ct und 20 ct Münzen angenommen. Werden zuviele Münzen eingeworfen, verbleiben diese im Automaten.

Zusammenfassung

- Petri-Netze geeignet zur Modellierung logischer und nebenläufiger Prozesse
- Bestehen aus Stellen (Zustände, Bedingungen) und Transitionen (Ereignisse, Veränderungen)
- Markierte Stellen geben an, welche Zustände/Bedingungen gelten
- Stellen/Transitionen-Netze erweitern Bedingungs/Ereignis-Netze um Kantengewichte und Kapazitäten
- Verhaltenseigenschaften wie Lebendigkeit oder Beschränktheit können durch den Erreichbarkeitsgraph oder durch strukturelle Analyse bestimmt werden

Literatur

- Petri, C. A.: Kommunikation mit Automaten, Universität Bonn 1962
- Reisig, W.: Petri-Netze – Eine Einführung, 2. Auflage. Springer, Berlin, Heidelberg 1986
- Reisig, W.: Systementwurf mit Petri-Netzen. Springer, Berlin, Heidelberg 1985
- Rosenstengel, B., Winand, U.: Petri-Netze – Eine anwendungsorientierte Einführung, 4. Auflage. Vieweg, Braunschweig 1991
- Leszak, M., Eggert, H.: Petri-Netz-Methoden und Werkzeuge. IFB 197, Springer, Berlin, Heidelberg 1988.