Process Mining

Klausur: 22.7/27.7

Inhalt

[Motivation und Überblick 3](#_Toc8492055)

[Tools 3](#_Toc8492056)

[Process Mining Fragestellungen 3](#_Toc8492057)

[Mathematische Grundlagen 3](#_Toc8492058)

[Data Mining 4](#_Toc8492059)

[Petrinetze 5](#_Toc8492060)

[Andere Systemmodelle 6](#_Toc8492061)

[Eingabeseite/Ereignis-Log 7](#_Toc8492062)

[Alpha-Algorithmus (Filtert Noise nicht) 8](#_Toc8492063)

[Heuristik Miner (filtert Noise) 9](#_Toc8492064)

[Synthese-basierte Verfahren 9](#_Toc8492065)

[Conformance Checking – Qualitätsüberprüfung 9](#_Toc8492066)

[Enhancement – Daten + Modell = besseres Modell 9](#_Toc8492067)

[Erweiterte Überblicksgrafik und Ausblick 9](#_Toc8492068)

# Motivation und Überblick

## Tools

PROM-Toolbox (PromLite)

Ceronis (keine Nebenläufigkeit)

Disco

## Process Mining Fragestellungen

Welchem Arbeitsablauf folgen die Teilnehmer (eines Prozesses) wirklich?

* Abweichungen zur erwarteten Praxis
* Gründe für Abweichungen
* Sind Probleme zu erwarten?
* Was sind mögliche Gegenmaßnahmen

Gibt es Engpässe im Prozess?

* Wie kann ich diese vermeiden?

Was sind die meistbenutzten Pfade?

Können Probleme vorausgesagt werden?

* Abweichungen, Kostenrisiken, Terminrisiken, Qualitätsrisiken

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Pat.Nr | Untersuchung | Zeit | Arzt | Alter | Diagnose | Kosten | Name |
| 1 | Bluttest | 31.1 |  |  |  |  |  |
| 2 | Röntgen | 31.1 |  |  |  |  |  |
| .. | .. |  |  |  |  |  |  |
| 1 | x | 2.2 |  |  |  |  |  |

Jede Zeile wird als Aktivität / Ereignis in Arbeitsablauf betrachtet. Beispielsweise könnte man Untersuchungen analysieren: Zu jedem Patienten gehört ein Behandlungsverlauf. Die Menge der Behandlungsverläufe können wir durch ein Prozessmodell darstellen und analysieren.

# Mathematische Grundlagen





## Formale Sprachen

Alphabet = Endliche Menge T von Zeichen

Wort = Endliche Folge w = von Zeichen aus T

Leeres Wort = 

Länge eines Wortes w ist die Anzahl seiner Zeichen |w|

### Beispiel:

T = {a, b} u = ab a als Aktivität

 = 0 v = bbb

| a b a | = 3 uv = abbbb

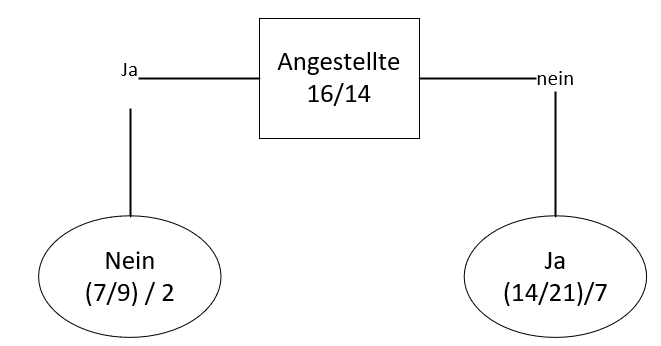
 = Menge aller Wörter über T mit Länge >= 1

* = 
* = Menge aller Wörter über T der Länge n für ein 

# Data Mining

Übungsaufgabe zu Entscheidungsbäumen

|  |  |
| --- | --- |
| Angestellte Ja | Angestellte Nein |
| C(ja) = 2  C(nein) = 7 | C(ja) = 14  C(nein) = 7 |



= 0,76

🡪 <0.37 letzte Vorlesung

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Vorhergesagte Klasse | |
| Ja | Nein |
| Tatsächliche Klasse | Ja | 14 | 2 |
| Nein | 7 | 7 |

Fehlerrate =

Genauigkeit =

Trefferquote =

# Petrinetze

Ein einfaches Petrinetz

besteht aus

* Einer endlichen Menge von Stellen P
* Einer endlichen Menge von Transitionen T
* Die Flussrelation

Interpretation:

* Transitionen repräsentieren Aktionen
* Stellen repräsentieren lokale Zustände

°

Eine Markierung ist eine Abbildung . Diese repräsentiert den Zustand

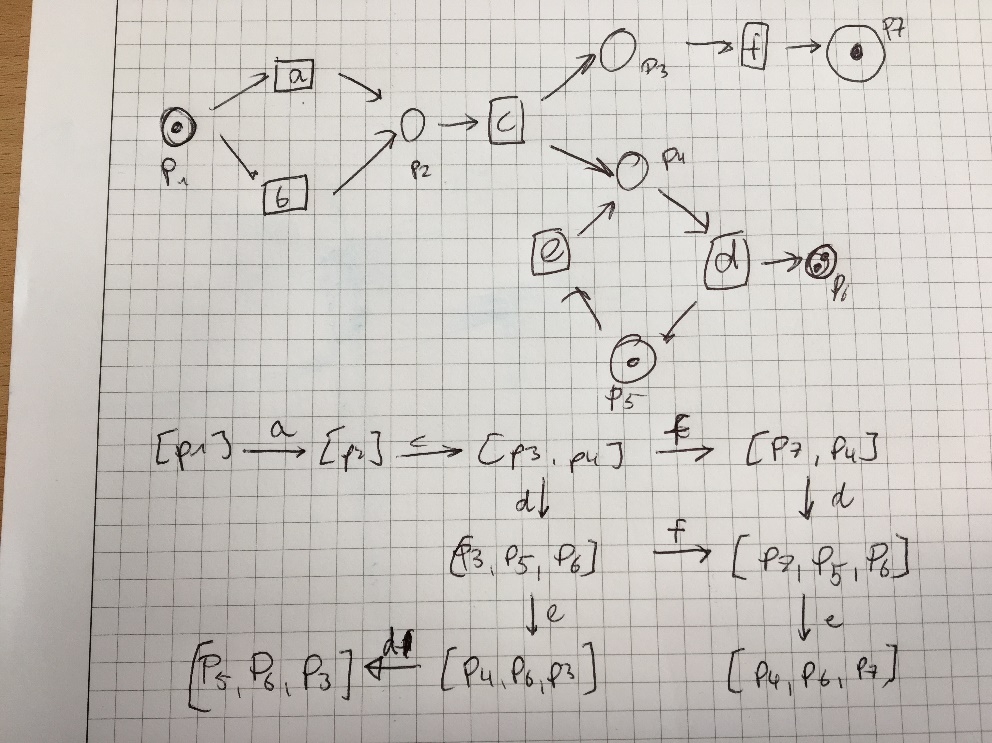
Ein markiertes Netz besteht aus einem einfachen Netz und einer Anfangsmarkierung m0.

Jetzt definieren wir das Verhalten von markierten Petrinetzen. In welchem Zustand ist welche Aktion möglich und in welchen Folgezustand führt eine mögliche Aktion.

Schaltregel

Eine Transition t ist aktiviert in einer Markierung m, falls m >= °t

In allen Vorbereichstellen von t liegt eine Marke.

Eine aktivierte Transition t kann schalten ( = Ausführung einer Aktion). Das Schalten von t führt in den Folgezustand . Dafür schreiben wir 

Eine Schaltfolge ist ein Wort mit . Dafür schreiben wir .

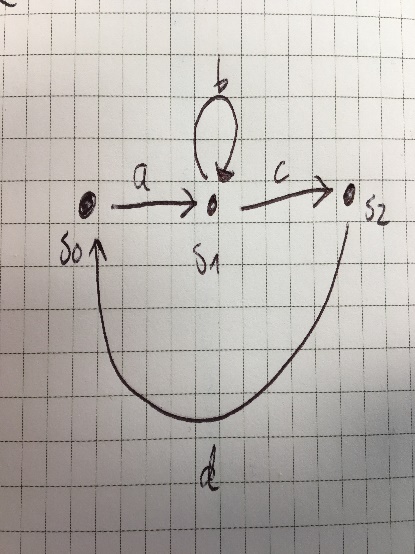
Die Sprache von N=(P, T, F, m0) ist definiert als

Die Menge der erreichbaren Markierungen ist

## Exkurs Transitionssysteme

Ein Transitionssystem besteht aus

* S Menge der (globalen) Zustände
* Anfangszustand
* T endliche Menge von Transitionen
* Zustandsübergangsrelation



Für

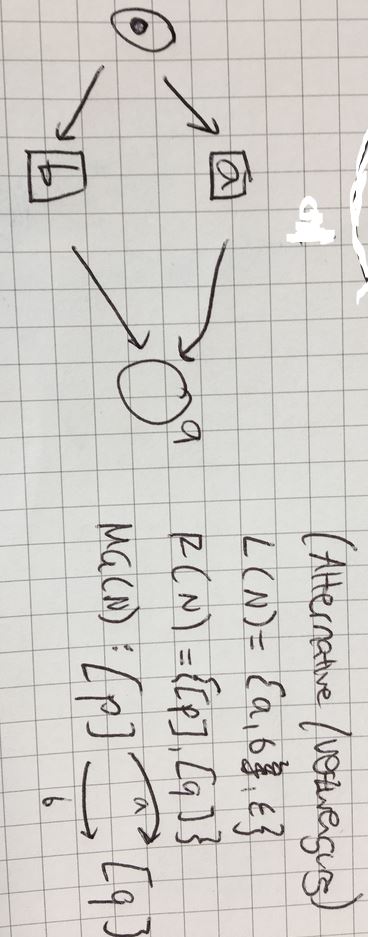
Interpratation:

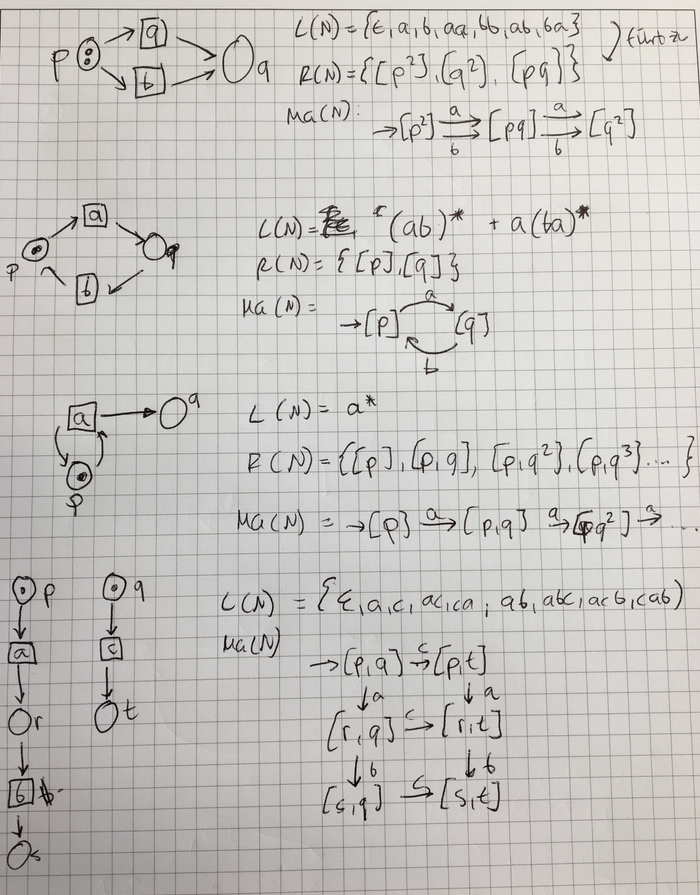
T kann in s schalten und führt in den Folgezustand s‘

Der Markierungsgrad eines markierten Netzes N ist das Transitionssystem

)

Beispiele:





Ein Schritt ist eine Multimenge über T. Er kann schalten in m, falls

Wenn schaltet, so führt das zu der Folgemarkierung

Dafür schreiben wir

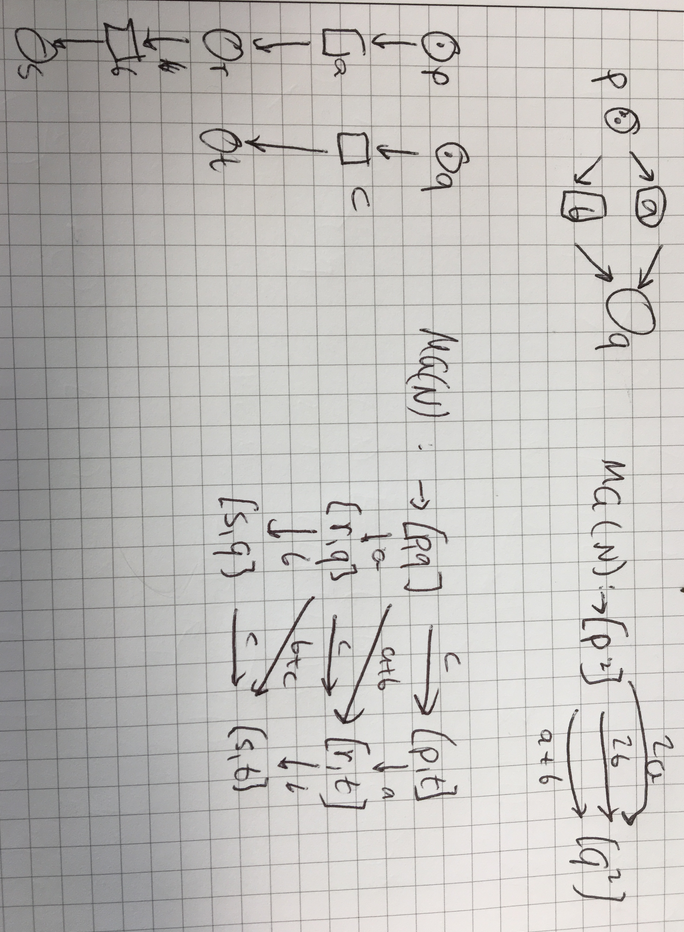
Die Transitionen innerhalb eines Schritts sind nebenläufig mit

Ein Wort heißt Schrittschaltfolge, falls Markierungen m1,.., mn existieren mit

Wir schreiben auch . Die Menge alle Schrittschaltfolgen heißt Schrittsprache Ls(N) von N.

Der Schrittmarkierungsgraph ist wie folgt definiert:

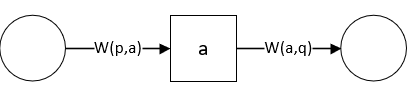
Im Schrittmarkierungsgraphen kann man Nebenläufigkeit ablesen im Gegensatz zum Markierungsgraphen

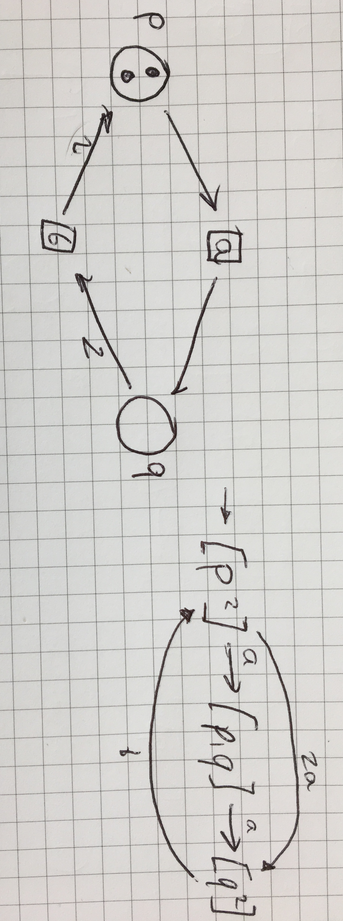


## Stellen / Transitionsnetze (S/T-Netze)

Ein S/T-Netz N=(P, T, F, W) besteht aus

* Einem einfachen Netz (P, T, F)
* Einer Kantengewichtsfunktion W





Wir erweitern die Definitionen °t und t° wie folgt:

Folgende Definition können wir unverändert von einfachen Netzen übernehmen:

* Schaltregel
* Schaltfolge
* Sprache
* Markierungsgraph
* Und dasselbe für Schritte
* Erreichbare Markierungen

# re Systemmodelle

# Eingabeseite/Ereignis-Log

# Alpha-Algorithmus (Filtert Noise nicht)

# Heuristik Miner (filtert Noise)

# Synthese-basierte Verfahren

# Conformance Checking – Qualitätsüberprüfung

# Enhancement – Daten + Modell = besseres Modell

# Erweiterte Überblicksgrafik und Ausblick