

# **Modellierung und Analyse von Petrinetzen am Beispiel einer Fertigungsanlage**

**Proseminar - Arbeit**

des Studiengangs Angewandte Informatik International Business Competence  
an der Dualen Hochschule Baden-Württemberg Mannheim

von

**Arwed Mett, Tobias Dorra, Dominik Gehrig**

25.06.2015

**Bearbeitungszeitraum**

7 Wochen

**Kurs**

AIBC 2014

**Matrikelnummer**

Arwed Mett 4278042

Dominik Gehrig 6377440

Tobias Dorra 6139224

**Gutachter**

Herr Ulf Runge

# Erklärung

Ich erkläre hiermit ehrenwörtlich:

1. dass ich meine Proseminar - Arbeit mit dem Thema *Modellierung und Analyse von Petrinetzen am Beispiel einer Fertigungsanlage* ohne fremde Hilfe angefertigt habe;
2. dass ich die Übernahme wörtlicher Zitate aus der Literatur sowie die Verwendung der Gedanken anderer Autoren an den entsprechenden Stellen innerhalb der Arbeit gekennzeichnet habe;
3. dass ich meine Proseminar - Arbeit bei keiner anderen Prüfung vorgelegt habe;
4. dass die eingereichte elektronische Fassung exakt mit der eingereichten schriftlichen Fassung übereinstimmt.

Ich bin mir bewusst, dass eine falsche Erklärung rechtliche Folgen haben wird.

Mannheim, 25.06.2015

Arwed Mett, Tobias Dorra, Dominik Gehrig

## **Zusammenfassung**

Diese Proseminar-Arbeit führt in die Thematik der Petrinetze ein. Petrinetze sind ein Formalismus zur Beschreibung vernetzter Systeme. Ihre Stärken liegen besonders bei der Darstellung nebenläufiger Prozesse. Transitionen und Plätze sind die Komponenten eines Petrinetzes. Während Transitionen die elementaren Aktionen des modellierten Systems abbilden, können Plätze Objekte oder Informationen speichern beziehungsweise lagern. Nachdem dem Leser die grundlegende Funktionsweise von Petrinetzen bekannt ist, wird auf die Bestimmung von Platzinvarianten als exemplarische Analyse-methode eingegangen. Aus Platzinvarianten lassen sich Gleichungen ableiten, die zur weiteren Analyse oder zur Verifikation des Modells genutzt werden können. Anschließend wird das vermittelte Wissen anhand einer Fallstudie noch einmal in ein größeres Beispiel gefasst.

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Allgemeines . . . . .	1
1.2	Historie . . . . .	2
<b>2</b>	<b>Theorie</b>	<b>3</b>
2.1	Aufbau von Petrinetzen . . . . .	3
2.2	Ablaufregeln . . . . .	4
2.3	Weitere Ausdrucksmöglichkeiten . . . . .	8
2.4	Variationen . . . . .	8
2.5	Analyse von Petrinetzen mit Platzinvarianten . . . . .	9
<b>3</b>	<b>Fallstudie</b>	<b>15</b>
3.1	Modellierung . . . . .	15
3.2	Analyse und Verifikation . . . . .	18
<b>4</b>	<b>Fazit</b>	<b>21</b>
	<b>Abbildungsverzeichnis</b>	<b>i</b>
	<b>Tabellenverzeichnis</b>	<b>ii</b>
	<b>Literatur</b>	<b>iii</b>

# 1 Einleitung

## 1.1 Allgemeines

Der Formalismus der Petri-Netze ging aus der Theoretischen Informatik hervor und diente ursprünglich lediglich der Beschreibung von verteilten technischen Systemen.

Meist wurden technische Systeme mit Hilfe eines Automaten beschrieben. Mit der ansteigenden Komplexität großer Systeme wächst die Anzahl der Zustände potentiell exponentiell an. Dies wird besonders bei der Modellierung von verteilten Systemen schnell unübersichtlich. Um solche Systeme trotzdem modellieren zu können, werden sie mit Hilfe von Petri-Netzen beschrieben. Der Formalismus der Petri-Netze beschränkt sich heutzutage nicht mehr nur auf die Themenbereiche der Informatik, sondern hat sich auch als bewährtes Mittel in der Biologie und im Geschäftsprozessmanagement etabliert. Die große Stärke der Petri-Netze besteht darin, nebenläufige Prozesse darzustellen, wodurch beispielsweise so genannte „deadlocks“<sup>1</sup> in frühzeitig erkannt werden können und die Laufzeit eines Systems durch Analyse verkürzt werden kann. Außerdem lassen sich komplexe Systeme durch eine Modellierung besser verstehen.

---

<sup>1</sup> (vgl. [15a])

## 1.2 Historie

Carl Adam Petri, geboren am 12. Juli 1926 in Leipzig und gestorben am 2. Juli 2010, ist der Schöpfer der Petrinetze.

Bereits im Alter von 13 Jahren versuchte er sein Wissen über chemischen Prozesse zu visualisieren und diese Prozesse abzubilden.

Im Jahre 1941 begann er mit seinen theoretischen Forschungen, die damals durch den Entwickler, der ersten frei programmierbaren Computer, Konrad Zuse (1910-95), stark beeinflusst worden sind.

Zu seiner Studienzeit und der daraufhin folgenden Zeit als Assistent an der technischen Universität Hannover (1950-56) und Bonn (1956-62) merkte er, dass es nötig ist einen Formalismus zur Lösung elementarer Probleme der Informatik zu schaffen. Dabei wollte er eine Alternative zu Automaten schaffen. Aus diesem Grund lehnte er Modelle mit einem globalen Zustand ab und entwickelte ein aus dezentralisierten Zuständen bestehendes Modell. Diese Ideen beschrieb er in seiner Dissertation über „Kommunikation mit Automaten“. Auch wenn Petri nicht sehr viel von Petrinetzen in dieser Dissertation erwähnt hatte, so hat er jedoch deren Eigenschaften in mathematische Notationen integriert. Petrinetze, so wie wir sie heute kennen, hat Petri zum ersten Mal bei einer Konferenz im Jahre 1965 erwähnt.

Jedoch gab es für seine Theorien zu seiner Zeit wenig Anwendungen, weshalb sie in der Wissenschaftsgemeinde keinen Anklang fanden. Dies wandelte sich mit der Entwicklung parallelisierter Computer-Systeme und dezentralisierter Netzwerke. Im Jahre 2008 erhielt Petri den „Computer Pioneer Award“.<sup>1</sup>

---

<sup>1</sup> (Vgl. [15b])

## 2 Theorie

### 2.1 Aufbau von Petrinetzen

Im Prinzip können Petrinetze als Graphen aufgefasst werden. Diese Graphen besitzen allerdings besondere Knoten, die als Komponenten bezeichnet werden. Im folgenden werden die Komponenten eines Petrinetzes beschrieben.

**Plätze** sind die passiven Komponenten eines Petrinetzes. Sie repräsentieren Zustände oder lagern Objekte. Ein Platz wird durch einen Kreis dargestellt:



Abbildung 2.1: Beispiel für einen leeren Platz und einen Platz mit einer (abstrakten) Marke.

Die Objekte, die von Plätzen gelagert werden, werden allgemein als **Marken** bezeichnet. Grundsätzlich kann ein Platz beliebig viele Marken enthalten. Außerdem können in einem Modell unterschiedliche Typen von Marken verwendet werden. Eine Marke kann ein konkretes Objekt aus der realen Welt repräsentieren (Maschinen, Werkstücke, Menschen, ...). Es können jedoch auch abstrakte Marken verwendet werden, die dann meist als schwarzer Punkt dargestellt werden, wie in Abbildung 2.1 zu sehen. Abstrakte Marken sind sehr nützlich, um Zustände, in welchen sich das System befinden kann, zu modellieren.

**Transitionen** sind die aktiven Komponenten eines Petrinetzes. Sie beschreiben eine elementare Aktion, die Dinge erzeugen, transportieren, verändern oder vernichten <sup>1</sup>. Eine Transition wird durch ein Quadrat dargestellt:

Transitionen können zusätzlich noch mit einer Formel beschriftet werden. Diese ist eine Vorbedingung, die erfüllt sein muss, damit die Transition aktiviert werden kann. Wann

---

<sup>1</sup> [Her15]



Abbildung 2.2: Beispiel für eine Transition

genau eine Transition aktiviert ist und was das bedeutet, wird später in Kapitel 2.2 noch diskutiert.

**Kanten** sind selbst keine Komponenten, sondern stellen Beziehungen zwischen Komponenten her. Kanten werden durch einen Pfeil gekennzeichnet und verbinden immer eine Transition mit einem Platz oder umgekehrt. Niemals jedoch verbinden sie zwei Plätze oder zwei Transitionen miteinander.

Kanten, die von einer Transition auf eine Stelle zeigen, drücken aus, dass diese Transition Objekte erzeugt und sie in der Stelle ablegt. Kanten, die von einer Stelle auf eine Transition zeigen, drücken aus, dass diese Transition Objekte aus der Stelle entnimmt.

Eine Kante kann zusätzlich noch mit einem Term beschriftet werden. Das Ergebnis dieses Terms müssen Marken sein. Damit wird angegeben, wie viele und welche Marken die Transition aus der Stelle entnimmt beziehungsweise in ihr erzeugt. Im Normalfall ist die Beschriftung einfach ein konstanter Term, es sind jedoch auch Terme mit beliebigen Variablen erlaubt.

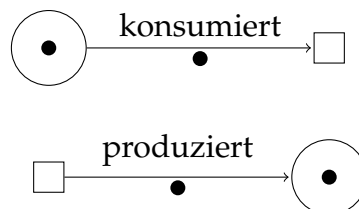


Abbildung 2.3: Beispiele für Kanten

## 2.2 Ablaufregeln

Der allgemeine Aufbau von Petrinetzen ist nun bekannt. Nach welchen Regeln sich ein Petrinetz genau verhält, wurde jedoch noch nicht festgelegt. In den folgenden Abschnitten sollen diese Regeln erläutert werden.

Als **Markierung** eines Petrinetzes bezeichnet man die Verteilung von Marken auf die Plätze. In Abbildung 2.4 und 2.5 sind zwei unterschiedliche Markierungen eines Petrinetzes dargestellt. Ein Petrinetz zusammen mit einer Anfangsmarkierung, in der sich das Petrinetz zu Beginn befindet, wird Systemnetz genannt.



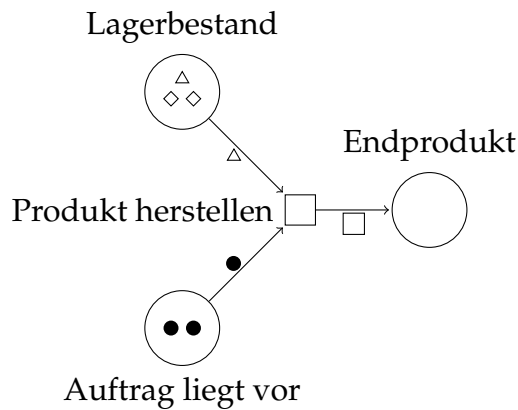


Abbildung 2.4: Markierung 1 eines Petrinetzes

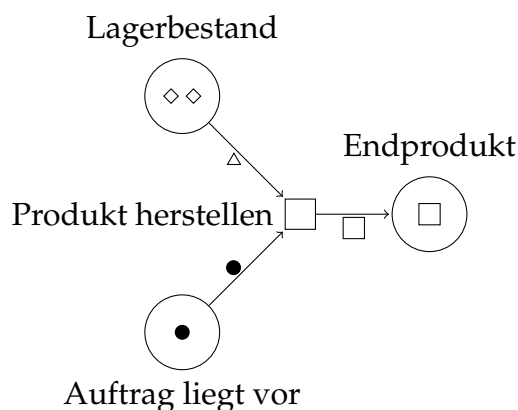


Abbildung 2.5: Markierung 2 eines Petrinetzes

Als nächstes wird erklärt, wann man eine Transition als **aktiviert** bezeichnet. Anschaulich gesehen ist eine Transition  $t$  in einer bestimmten Markierung aktiviert, wenn alle Plätze, aus denen  $t$  Marken entnimmt, dafür genug Marken beinhalten. Formal muss also jeder Platz, von dem eine Kante auf  $t$  zeigt, von jedem Markentyp genauso viele oder mehr Marken beinhalten als in der Beschriftung der Kante angegeben.

Im Petrinetz aus Abbildung 2.4 entnimmt die Transition *Produkt herstellen* eine dreieckige Marke aus dem Platz *Lagerbestand* und eine schwarze Marke aus *Auftrag liegt vor*. Beide Plätze enthalten die geforderten Marken. Also ist die Transition aktiviert.

In Abbildung 2.5 ist das selbe Petrinetz in einer anderen Markierung dargestellt. Hier enthält *Auftrag liegt vor* genug schwarze Marken. Aus dem Platz *Lagerbestand* entnimmt die Transition jedoch eine dreieckige Marke, aber in dem Platz sind nur Rautenförmige Marken enthalten. Daher ist die Transition in Markierung 2 nicht aktiviert.

Diese Vorgehensweise funktioniert für Kanten, die mit konstanten Termen beschriftet sind. Wenn im Term einer Kante Variablen vorkommen, muss zunächst eine Belegung

der Variablen mit echten Werten gefunden werden, für die diese Bedingung zutrifft. Außerdem muss die Formel in der Transition für die Variablenbelegung wahr sein. Wenn eine passende Variablenbelegung existiert, dann ist die Transition in der entsprechenden Markierung unter dieser Variablenbelegung aktiviert. Andernfalls ist die Transition nicht aktiviert.

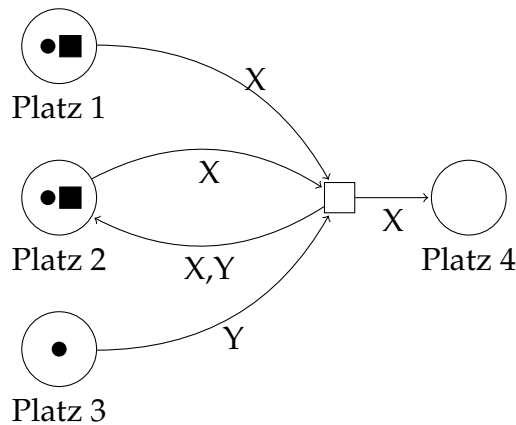


Abbildung 2.6: Ein Petrinetz mit Variablen in den Kantenbeschriftungen

Abbildung 2.6 zeigt ein Petrinetz mit Variablen in den Kantenbeschriftungen. In der gezeigten Markierung gibt es zwei Variablenbelegungen, für die die Transition aktiv ist:

- $X = \blacksquare$        $Y = \bullet$
- $X = \bullet$          $Y = \bullet$

Eine Variablenbelegung mit  $Y = \blacksquare$ , bei der die Transition aktiviert ist, kann es in der gezeigten Markierung nicht geben, denn in *Platz 3* ist ausschließlich eine kreisförmige Marke vorhanden.

Wenn eine Transition aktiviert ist, dann kann sie **eintreten**. Beim Ablauf eines Petrinetzes kann in einer Markierung immer nur eine Transition eintreten. Welche der aktivierten Transitionen eintritt, ist nicht deterministisch bestimmt. Durch das Eintreten einer Transition wird eine neue Markierung für das Petrinetz erzeugt. Dabei werden aus allen Plätzen, von denen eine Kante auf die Transition zeigen, die in der Kantenbeschriftung angegebenen Marken entfernt. Außerdem werden zu allen Plätzen, auf die eine Kante zeigt, die bei der Transition beginnt, die in der Kantenbeschriftung angegebenen Marken hinzugefügt.

Wenn die Kantenbeschriftungen Variablen enthalten, müssen zunächst die Werte aus der Variablenbelegung, für die die Transition aktiv ist, eingesetzt werden.

Wenn die Transition *Produkt herstellen* in Abbildung 2.4 eintritt, dann wird aus dem Platz *Lagerbestand* ein Marke  $\triangle$  entfernt und aus dem Platz *Auftrag liegt vor* wird eine

Marke  $\bullet$  entfernt. Außerdem wird zum Platz *Endprodukt* eine Marke  $\square$  hinzugefügt. So entsteht die Markierung, die in Abbildung 2.5 zu sehen ist.

Wie oben bereits erläutert, ist die Transition aus Abbildung 2.6 für die Variablenbelegung  $X = \blacksquare, Y = \bullet$  aktiviert. Wenn die Transition unter dieser Variablenbelegung eintritt, wird aus *Platz 3* eine Marke  $\bullet$  und aus *Platz 1* eine Marke  $\blacksquare$  entfernt. Aus *Platz 2* wird zunächst ebenfalls eine Marke  $\blacksquare$  entfernt und die Marken  $\blacksquare$  und  $\bullet$  wieder hinzugefügt. Außerdem wird zu *Platz 4* eine Marke  $\blacksquare$  hinzugefügt.

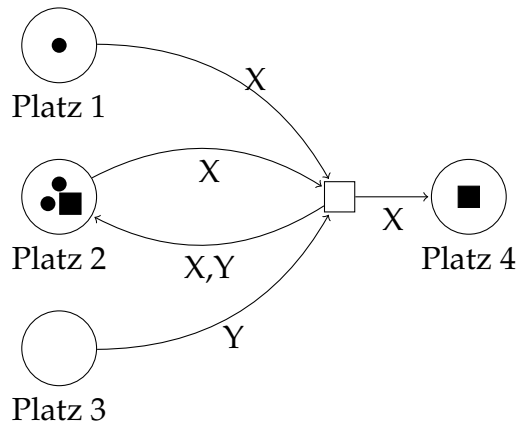


Abbildung 2.7: Das Petrinetz aus Abbildung 2.6 nach dem Eintreten der Transition.

Ein weiteres Beispiel für den Ablauf eines Systemnetzes sind **Zähler**. Diese erleichtern unter anderem die Modellierung von Lagern mit begrenzten Stückzahlen. Der Zähler, der in Abbildung 2.8 zu sehen ist, besteht aus einer Transition, die maximal 10 mal eintreten kann. Als Marke in *Platz 1* wird eine natürliche Zahl verwendet. Bei jedem Eintritt der Transition wird diese in Form der Variable  $x$  aus dem Platz entnommen und dafür die Zahl  $x - 1$  wieder hinzugefügt. So verringert sich die Zahl in dem Platz bei jedem Eintreten der Transition um 1. Die Transition hat außerdem als Vorbedingung, dass  $x > 0$  gelten muss. Deshalb kann sie insgesamt nur 10 mal eintreten.

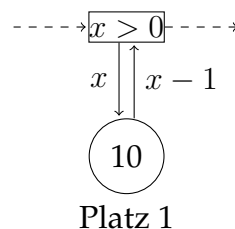


Abbildung 2.8: Ein Zähler

## 2.3 Weitere Ausdrucksmöglichkeiten

**Schnittstellen** ermöglichen auf Aktivitäten aus der Umgebung einzugehen. Mit Hilfe der bisher vorgestellten Bestandteile lassen sich bereits geschlossene Systeme modellieren. Sollen allerdings Interaktionen eines Geschäftsprozesses mit z.B. einem Kunden modelliert werden, benötigt man Schnittstellen. Diese werden durch eine Transition dargestellt.

**Kalte Transitionen** dienen der Modellierung von Transitionen, bei denen nicht klar ist ob sie eintreffen. Wird z.B. ein Prozess von einem Kunden durch dessen Bezahlung ausgelöst, ist im Vorhinein nicht klar ob dieses Ereignis jemals eintreffen wird. Ist das Gegenteil der Fall wird die Transition als „warm“ bezeichnet. Kalte Transitionen werden durch ein  $\epsilon$  in der Beschriftung kenntlich gemacht.

$\epsilon$

## 2.4 Variationen

Mit den zunehmenden Anforderungen an Petrinetze sind zahlreiche Variationen von Petrinetzen entstanden. Dabei werden die Varianten in zwei Kategorien gegliedert:

- low-level Petrinetze
- high-level Petrinetze

### 2.4.1 Elementare Petrinetze

Häufig werden Petri Netze dazu verwendet darzustellen, wo ein Kontrollfluss aktuell steht.<sup>1</sup> Will man einen Prozess genauer analysieren geht dies am einfachsten in dem man den Prozess abstrakt darstellt. Dafür bieten sich elementare Petrinetze an, die auch als elementare Systemnetze bezeichnet werden.

Bei einem elementaren Petrinetz werden die Marken auf den Kanten weggelassen. Außerdem gibt es nur eine Form von Marken, die in der Regel durch einen schwarzen Punkt • visualisiert wird. Dabei beschränkt sich ein elementares Petrinetz auf Transitionen, Plätze und Marken. Aufgrund der abstrakten Modellierung und geringen Anzahl an Komponenten gehören elementare Petrinetze zu den low-level Petrinetzen.

---

<sup>1</sup> (vgl. [Rei10, Seite 39])

## 2.4.2 Gefärbte Petrinetze

Gefärbte Petrinetze werden hauptsächlich zur Modellierung komplexer Systeme verwendet, da Sie zu der Gruppe der high-level Petrinetze gehören. Im Unterschied zu anderen Petrinetzen sind die gefärbten Petrinetze wesentlich übersichtlicher, können allerdings jederzeit wieder in ein low-level Petrinetz zurückgeführt werden. Dabei bieten die gefärbten Petrinetze weitere Funktionalitäten<sup>1</sup> wie z.B.:

- Typen
- Knotenfunktionen
- Typfunktionen
- Wächterfunktionen
- Kantenausdrucksfunktionen
- Initialisierungsfunktionen

## 2.5 Analyse von Petrinetzen mit Platzinvarianten

Es gibt verschiedene Möglichkeiten zur Analyse von Petrinetzen. Grundsätzlich kann man dabei zwischen der Analyse von Zustands- und Ablaufeigenschaften unterscheiden. Bei ersterem untersucht man Eigenschaften, die allgemein für alle im Systemnetz erreichbaren Markierungen gelten. Ablaufeigenschaften wiederum beschreiben Eigenschaften von einzelnen Markierungen, welche nach mehreren Schritten aus anderen Markierungen hervorgehen können.

In den folgenden Abschnitten wird exemplarisch die Bestimmung von Platzinvarianten in elementaren Systemnetzen beschrieben. Dabei geht es darum, Gleichungen zu finden, die für jede erreichbare Markierung gelten. Diese kann man dann beispielsweise für die Verifikation des Modells verwenden.

### 2.5.1 Einführung

Oft werden in Petrinetzen Material- oder Informationsflüsse modelliert. Dabei wird nichts erzeugt oder verbraucht, sondern das Material oder die Informationen werden bloß transportiert. Äquivalent dazu werden auch die entsprechenden Marken im Petrinetz nicht erzeugt oder zerstört, sondern nur von Platz zu Platz verschoben. Ein

---

<sup>1</sup> (vgl.: [Fun06])

Beispiel ist in Abbildung 2.9 zu sehen. Die Gesamtzahl der Marken bleibt hier immer gleich. Wenn man mit  $|Platz|$  die Anzahl der Marken im gegebenen Platz bezeichnet, kann man bei diesem Beispiel sagen, dass die Gleichung

$$|P_1| + |P_2| + |P_3| = 5$$

immer wahr ist. Ziel der Platzinvarianten ist es solche Gleichungen auch in größeren Petrinetzen zu finden.

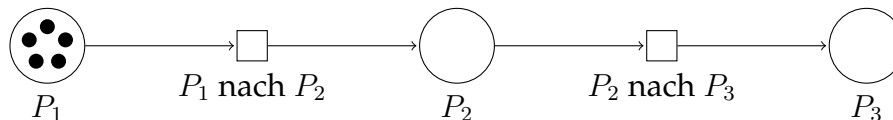


Abbildung 2.9: Ein einfaches Petrinetz mit einer Platzinvariante.

Platzinvarianten sind jedoch noch etwas allgemeiner anwendbar. In Abbildung 2.10 ist ein einfaches Systemnetz zum Verpacken von Produkten dargestellt. Auch hier werden wieder Materialflüsse dargestellt. Dabei werden Produkte mit Verpackungen zu verpackten Produkten kombiniert. Auch wenn so kein Material erzeugt oder zerstört wird, bleibt die Anzahl der Marken hier nicht gleich. Trotzdem lässt sich das Petrinetz mit einer Platzinvariante beschreiben.

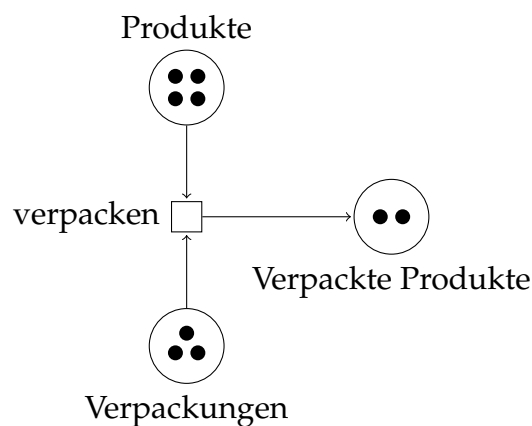


Abbildung 2.10: Auch für Petrinetze wie dieses kann man Invarianten finden.

Eine Marke in *VerpackteProdukte* ist nämlich genauso viel wert, wie eine Marke in *Produkte* zusammen mit einer Marke aus *Verpackungen*. Um auszudrücken, dass eine Marke aus *VerpackteProdukte* zwei Marken aus den anderen Plätzen entspricht, kann man die Anzahl der Marken in *VerpackteProdukte* mit zwei multiplizieren. Dann gilt immer die Gleichung:

$$2 \cdot |VerpackteProdukte| + |Produkte| + |Verpackungen| = 11$$

## 2.5.2 Formale Definition

Gegeben sei ein Elementares Petrinetz mit  $k$  Plätzen. Wenn man die Plätze als  $P_1, P_2, P_3, \dots, P_k$  bezeichnet und jedem Platz  $P_t$  eine Zahl  $n_t$  zuordnet, dann hat die Gleichung einer Platzinvariante die Form:

$$\sum_{s=1}^k n_s \cdot |P_s| = n_1 \cdot |P_1| + n_2 \cdot |P_2| + \dots + n_k \cdot |P_k| = c$$

Dabei wird  $c$  als die Konstante der Platzinvariante bezeichnet.

Die Zahlen  $n_1, n_2, \dots, n_t$  werden genau dann als Platzinvariante bezeichnet, wenn sich  $c$  beim Eintreten einer beliebigen Transition in einer beliebigen Markierung nicht ändert.

## 2.5.3 Platzinvarianten bestimmen

Die Platzinvarianten aus den beiden vorherigen Beispielen (Abbildung 2.9 und 2.10) kann man mit etwas Übung schon beim Betrachten des Petrinetzes erkennen. Bei größeren Petrinetzen wird das jedoch schwieriger und fehleranfälliger. Deshalb soll im folgenden Abschnitt ein Verfahren zum Bestimmen von Platzinvarianten in beliebig komplexen Petrinetzen vorgestellt werden. Als Beispiel wird dabei das Petrinetz aus Abbildung 2.11 dienen.

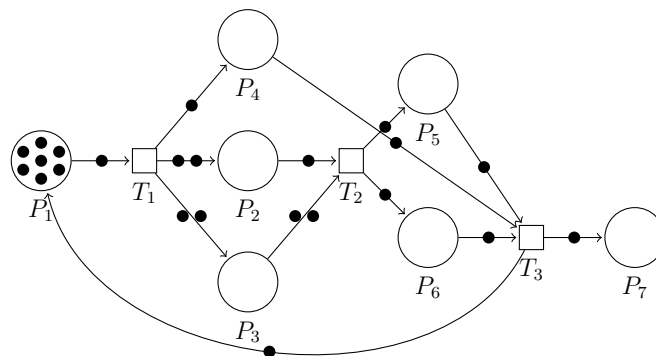


Abbildung 2.11: Ein Beispielnetz zum Bestimmen von Platzinvarianten

Um die Notation zu vereinfachen, definieren wir uns zunächst die Hilfsfunktion  $d(t, p)$ . Diese soll die Veränderung beschreiben, die eine Transition an einem Platz durchführt, wenn sie eintritt. Wenn  $t$  eine Transition ist und  $p$  ein Platz, dann ist

$$d(t, p) := (\text{Anzahl der Marken, die } t \text{ in } p \text{ erzeugt}) - (\text{Anzahl der Marken, die } t \text{ aus } p \text{ entnimmt})$$

Im Beispielpetritnetz aus Abbildung 2.11 lauten die Funktionswerte von  $d$  wie folgt:

$$\begin{array}{lll}
 d(T_1, P_1) = -1 & d(T_2, P_1) = 0 & d(T_3, P_1) = 1 \\
 d(T_1, P_2) = 2 & d(T_2, P_2) = -1 & d(T_3, P_2) = 0 \\
 d(T_1, P_3) = 2 & d(T_2, P_3) = -2 & d(T_3, P_3) = 0 \\
 d(T_1, P_4) = 1 & d(T_2, P_4) = 0 & d(T_3, P_4) = -1 \\
 d(T_1, P_5) = 0 & d(T_2, P_5) = 1 & d(T_3, P_5) = -1 \\
 d(T_1, P_6) = 0 & d(T_2, P_6) = 1 & d(T_3, P_6) = -1 \\
 d(T_1, P_7) = 0 & d(T_2, P_7) = 0 & d(T_3, P_7) = 1
 \end{array}$$

Um eine Platzinvariante zu finden, müssen konkrete Werte für die Variablen  $n_1, n_2, \dots, n_t$  bestimmt werden. Diese müssen der Bedingung aus Abschnitt 2.5.2 entsprechen: Beim Eintreten einer beliebigen Transition darf sich die Konstante  $c$  der Schleifeninvariante nicht verändern. Die Veränderung von  $c$  beim Eintreten einer Transition  $t$  berechnet sich durch:

$$\Delta c = \sum_{s=1}^k n_s \cdot d(t, P_s) = n_1 \cdot d(t, P_1) + n_2 \cdot d(t, P_2) + \dots + n_k \cdot d(t, P_k)$$

Indem man nun für jede Transition die Veränderung der Konstante mit 0 gleichsetzt, erzwingt man, dass die Bedingung für die Platzinvariante eingehalten wird. Für jede Transition  $t$  muss also gelten:

$$\sum_{s=1}^k n_s \cdot d(t, P_s) = n_1 \cdot d(t, P_1) + n_2 \cdot d(t, P_2) + \dots + n_k \cdot d(t, P_k) = 0$$

So entsteht ein lineares Gleichungssystem, das zum Beispiel mit dem Gaußschen Eliminationsverfahren gelöst werden kann. Beim Petritnetz aus Abbildung 2.11 ergeben sich die folgenden Gleichungen:

$$\left| \begin{array}{l}
 T_1 : n_1 \cdot (-1) + n_2 \cdot 2 + n_3 \cdot 2 + n_4 \cdot 1 + n_5 \cdot 0 + n_6 \cdot 0 + n_7 \cdot 0 = 0 \\
 T_2 : n_1 \cdot 0 + n_2 \cdot (-1) + n_3 \cdot (-2) + n_4 \cdot 0 + n_5 \cdot 1 + n_6 \cdot 1 + n_7 \cdot 0 = 0 \\
 T_3 : n_1 \cdot 1 + n_2 \cdot 0 + n_3 \cdot 0 + n_4 \cdot (-1) + n_5 \cdot (-1) + n_6 \cdot (-1) + n_7 \cdot 1 = 0
 \end{array} \right|$$

Durch Lösen des Gleichungssystems kommt man auf die Werte für  $n_1, n_2, \dots, n_7$ . Jede Lösung ist eine Platzinvariante des Petritnetzes. Das Gleichungssystem hat unendlich



viele Lösungen, entsprechend gibt es auch unendlich viele Platzinvarianten. Einige exemplarische Lösungen lauten:

$$\text{Platzinvariante 1 : } n_1 = 2 \quad n_2 = 0 \quad n_3 = 1 \quad n_4 = 0 \quad n_5 = 2 \quad n_6 = 0 \quad n_7 = 0$$

$$\text{Platzinvariante 2 : } n_1 = 1 \quad n_2 = 0 \quad n_3 = 0 \quad n_4 = 1 \quad n_5 = 0 \quad n_6 = 0 \quad n_7 = 0$$

$$\text{Platzinvariante 3 : } n_1 = 2 \quad n_2 = 0 \quad n_3 = 1 \quad n_4 = 0 \quad n_5 = 0 \quad n_6 = 2 \quad n_7 = 0$$

$$\text{Platzinvariante 4 : } n_1 = -2 \quad n_2 = -2 \quad n_3 = 1 \quad n_4 = 0 \quad n_5 = 0 \quad n_6 = 0 \quad n_7 = 2$$

Um die Gleichung der Platzinvarianten zu bestimmen, fehlen noch die Konstanten der Platzinvarianten. Diese können aus der Anfangsmarkierung nach der Formel aus Abschnitt 2.5.2 berechnet werden. Für die oben bestimmten Platzinvarianten ergibt sich so:

$$\text{Platzinvariante 1 : } c = 14$$

$$\text{Platzinvariante 2 : } c = 7$$

$$\text{Platzinvariante 3 : } c = 14$$

$$\text{Platzinvariante 4 : } c = -14$$

Anschließend können die Gleichungen der Platzinvarianten aufgeschrieben werden. Dafür müssen die Werte für  $n_1, n_2, \dots, n_k$  sowie für  $c$  in die Formel aus Abschnitt 2.5.2 eingesetzt werden. So erhält man für das Systemnetz aus Abbildung 2.11 die folgenden Gleichungen:

$$\text{Platzinvariante 1 : } 2 \cdot |P_1| + |P_3| + 2 \cdot |P_5| = 14$$

$$\text{Platzinvariante 2 : } |P_1| + |P_4| = 7$$

$$\text{Platzinvariante 3 : } 2 \cdot |P_1| + |P_3| + 2 \cdot |P_6| = 14$$

$$\text{Platzinvariante 4 : } (-2) \cdot |P_1| + (-2) \cdot |P_2| + |P_3| + 2 \cdot |P_7| = -14$$

Diese Gleichungen gelten in allen erreichbaren Markierungen.

Das gezeigte Verfahren ist universell einsetzbar und ermöglicht es, alle Platzinvarianten eines gegebenen Systemnetzes zu finden. Da es in der Hauptsache auf dem Lösen eines linearen Gleichungssystems beruht, sollte es außerdem möglich sein, das Verfahren zu automatisieren. Wie man anhand der Gleichung der Platzinvariante 4 sieht, gehen die Ergebnisse der Analyse über die in Abschnitt 2.5.1 vorgestellte Analogie zu Material- oder Informationsflüssen hinaus, denn durch die negativen Koeffizienten lässt sich

diese hier nicht anwenden. Welche Möglichkeiten die Gleichungen zum weiteren Vorgehen bieten, wird in Kapitel 3 - [Fallstudie](#) besprochen.

## 3 Fallstudie

Im Folgenden soll eine Produktionsanlage eines Baukastens für einen Anemometer <sup>1</sup> modelliert werden, welches im Rahmen eines Projektes <sup>2</sup> der Gruppe Woodchucks an der DHBW Mannheim entwickelt wurde.

### 3.1 Modellierung

Der Kreislauf beginnt mit der Eingabe eines neuen Auftrages. Dieser ermöglicht die Ausführung der Transition („Auftrag bearbeiten“). Somit wird die Ausführung der neuen Transition „Grundplatte bauen“ ermöglicht.

Diese beginnt nun mit dem Bau der Grundplatte. Hieraus werden zwei neue Transitionen angesteuert, sobald eine Einheit produziert worden ist.

Bevor die Transition „Flügel herstellen“ ausgeführt werden kann, wird der CNC-Modus der Fräse geändert. Nun wird begonnen die Flügel herzustellen. Sobald vier Einheiten produziert worden sind, wird der CNC-Modus umgestellt, um mit der Fräse eine neue Grundplatte erzeugen zu können.

Sobald nun die Flügel hergestellt sind und eine Menge von mindestens 40 Einheiten Holz verbraucht worden sind, so wird eine neue Transition „Holz kaufen“ ausgeführt, welche nach deren Ablauf das Lager mit dem neuem Holz nachfüllt. Nach der Produktion von 4 Einheiten der Flügel beginnt nun die Transition „Zusammensetzen“. Bei dieser Aktion werden nun die einzelnen Komponenten:

- Standbeine
- Flügel
- Grundplatte

zusammengesetzt und für die weitere Verarbeitung, als neue Einheit zur Verfügung gestellt.

---

<sup>1</sup> Ein Gerät zum messen der Windstärke

<sup>2</sup> <http://mett.ddns.net> [Stand: Juni]

Sollten nun bei dieser Transition mindestens 8 Standbeine verbraucht worden sein, so wird eine neue für den Einkauf und Einlagerung von Standbeinen in Kraft gesetzt.

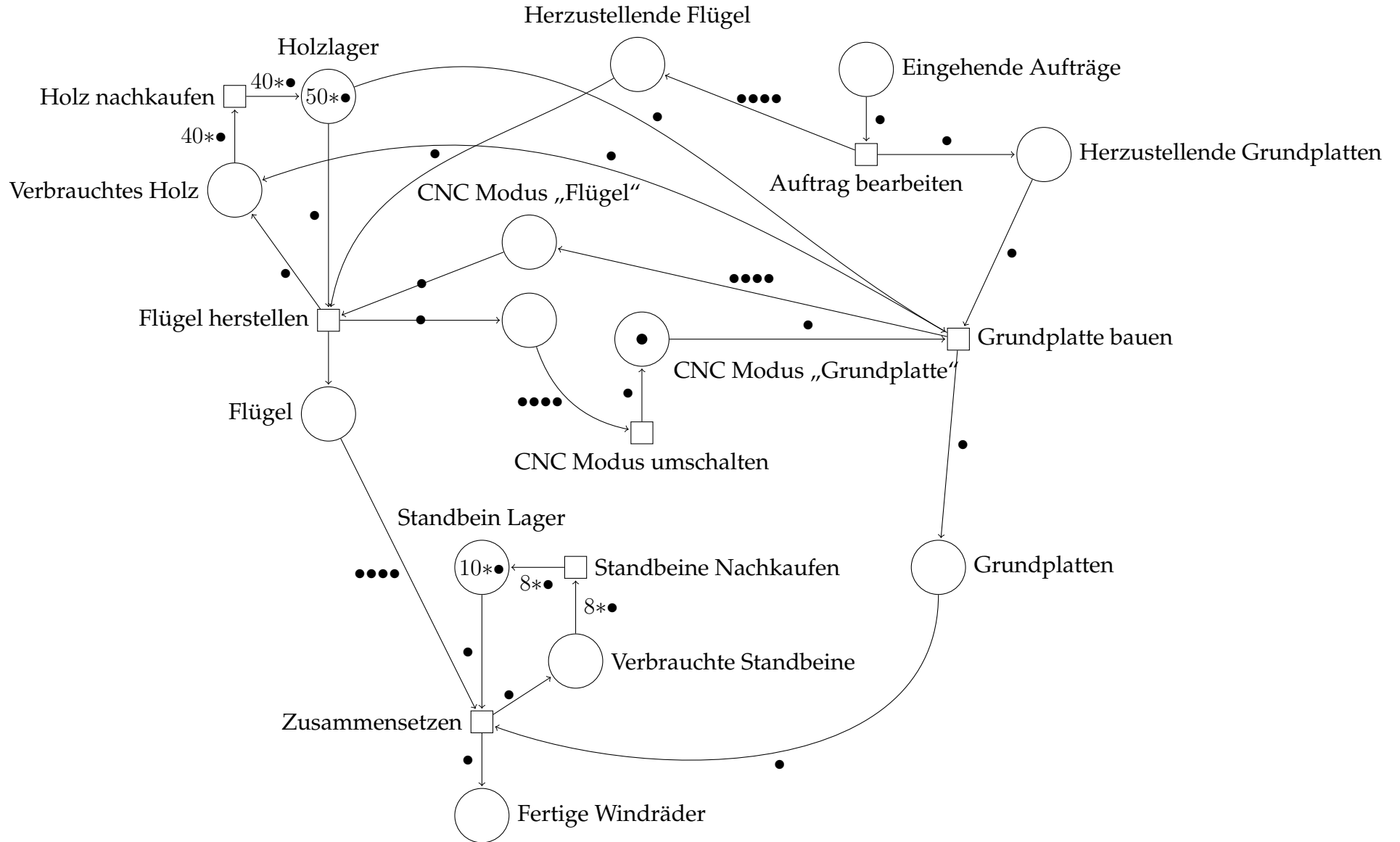


Abbildung 3.1: Modell der Anemometer Produktion

## 3.2 Analyse und Verifikation

Bei diesem Modell ist es wichtig, dass genau so viele Windräder hergestellt werden, wie eingehende Aufträge vorhanden sind. Dass dies tatsächlich der Fall ist, soll im Folgenden mit Hilfe der Platzinvarianten verifiziert werden.

Um die Schreibweise kurz zu halten, werden die Komponenten des Petrinetzes entsprechend den Tabelln 3.1 und 3.2 auf Seite 19 umbenannt.

Es wird weiterhin davon ausgegangen, dass zu Beginn  $a$  Aufträge vorliegen. Im Platz  $P_9$  befinden sich also  $a$  Marken.

Zunächst müssen die Platzinvarianten des Petrinetzes bestimmt werden. Dafür kann das in Abschnitt 2.5.3 vorgestellte Verfahren verwendet werden. Darauf soll hier nicht erneut eingegangen werden. Die Gleichungen der erhaltenen Platzinvarianten lauten:

$$|P_{13}| - \frac{1}{4} \cdot |P_5| - \frac{1}{4} \cdot |P_3| = -\frac{1}{4} \quad (3.1)$$

$$|P_{12}| + |P_{10}| = 10 \quad (3.2)$$

$$|P_{11}| + |P_9| + \frac{1}{4} \cdot |P_6| + \frac{1}{4} \cdot |P_3| = a \quad (3.3)$$

$$|P_8| - \frac{1}{4} \cdot |P_6| + \frac{1}{4} \cdot |P_5| = 0 \quad (3.4)$$

$$|P_7| + \frac{1}{4} \cdot |P_5| + \frac{1}{4} \cdot |P_4| = 1 \quad (3.5)$$

$$|P_1| + |P_2| = 50 \quad (3.6)$$

Nach dem vollständigen Bearbeiten aller Aufträge befinden sich die hergestellten Windräder in Platz  $P_{11}$ . Die Plätze für die eingehenden Aufträge ( $P_9$ ) sowie für die Teilaufträge zum Herstellen der Einzelteile ( $P_6$  und  $P_8$ ) und die Zwischenlager ( $P_3$  und  $P_{13}$ ) sind dann leer. Es gilt also:

$$|P_{13}| = 0$$

$$|P_9| = 0$$

$$|P_8| = 0$$

$$|P_6| = 0$$

$$|P_3| = 0$$

<b>Verkürzte Bezeichnung</b>	<b>Transition</b>
$T_1$	Auftrag bearbeiten
$T_2$	Flügel herstellen
$T_3$	Grundplatte bauen
$T_4$	Zusammensetzen
$T_5$	CNC-Modus umschalten
$T_6$	Standbeine nachkaufen
$T_7$	Holz nachkaufen

Tabelle 3.1: Umbenennung der Transitionen

<b>Verkürzte Bezeichnung</b>	<b>Platz</b>
$P_1$	Holzlager
$P_2$	Verbrauchtes Holz
$P_3$	Flügel
$P_4$	(ohne Beschriftung)
$P_5$	CNC-Modus Flügel
$P_6$	Herzustellende Flügel
$P_7$	CNC-Modus Flügel
$P_8$	Herzustellende Grundplatten
$P_9$	Eingehende Aufträge
$P_{10}$	Standbein Lager
$P_{11}$	Fertige Windräder
$P_{12}$	Verbrauchte Standbeine
$P_{13}$	Grundplatten

Tabelle 3.2: Umbenennung der Plätze

Um nachzuweisen, dass pro Auftrag ein Windrad hergestellt wird, kann Gleichung 3.3 verwendet werden. Wenn man dort für  $|P_9|$ ,  $|P_6|$  und  $|P_3|$  die soeben festgelegten Werte einsetzt, erhält man die Gleichung:

$$|P_{11}| = a$$

In  $P_{11}$  befinden sich die fertigen Windräder;  $a$  ist die Anzahl der Aufträge in der Anfangsmarkierung. Die Gleichung zeigt also, dass pro Auftrag tatsächlich genau ein Windrad hergestellt wird.



## 4 Fazit

Petrinetze erlauben es, Systeme formal zu modellieren. Die Stärken von Petrinetzen liegen jedoch vor allem bei der Analyse und Verifikation der Modelle. Die Analyse auf Platzinvarianten liefert Aufschlüsse über die Funktionsweise eines Petrinetzes. Diese kann man zum Beispiel dafür verwenden, das Modell zu verifizieren. Die Schwierigkeit der Petrinetze liegt bei der Modellierung. Komplexe Systeme werden schnell unübersichtlich und können von Menschen kaum noch modelliert werden. Abhilfe schaffen hierfür die high-level Petrinetze, die in low-level Petrinetze umgewandelt werden können. Außerdem lassen sich viele konkretere Modellierungssprachen direkt auf Petrinetze zurückführen. Dazu zählen zum Beispiel BPMN 2.0 und EPK<sup>1</sup>, zwei Sprachen zur Modellierung von Geschäftsprozessen. Entsprechende Software kann die modellierten Geschäftsprozesse dann intern in Petrinetze umwandeln. So kann dem Benutzer eine einfache Möglichkeit zur Modellierung geboten werden. Gleichzeitig können mächtige Funktionen zur Analyse, Verifikation und Simulation, die auf den theoretischen Erkenntnissen über Petrinetze aufbauen, implementiert werden. Zusammenfassend lässt sich sagen, dass Petrinetze ein mächtiges Werkzeug sind. Im Geschäftsprozessmanagement wird ein Anwender aufgrund ihrer Komplexität, jedoch eher selten direkt mit ihnen in Berührung kommen.

---

<sup>1</sup> [Ber13]

# Abbildungsverzeichnis

2.1	Beispiel für einen leeren Platz und einen Platz mit einer (abstrakten) Marke. . . . .	3
2.2	Beispiel für eine Transition . . . . .	4
2.3	Beispiele für Kanten . . . . .	4
2.4	Markierung 1 eines Petrinetzes . . . . .	5
2.5	Markierung 2 eines Petrinetzes . . . . .	5
2.6	Ein Petrinetz mit Variablen in den Kantenbeschriftungen . . . . .	6
2.7	Das Petrinetz aus Abbildung 2.6 nach dem Eintreten der Transition. . . . .	7
2.8	Ein Zähler . . . . .	7
2.9	Ein einfaches Petrinetz mit einer Platzinvariante. . . . .	10
2.10	Auch für Petrinetze wie dieses kann man Invarianten finden. . . . .	10
2.11	Ein Beispielnetz zum Bestimmen von Platzinvarianten . . . . .	11
3.1	Modell der Anemometer Produktion . . . . .	17

# Tabellenverzeichnis

3.1	Umbenennung der Transitionen . . . . .	19
3.2	Umbenennung der Plätze . . . . .	19

# Literatur

- [15a] *Deadlocking*. Juni 2015. URL: [https://technet.microsoft.com/en-us/library/ms177433\(v=sql.105\).aspx](https://technet.microsoft.com/en-us/library/ms177433(v=sql.105).aspx).
- [15b] *Obituary - Carl Adam Petri (1926-2010)*. Juni 2015. URL: <http://www.informatik.uni-hamburg.de/TGI/mitarbeiter/profs/petri/obituary.html>.
- [Ber13] Robin Bergenthum. *Verifikation von Halbgeordneten Abläufen in Petrinetzen*. Feb. 2013.
- [Fun06] Jan Funke. *Einführung in die Theorie gefärbter Petrinetze*. 2006.
- [Her15] Manuel Hertlein. *Petri-Netze / Eine Einführung*. 2015. URL: [https://www2.informatik.hu-berlin.de/top/lehre/WS05-06/se\\_systementwurf/Petrinetze-1.pdf](https://www2.informatik.hu-berlin.de/top/lehre/WS05-06/se_systementwurf/Petrinetze-1.pdf).
- [Rei10] Wolfgang Reisig. *Petrinetze Modellierungstechnik, Analysemethoden, Fallstudien*. 2010.
- [Wol] Jörg Diesel Wolfgang Reisig. *Konzepte der Petrinezete*.