

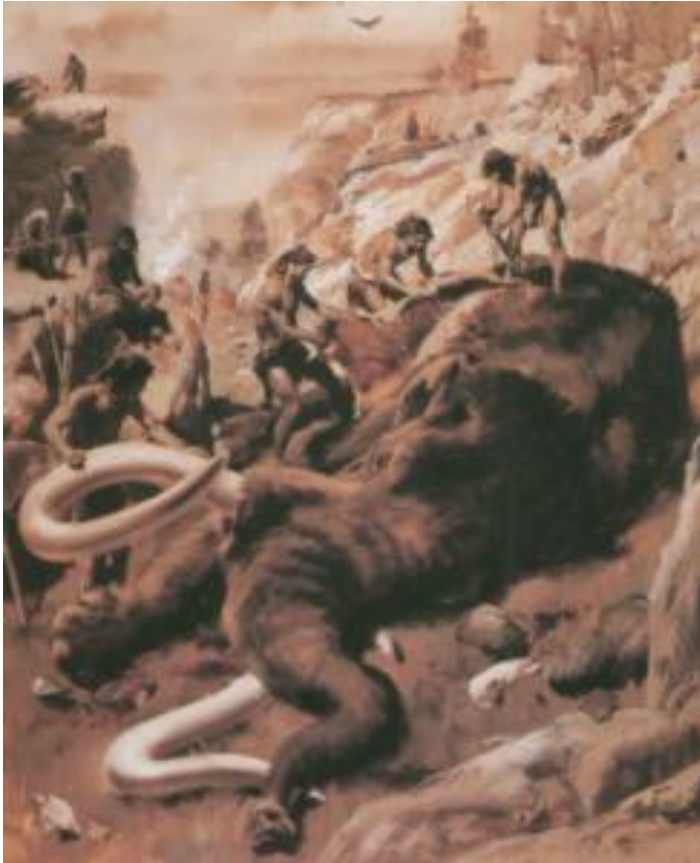
Prozessmodelle

Prof. Dr. Heiner Stuckenschmidt

Basiert auf Material von

Prof. Wil van der Aalst (TU Eindhoven)

Arbeitsteilung gestern...



... und heute



Vor- und Nachteile

- Vorteil: Spezialisierung auf einzelne Tätigkeiten erhöht die Effizienz
- Nachteil: erhöhter Koordinationsaufwand der einzelnen Arbeitsschritte in Arbeitsprozessen (Workflows)

Workflows

- Prozess, in dem ein bestimmtes ‚Produkt‘ entsteht.
- Die Erzeugung eines einzelnen Produktes ist ein **Case**
- Hierzu wird ein **Prozess** durchlaufen, der einen definierten Anfang und ein definiertes Ende hat.
- Ein Prozess besteht aus **Aktivitäten**, die anhand bestimmter **Bedingungen** ausgeführt werden.
- Mit Hilfe von Bedingungen entstehen Kontrollstrukturen: **Abfolge, Auswahl, Parallelität und Wiederholung**
- Eine Aktivität ist eine abgeschlossene Handlung, die von oder mit Hilfe von **Ressource** ausgeführt wird.

Beispiel für einen Workflow

Zubereitung

Arbeitszeit: ca. 20 Min. **Ruhezeit:** ca. 1 Std. / **Schwierigkeitsgrad:** normal / **Kalorien p. P.:** ca. 450 kcal

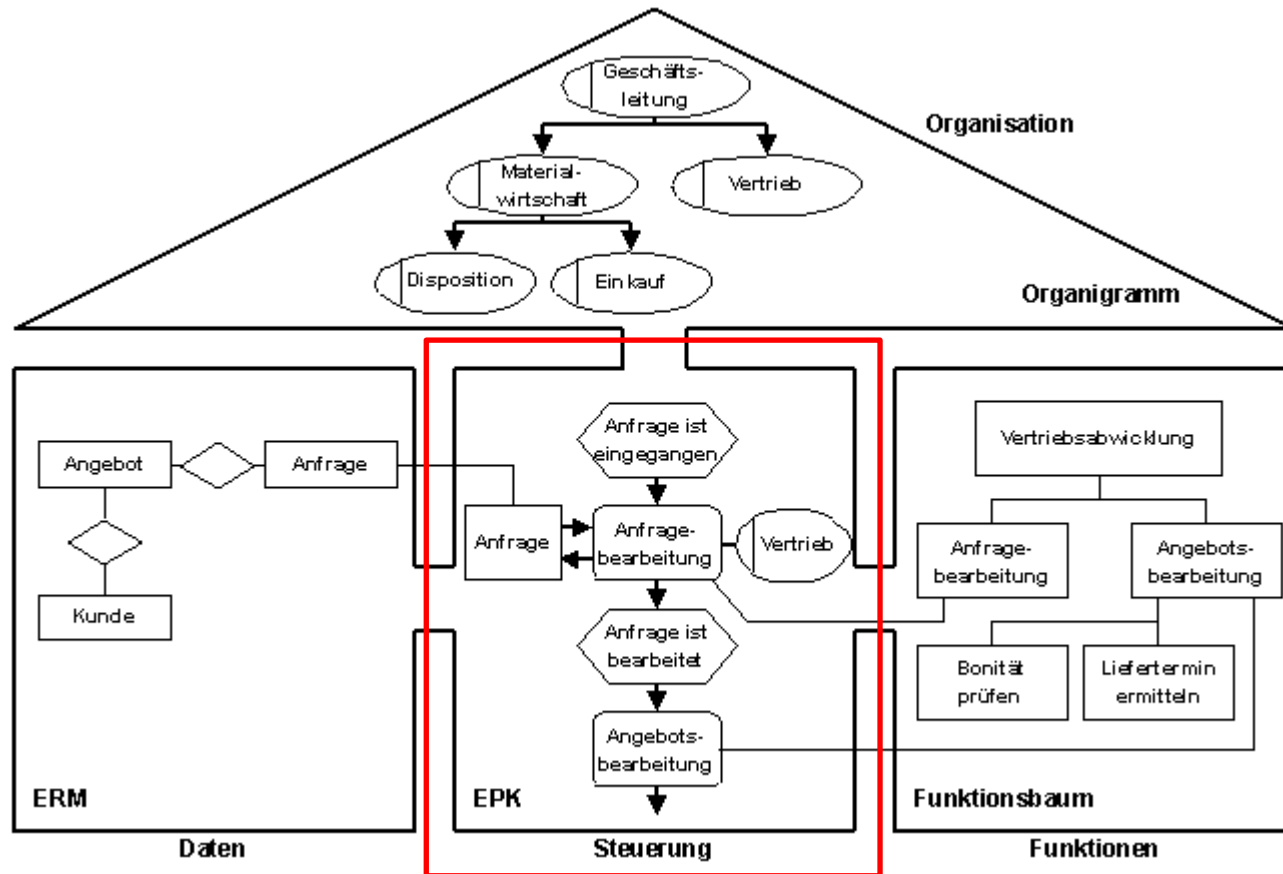
Grieß, Mehl und Ei mit den Knethaken des Handrührgerätes zu streuselartigen Krümeln verrühren. Nun nach und nach in kleinsten Mengen (teelöffelweise) kaltes Wasser hinzugeben und weiter kneten, bis kein trockener Grieß oder trockenes Mehl mehr zwischen den "Streuseln" zu sehen ist. Zum Schluss die Streusel mit der Hand zu einem festen Teig zusammenkneten. Eine Kugel formen und in Folie mindestens eine Stunde kalt stellen.

Nach der Ruhezeit den Teig in kleinen Portionen (ca. 80 - 100 g) durch die Nudelmaschine geben. Einmal falten und wieder durchgeben, insgesamt ca. 12 - 15 Mal! Der Teig hat die richtige Konsistenz, wenn er in den ersten Durchgängen noch reißt, und erst so nach dem 4. oder 5. Durchlauf geschmeidig wird. Der Teig darf sich nicht klebrig anfühlen. Dann braucht es auch kein Mehl, um den Teig sauber durch die Maschine zu bekommen.

Jetzt erst den Abstand der Walzen bis zur gewünschten Dicke (2 - 3 mm) verringern und nach belieben schneiden oder formen. Die Pasta auf einem Ständer oder einem Küchentuch locker auslegen, bis sie verwendet wird.

In viel und kräftig gesalzenem Wasser je nach Dicke ca. 2-3 Min. kochen. Die Zeit starten, wenn die Pasta in den Topf hinein gegeben wird, auch wenn das Wasser dann erst mal nicht mehr kocht!

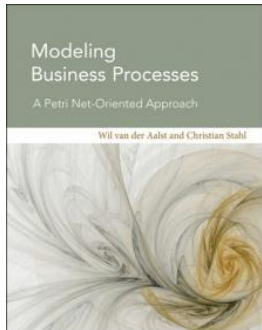
Erinnerung: Einbettung ins Unternehmen



In dieser Vorlesung

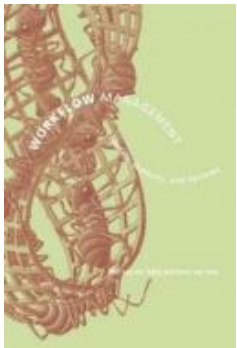
1. Petrinetze als generelles formales Modell für Prozesse
2. Automatische Analyse von Prozessen auf der Grundlage von Petrinetzen
3. Beschreibung von Workflows mit Petrinetzen
4. Analyse von Workflows
5. Gängige Prozessmodellierungssprachen und ihr Verhältnis zu Petrinetzen

(Klausur-) Relevante Literatur



Wil van der Aalst, Christian Stahl: Modeling Business Processes – A Petri-Net oriented Approach, MIT Press 2011.

- Kapitel 3: Basic Concepts of Petri Nets
- Kapitel 8.1 – 8.3: Analysing Petri Net Models
- Kapitel 4: Application of Petri Nets



Wil van der Aalst and Kees van Hee: Workflow Management – Models, Methods, and Systems MIT Press 2004.

- Kapitel 2.3: Mapping Workflow Concepts to Petri Nets
- Kapitel 4: Analyzing Workflows

Petri Netze: Grundlagen

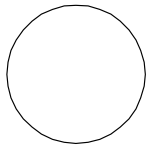
Basiert auf Material von
Prof. Wil van der Aalst (TU Eindhoven)

Klassische Petrinetze

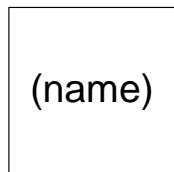
- Einfaches Prozessmodell
 - Elemente: **Stellen (places)**, **Transitionen (transitions)**, **Kanten (arcs)** und **Marken (tokens)**.
 - Graphische und mathematische Darstellung
 - Formale Semantik ermöglicht Prozessanalyse
 - Zahlreiche Tools für die automatische Analyse
 - Grundlage für praktische Modellierungssprachen
- Entwicklung:
 - Carl Adam Petri (1962, Doktorarbeit)
 - 60er und 70er Jahre: hauptsächlich Theorie
 - Seit den 80ern Fokus auf praktische Anwendungen

Graphische Darstellung

(name)



place



(name)

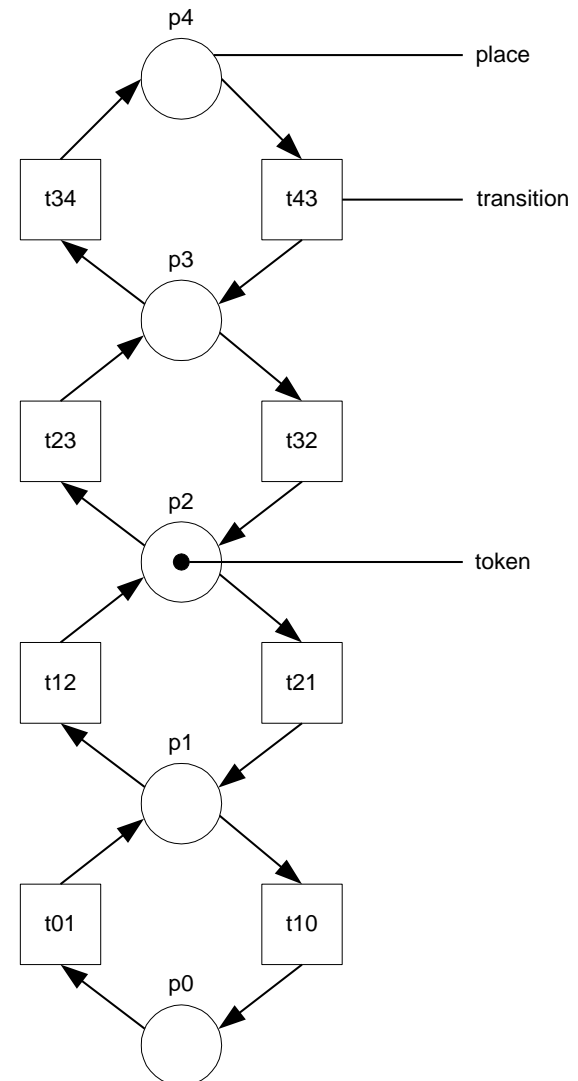
transition



arc (directed connection)

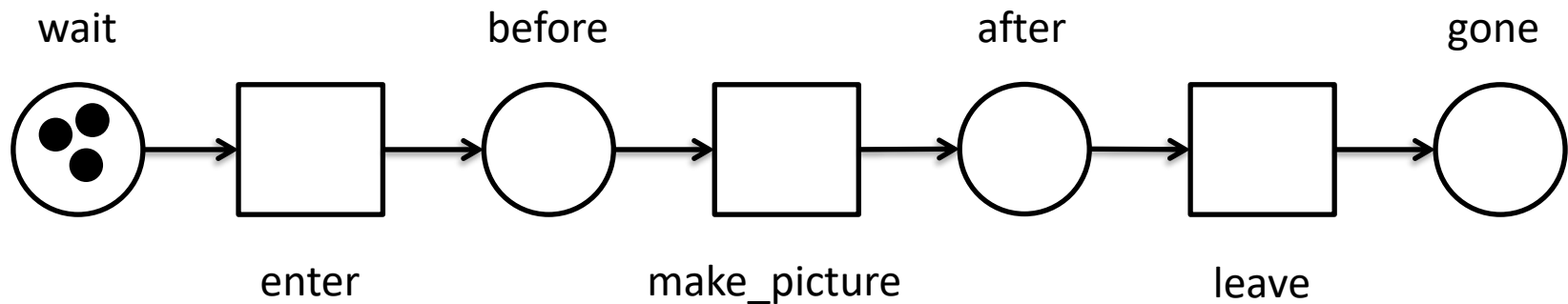


token



Beispielprozess

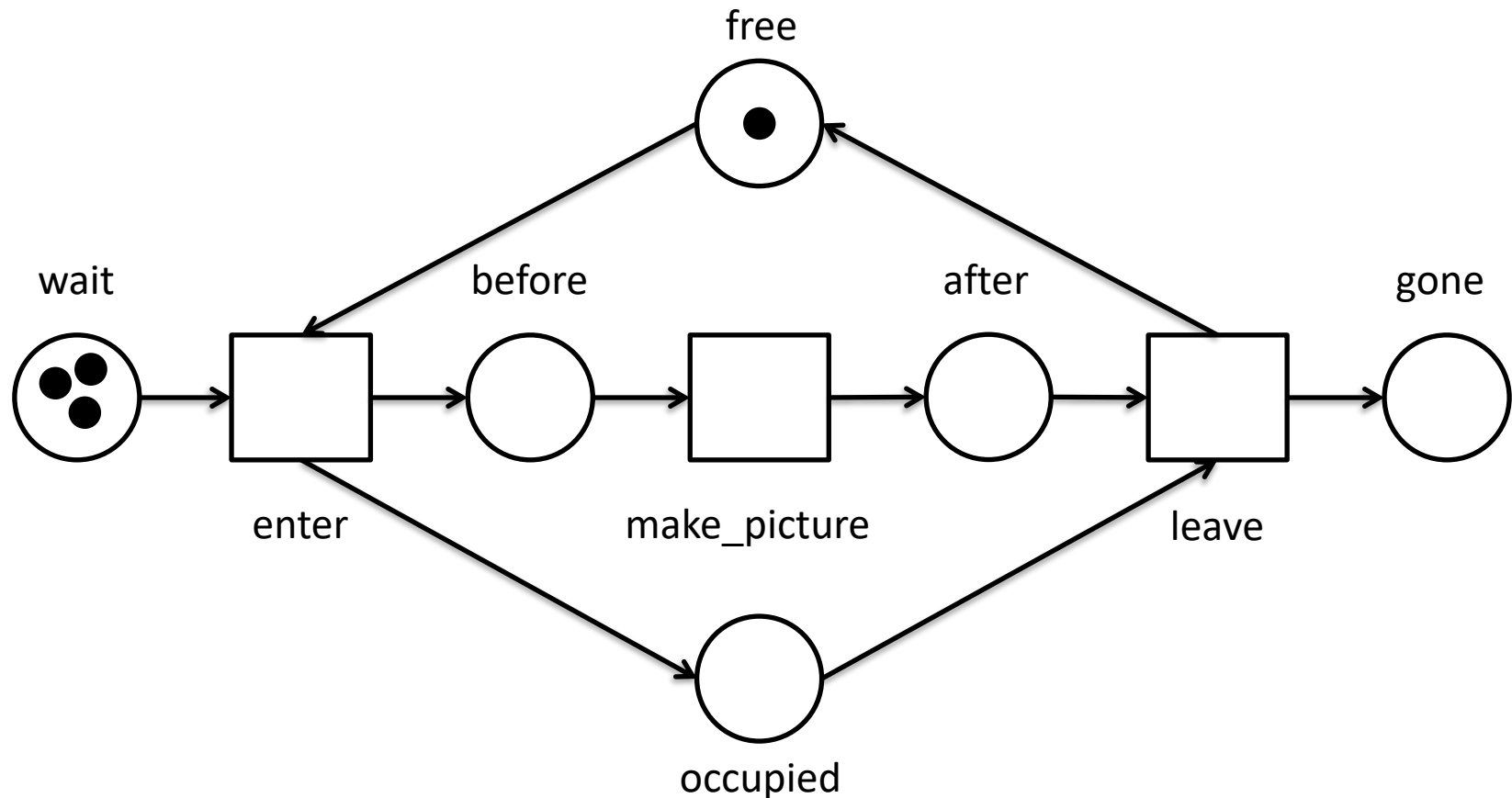
Röntgenvorgang in einem Krankenhaus:



Ist das Modell vernünftig ?

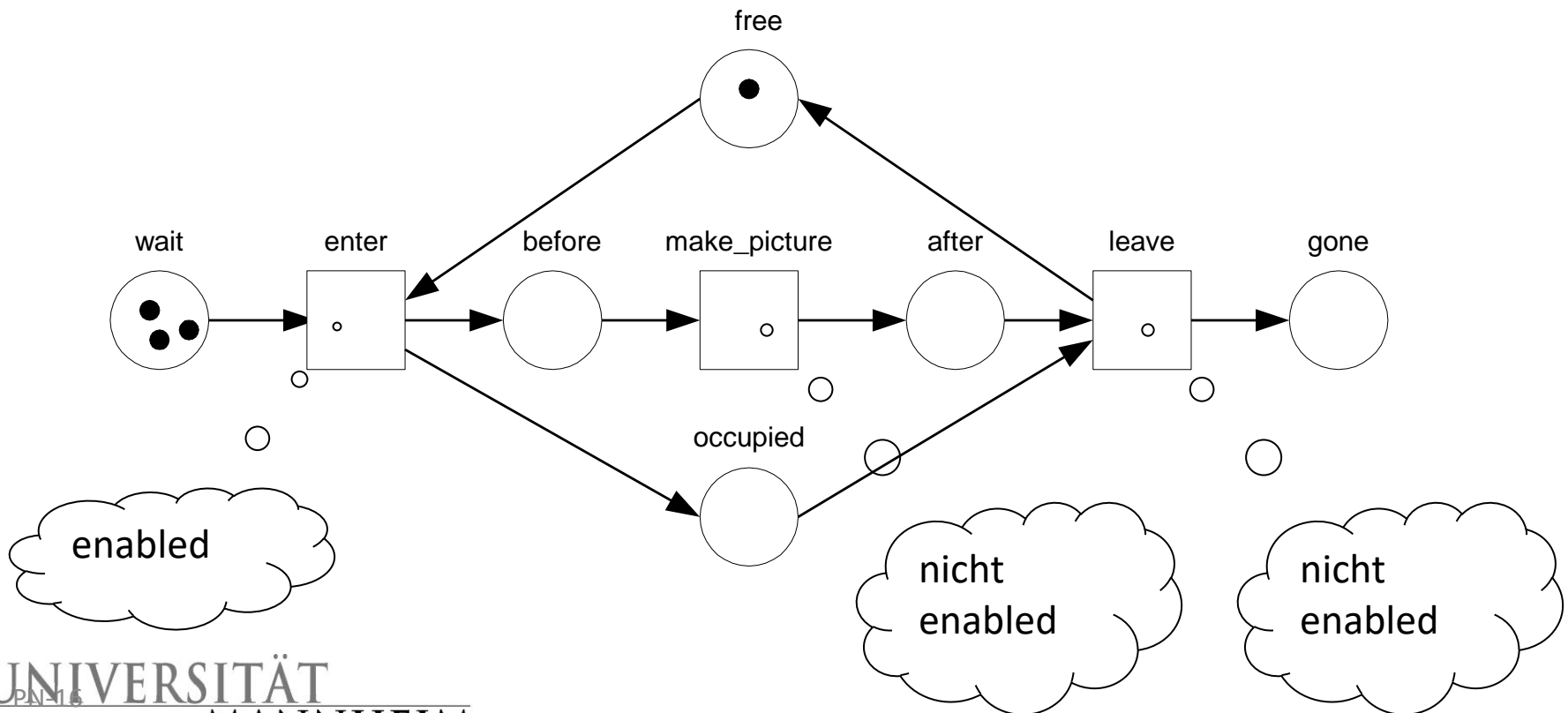
Beispielprozess

Wir müssen sicherstellen, dass nur eine Person zur Zeit im Röntgenraum ist!



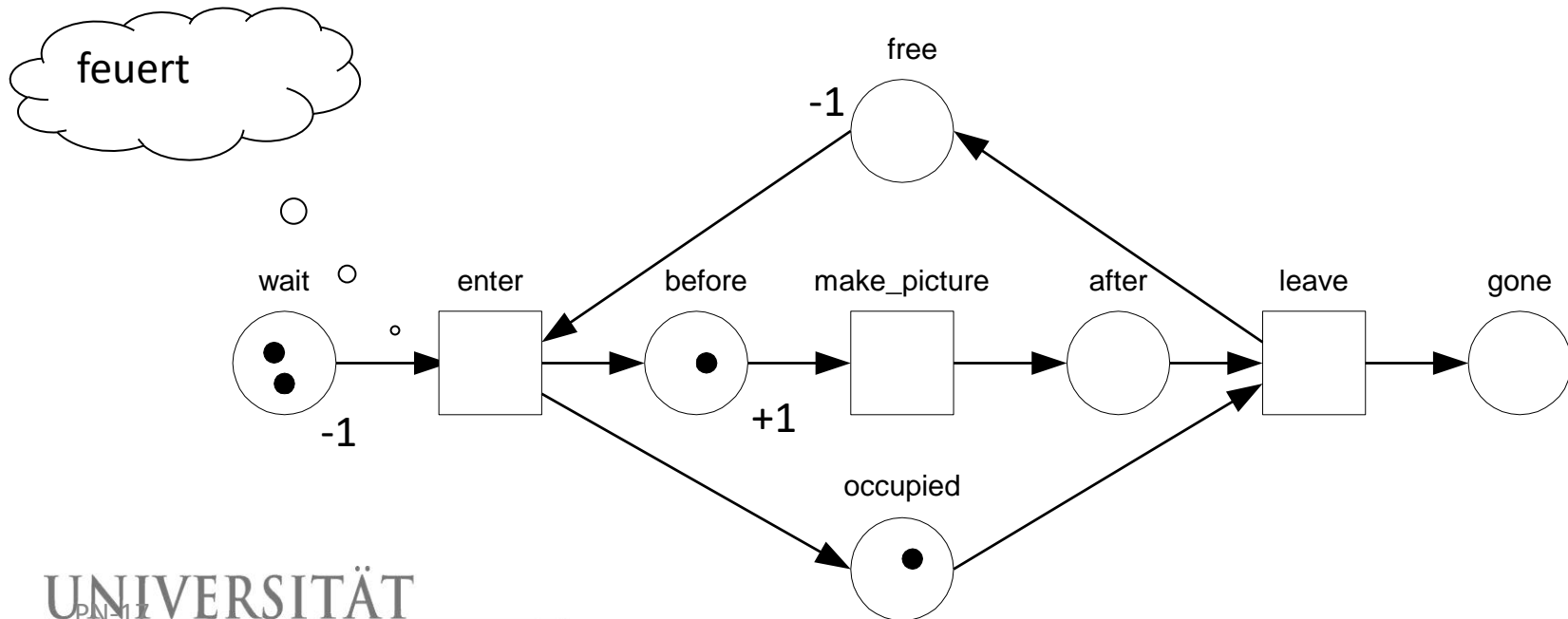
‘Enabled Transitions’

- Eine Transition ist **enabled** wenn jede ihrer Eingangsstellen ein Token enthält



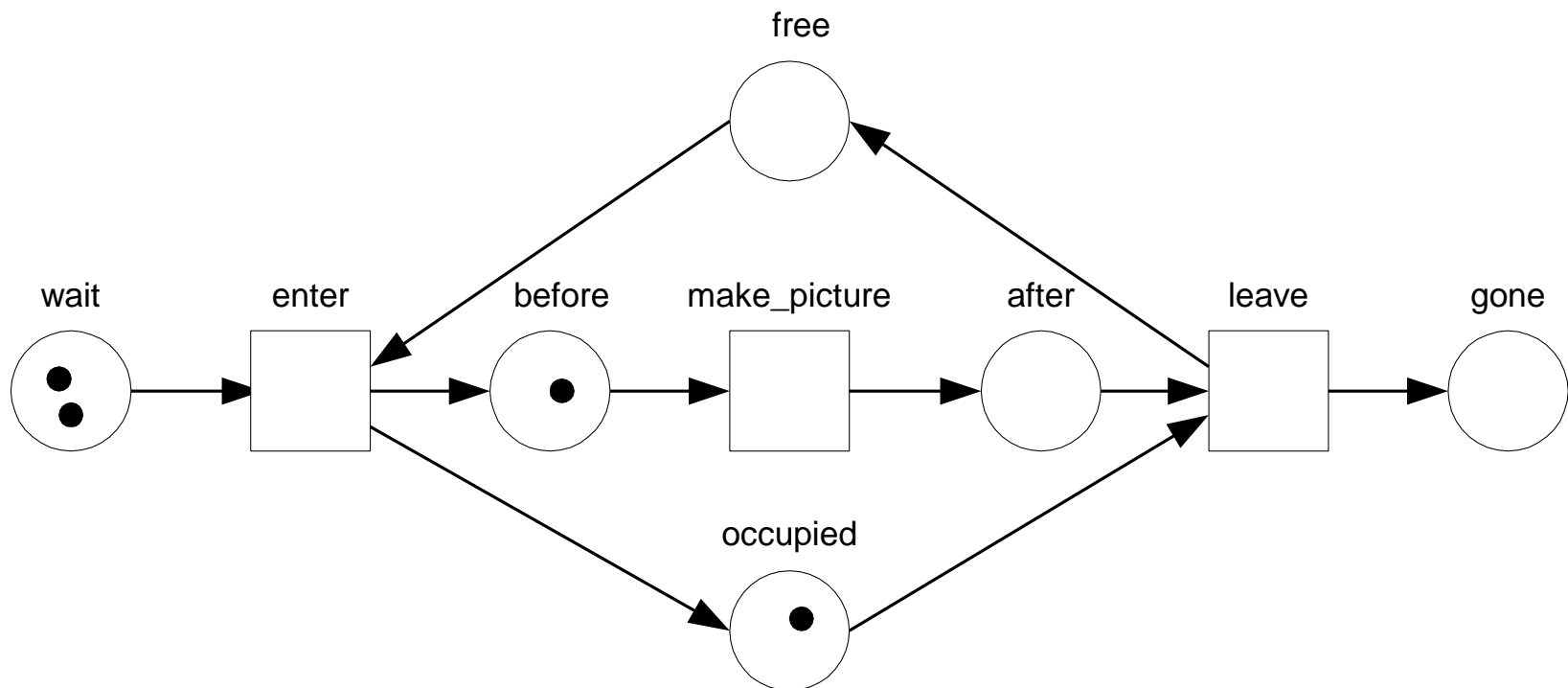
Feuern von Transitionen

- Wenn eine Transition **enabled** ist, kann sie **feuern**
- Wenn eine Transition feuert, wird
 - aus jeder Eingangsstelle ein Token **konsumiert** und
 - In jeder Ausgangsstelle wird ein Token **produziert**



Prozessausführung

- Das Netz wird von Zustand zu Zustand überführt, indem jeweils eine Transition, die enabled ist feuert



Anmerkungen

- Das feuern ist eine **atomare** Handlung.
- Auch wenn mehrere Transitionen enabled sind, feuert immer nur eine zur Zeit.
- Die Zahl der Tokens kann sowohl zu- als auch abnehmen, je nach Anzahl von Ein- und Ausgabestellen
- Die Netzwerkstruktur ändert sich nicht.
- Ein **Zustand (marking)** eines Petri-Netzes wird durch die Verteilung von Token auf Stellen definiert

Formale Definition

Ein (klassisches) Petri-netz ist ein Tupel (P, T, I, O) mit:

- P ist eine endliche Menge von Stellen,
- T ist eine endliche Menge von Transitionen,
- $I : P \times T \rightarrow \mathbf{N}$ ist die Input Funktion
- $O : T \times P \rightarrow \mathbf{N}$ ist die Output Funktion.

Jedes Petrinetz Diagramm kann eindeutig in diese Darstellung übersetzt werden.

Formale Definition (2)

Der Zustand (marking) eines Petri-Netzes (P, T, I, O) ist wie folgt definiert:

- $s: P \rightarrow \mathbf{N}$, i.e., eine Funktion, die jede Stelle auf eine natürliche Zahl $\{0, 1, 2, \dots\}$ abbildet, welche die Anzahl der Token auf der Stelle definiert.

Ausführungssemantik

Die Transition t ist im Zustand s enabled genau dann, wenn:

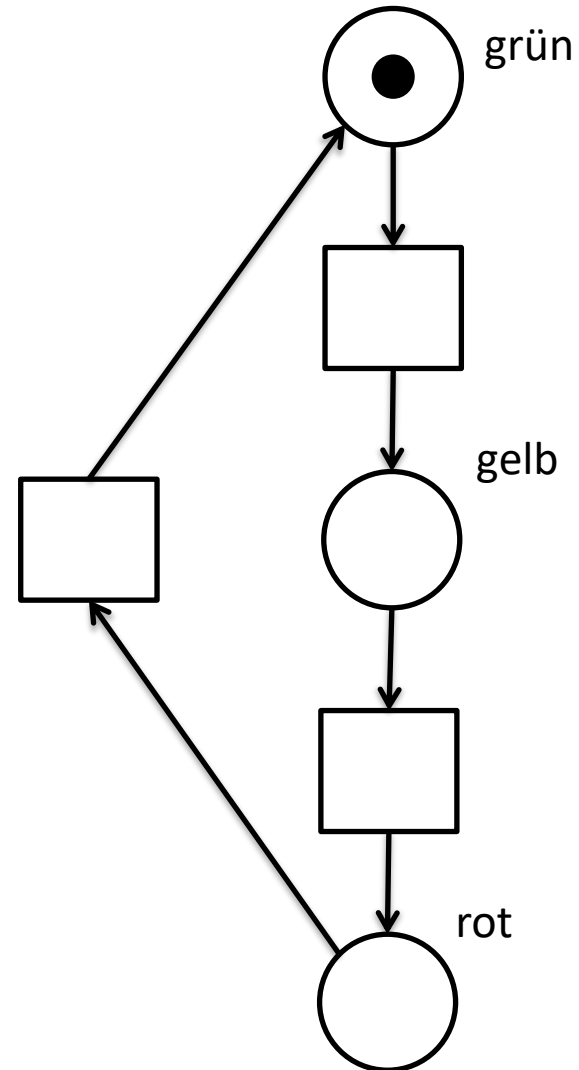
$$\forall p \in P: s(p) \geq l(p,t)$$

Die Transition t kann feuern und der daraus resultierende Zustand ist wie folgt definiert:

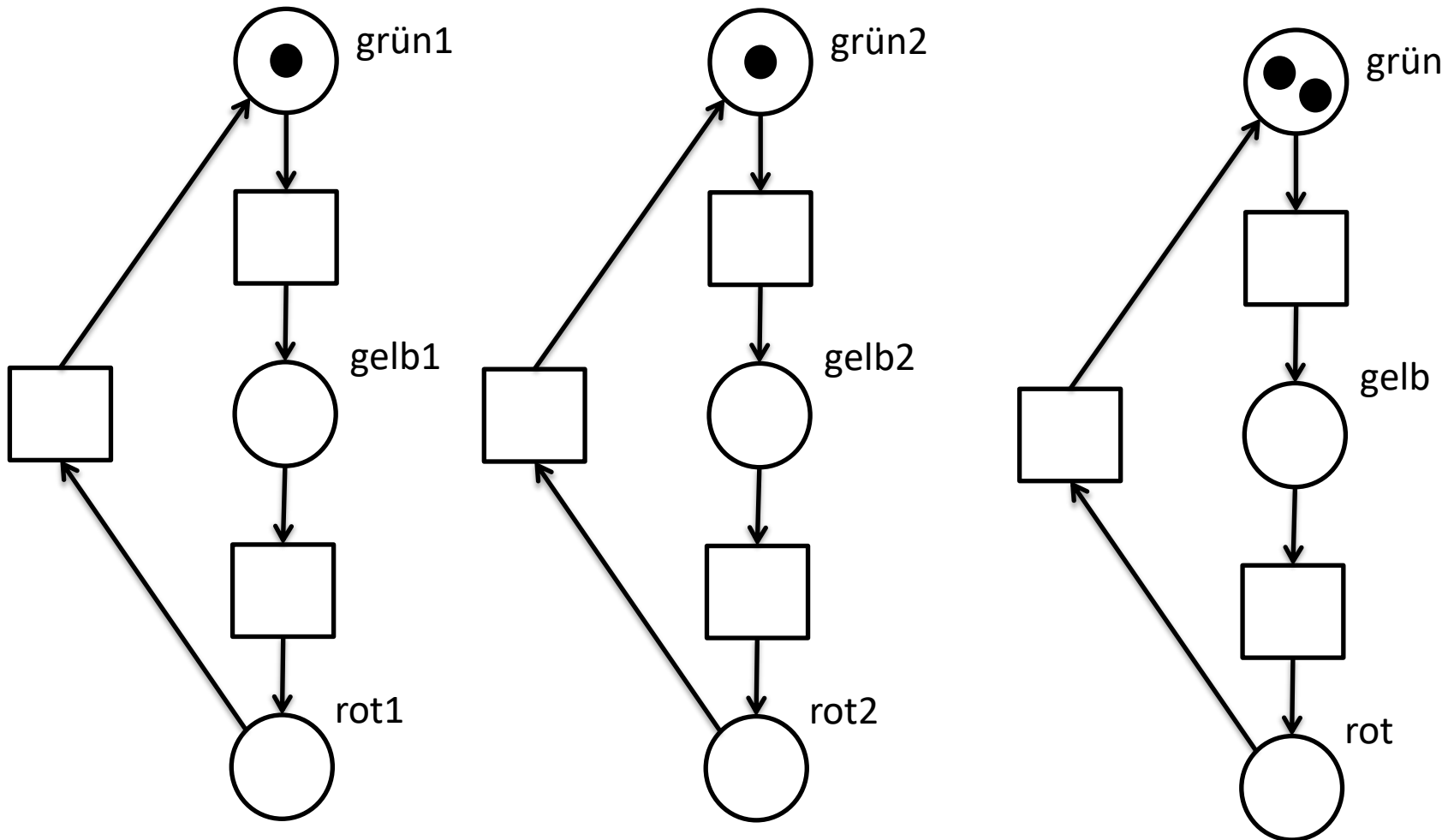
$$\forall p \in P: s'(p) = s(p) - l(p,t) + O(t,p)$$

Eine Folge von korrekten Zustandsübergängen heisst *run*

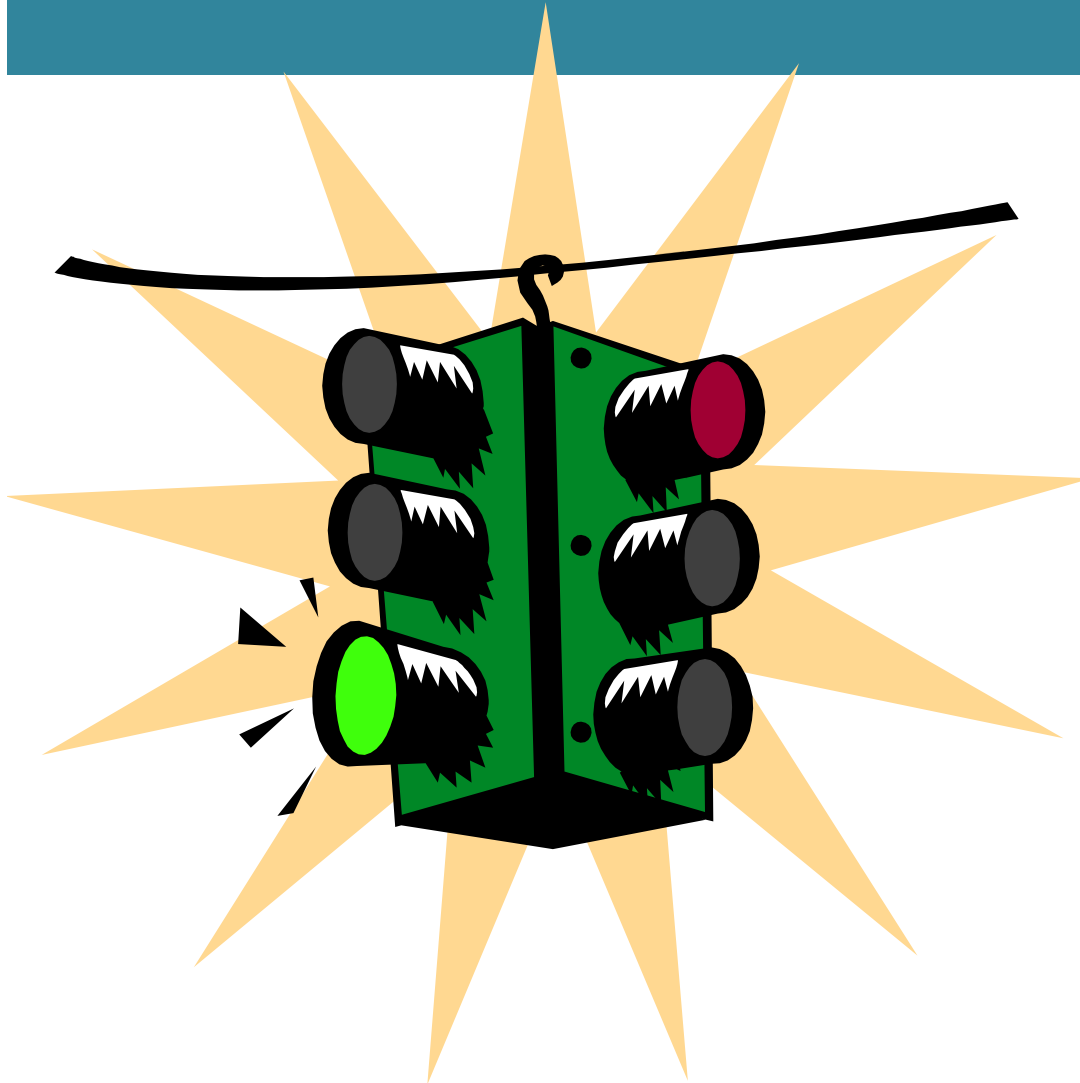
Beispiel: Verkehrsampel



Zwei Verkehrsampeln

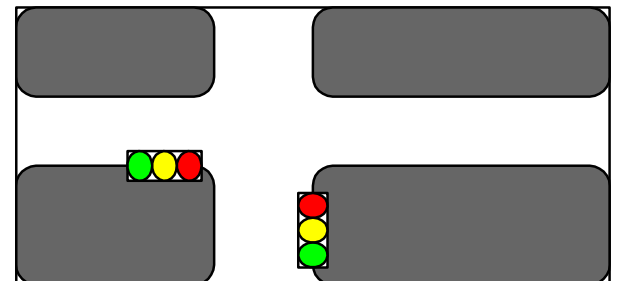


Koordination



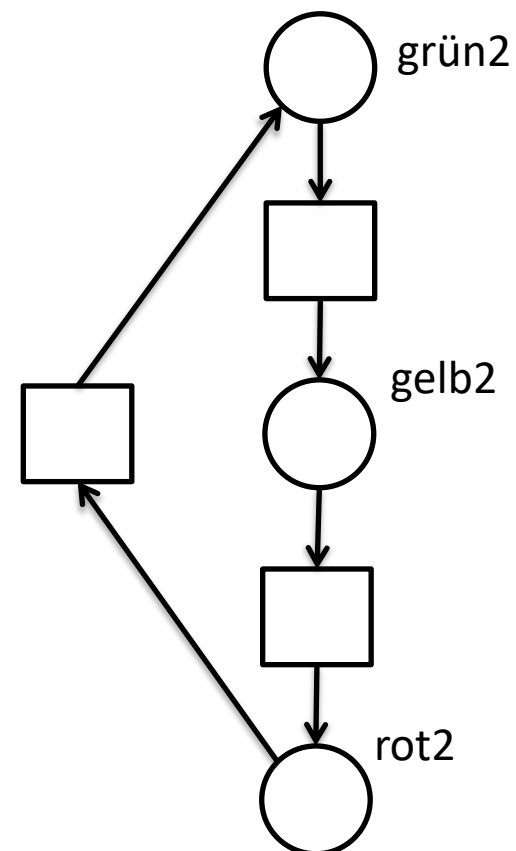
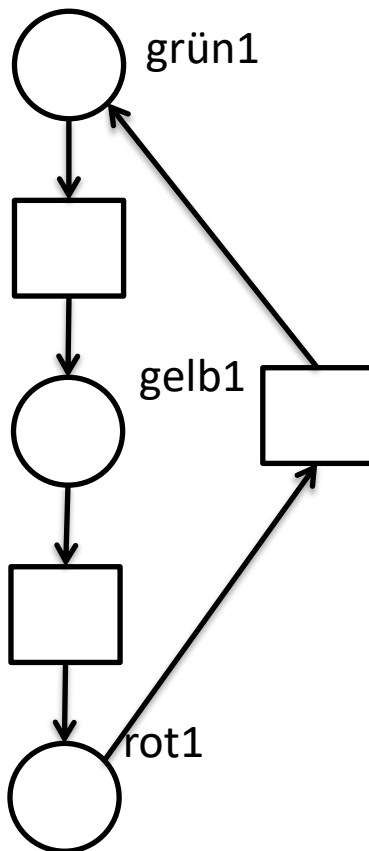
Ampeln an Kreuzungen müssen
Aufeinander abgestimmt sein !

1. Nicht beide Ampeln dürfen
gleichzeitig grün sein.
2. Ampeln sollen abwechselnd
grün sein.



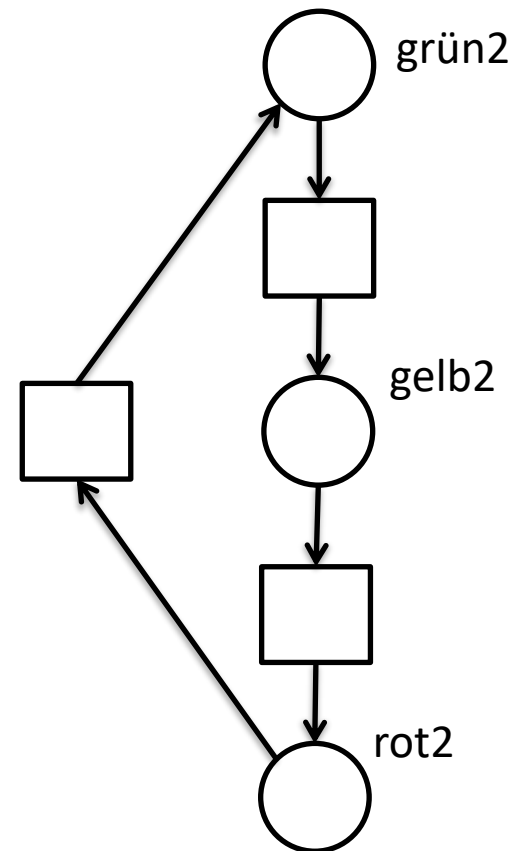
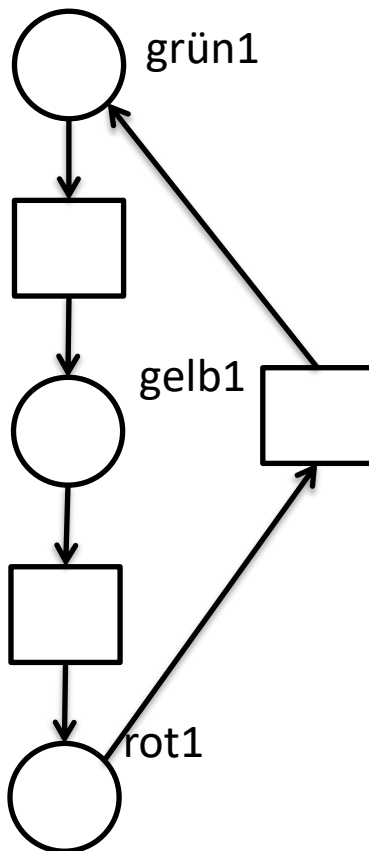
Lösung zu 1

- Nicht beide Ampel gleichzeitig grün:



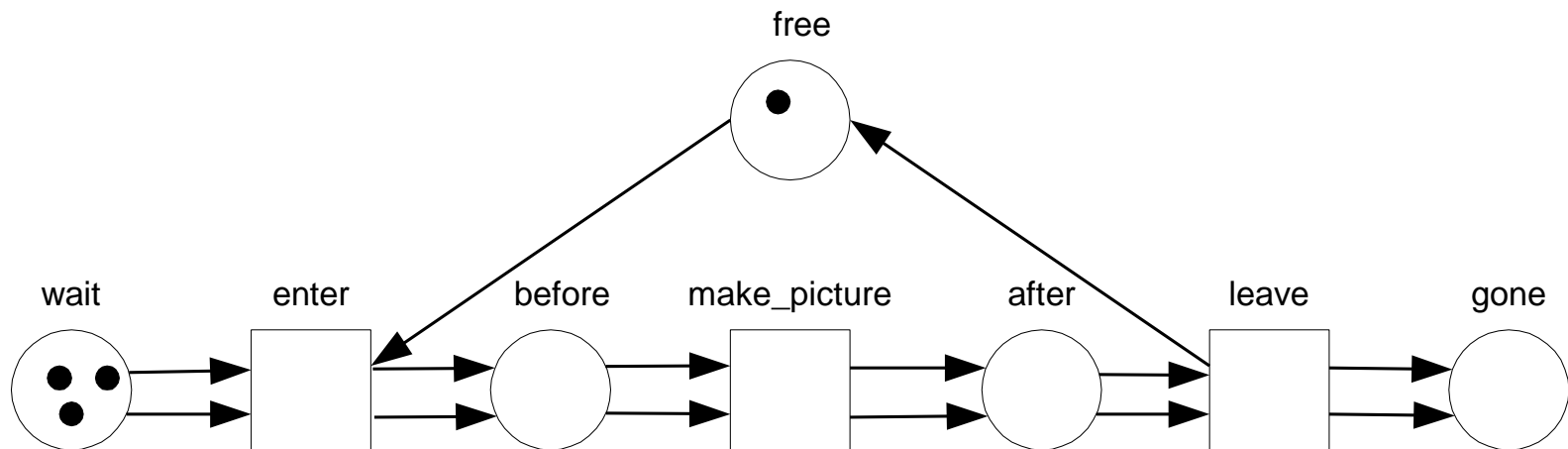
Lösung zu 2

- Ampeln abwechselnd grün:



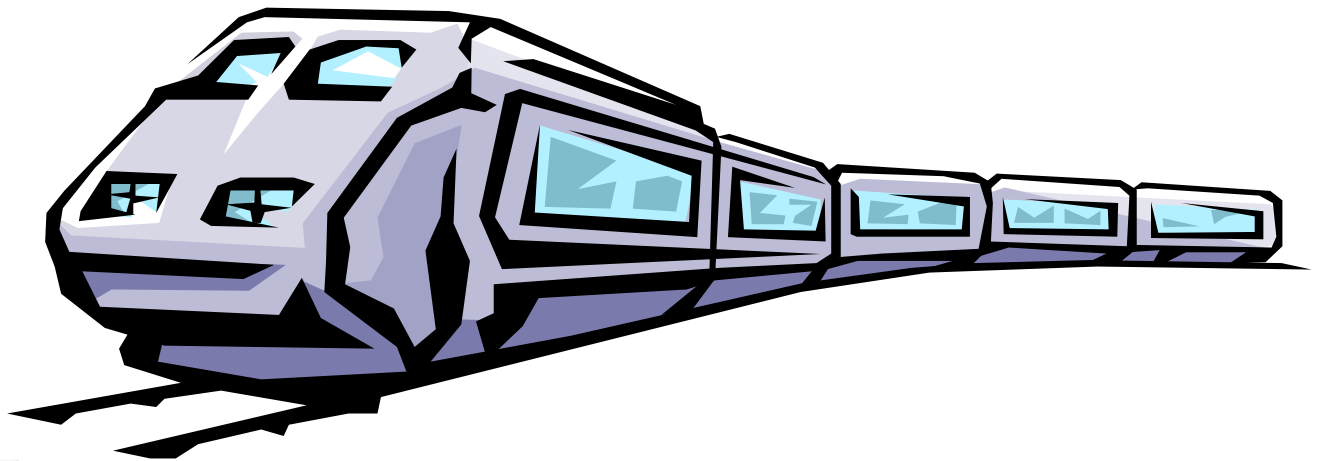
Parallele Kanten

- Anzahl der Kanten von Inputstellen legt fest,
 - wieviele Tokens vorhanden sein müssen, damit die Transition enabled ist
 - wieviele Token beim feuern konsumiert werden
- Anzahl der Kanten zu Outputstellen legt fest, wieviele Token produziert werden



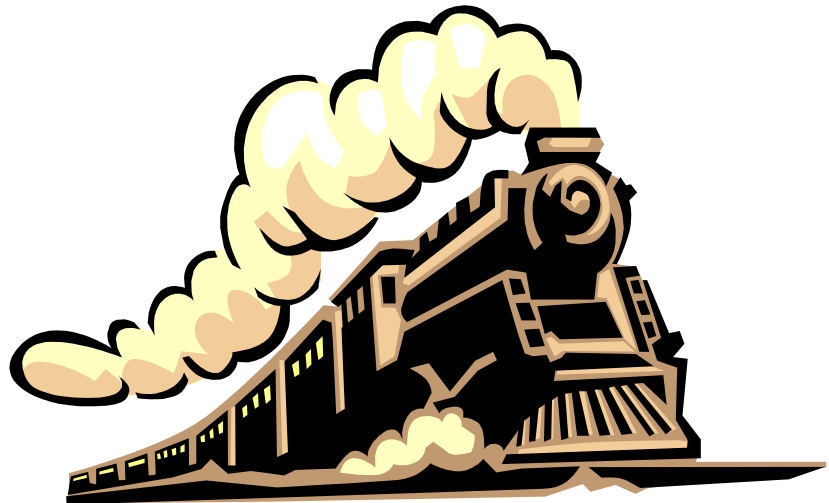
Übung: Zugsteuerung

- Gegeben sei ein Schienenkreis mit vier Segmenten (1,2,3,4) die nur in eine Richtung befahren werden können und 2 Zügen (A,B).
- Es dürfen keine zwei Züge gleichzeitig in einem Segment sein. Die Identität der Züge spielt keine Rolle



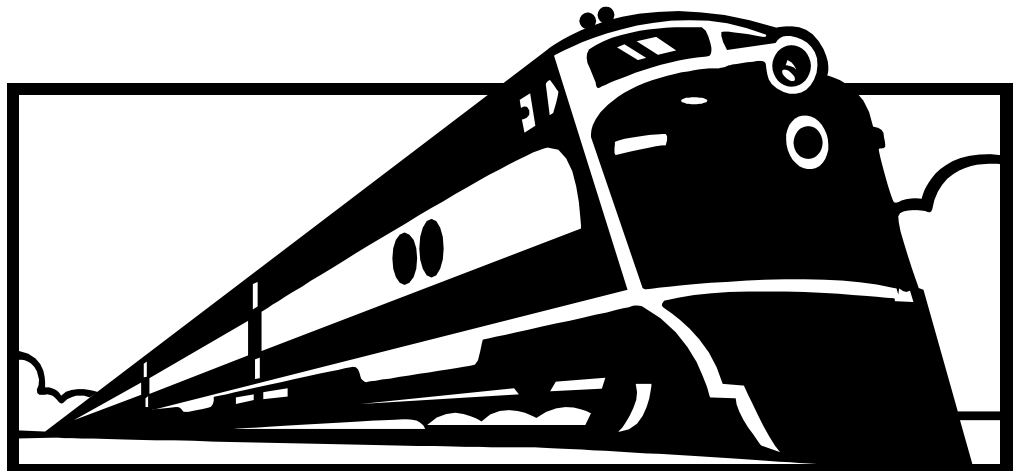
Übung: Zugsteuerung (2)

- Gegeben sei ein Schienenkreis mit vier Segmenten (1,2,3,4) die nur in eine Richtung befahren werden können und 2 Zügen (A,B).
- Es dürfen keine zwei Züge gleichzeitig in einem Segment sein. Die Identität der Züge soll diesmal berücksichtigt werden



Übung: Zugsteuerung (3)

- Gegeben sei ein Schienenkreis mit vier Segmenten (1,2,3,4) die nur in eine Richtung befahren werden können und 2 Zügen (A,B).
- Es dürfen keine zwei Züge gleichzeitig in einem Segment sein und auch das jeweils nächste Segment muss frei sein. Die Identität der Züge spielt keine Rolle



Übung: Zugsteuerung (4)

- Gegeben sei ein Schienenkreis mit vier Segmenten (1,2,3,4) die nur in eine Richtung befahren werden können und 2 Zügen (A,B).
- Segmente sind **frei**, **belegt** oder **reserviert**. Belegte Segmente können nicht reserviert werden. Züge müssen ein Segment reservieren, bevor sie es belegt können

