



*Vorlesung*  
***Methodische Grundlagen des  
Software-Engineering***  
im Sommersemester 2014

Prof. Dr. Jan Jürjens

TU Dortmund, Fakultät Informatik, Lehrstuhl XIV

Teil 2.4: Prozessextraktion

v. 02.06.2014

1



## 2.4 Prozessextraktion

[mit freundlicher Genehmigung basierend  
auf einem englischen Foliensatz von  
Prof. Dr. Wil van der Aalst (TU Eindhoven)]  
[inkl. Beiträge von Jutta Mülle und Dr. Silvia von Stackelberg,  
LS Prof. Böhm, Karlsruher Institut für Technologie]

### Literatur:

[vdA11] Wil van der Aalst: **Process Mining: Discovery, Conformance and Enhancement of Business Processes**, Springer-Verlag. 2011.

Unibibliothek (6 Exemplare): <http://www.ub.tu-dortmund.de/katalog/titel/1332248>  
(Bei Engpässen kann eine **Kopiervorlage** der relevanten Ausschnitte zur  
Verfügung gestellt werden.)

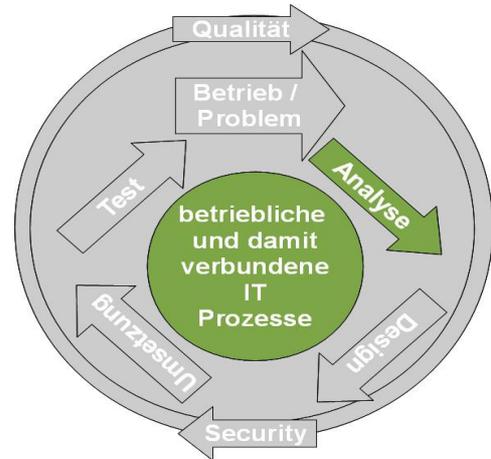
- **Kapitel 5**

- Geschäftsprozessmodellierung

- **Process-Mining**

- Einführung: Process-Mining
- Petrinetze
- Data-Mining
- Datenbeschaffung
- **Prozessextraktion**
- Konformanzanalyse
- Mining: Zusätzliche Perspektiven
- Betriebsunterstützung
- Werkzeugunterstützung
- Analysiere „Lasagne Prozesse“
- Analysiere „Spaghetti Prozesse“
- Kartographie und Navigation
- Epilog

- Modellbasierte Entwicklung sicherer Software

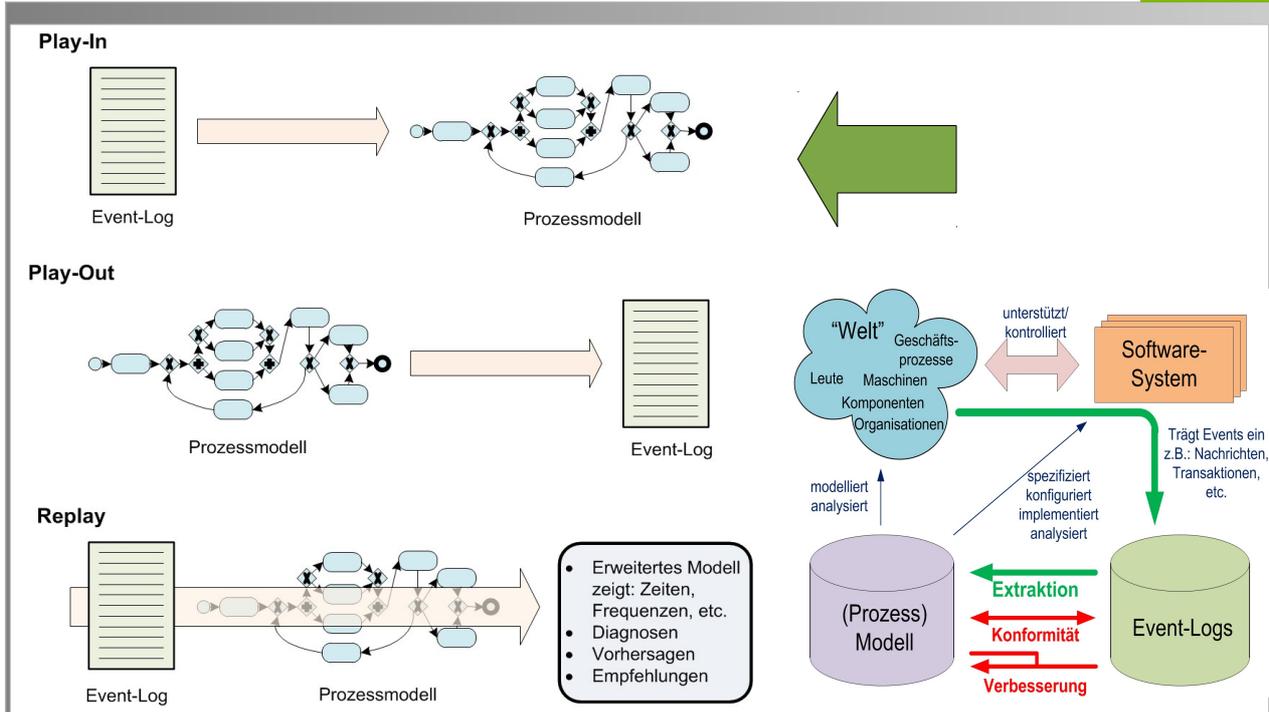




- **Vorheriger Abschnitt:** Datenbeschaffung und Datenspeicherformat XES.
- **Dieser Abschnitt:** „Prozessextraktion“:
  - Einführung und Beispiele
  - **$\alpha$ -Algorithmus:** Vorstellung und Beispiele
  - Prozessmodell: Ein schweres Problem?



- **Einführung und Beispiel**
- $\alpha$ -Algorithmus
  - Idee und Vorbereitungen
  - Formalisierung
  - Beispiel
  - Weitere Beispiele
  - Einschränkungen
- Allgemeine Herausforderungen beim Process-Mining



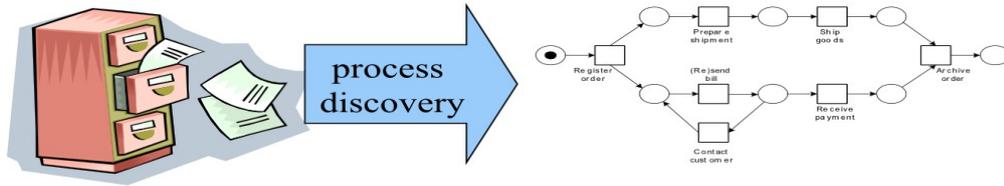
6

## Literatur:

Wil van der Aalst: Process Mining: Discovery, Conformance and Enhancement of Business Processes

- Kap. 1.5: S.19, Fig. 1.8

Idee: Umdrehen des traditionellen Ansatzes



Ausgangspunkt: **Logfile** (= Folge von Log-Daten)

- **Minimal**: Nummer der GP-Instanz („case“ / Fall), GP-Aktivität, zeitliche Ordnung
- **Optional**: Genaue Zeit, Nutzer, assoziierte Daten etc.
- „**Rauschfrei**“: GP-Instanznr. erlaubt irrelevante Daten wegzufiltern.
- **Vollständig**: Relevante Daten für verschiedene GP-Instanzen vorhanden.

Ergebnis: **Workflow-Netz**.

7



Abgebildete Logdatei enthält folgende Folgen von Logdaten für verschiedene GP-Instanzen:

Log-Datei:

```
case 1, task A
case 2, task A
case 3, task A
case 3, task B
case 1, task B
case 1, task C
case 2, task C
case 4, task A
case 2, task B
case 2, task D
case 5, task A
case 4, task C
case 1, task D
case 3, task C
case 3, task D
case 4, task B
case 5, task E
case 5, task D
case 4, task D
```

8

## Literatur:

Wil van der Aalst: Process Mining: Discovery, Conformance and Enhancement of Business Processes

- Kap. 5.1: S. 126



Abgebildete Logdatei enthält folgende Folgen von Logdaten für verschiedene GP-Instanzen:

- ABCD case 1,

Log-Datei:

```
case 1, task A
case 2, task A
case 3, task A
case 2, task B
case 1, task B
case 1, task C
case 2, task C
case 4, task A
case 2, task B
case 2, task D
case 5, task A
case 4, task C
case 1, task D
case 3, task C
case 3, task D
case 4, task B
case 5, task E
case 5, task D
case 4, task D
```

9

## Literatur:

Wil van der Aalst: Process Mining: Discovery, Conformance and Enhancement of Business Processes

- Kap. 5.1: S. 126



Abgebildete Logdatei enthält folgende Folgen von Logdaten für verschiedene GP-Instanzen:

- ABCD (case 1, case 3)
- ACBD (case 2, case 4)
- AED (case 5)

Zugehöriges Petrinetz ?

Log-Datei:

```
case 1, task A
case 2, task A
case 3, task A
case 3, task B
case 1, task B
case 1, task C
case 2, task C
case 4, task A
case 2, task B
case 2, task D
case 5, task A
case 4, task C
case 1, task D
case 3, task C
case 3, task D
case 4, task B
case 5, task E
case 5, task D
case 4, task D
```

Abstrahiert von „cases“ (insbes. von mehreren gleichen Abläufen).

Jedes Ereignis nur einmal enthalten => nicht einfach o.g. Folgen durch OR-Verzweigungen auflisten.

10

## Literatur:

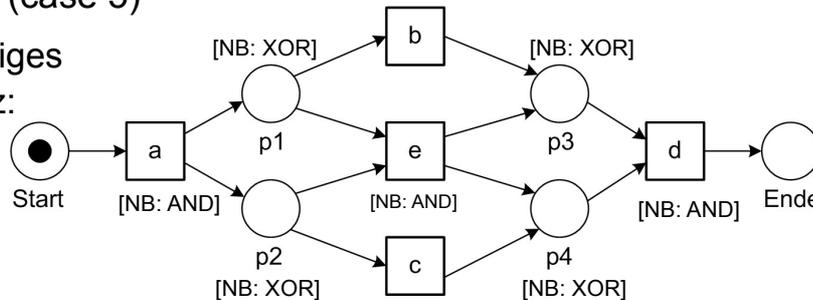
Wil van der Aalst: Process Mining: Discovery, Conformance and Enhancement of Business Processes

- Kap. 5.1: S. 126

Abgebildete Logdatei enthält folgende Folgen von Logdaten für verschiedene GP-Instanzen:

- ABCD (case 1, case 3)
- ACBD (case 2, case 4)
- AED (case 5)

Zugehöriges  
Petriernetz:



Abstrahiert von „cases“ (insbes. von mehreren gleichen Abläufen).  
Jedes Ereignis nur einmal enthalten => nicht einfach o.g. Folgen durch OR-Verzweigungen auflisten.  
Erzeugt keine weiteren Eventfolgen (nicht immer der Fall !)

Log-Datei:

```

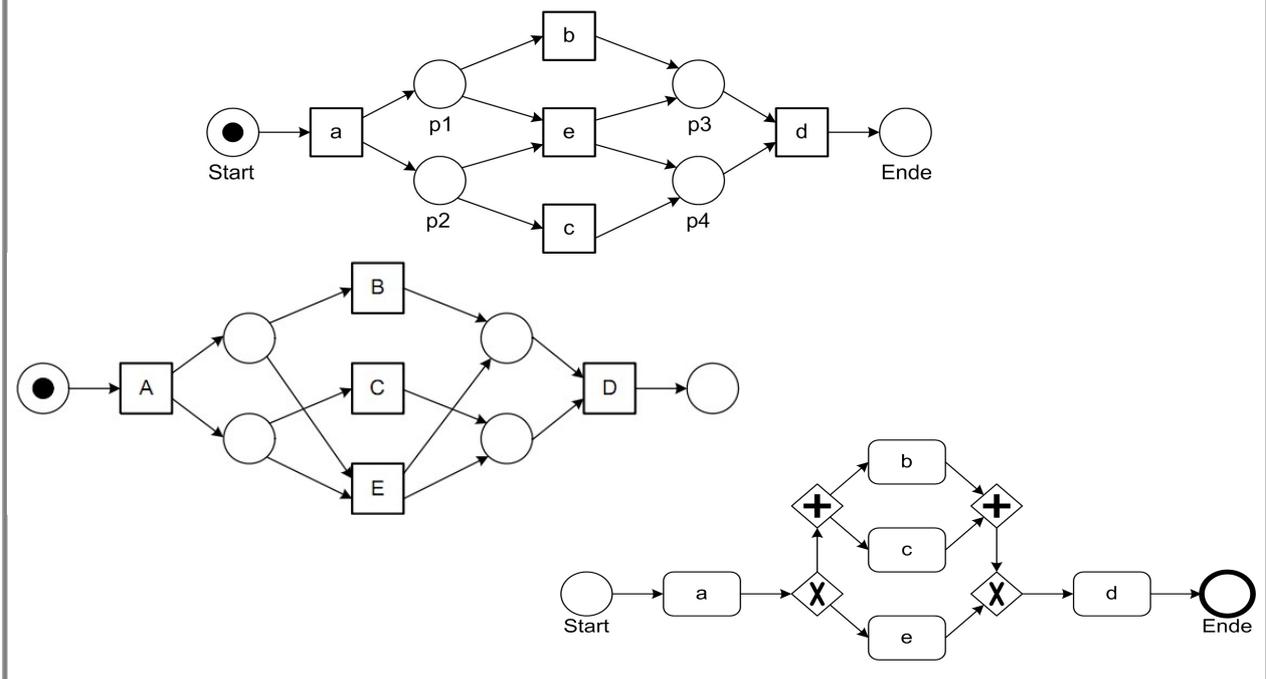
case 1, task A
case 2, task A
case 3, task A
case 3, task B
case 1, task B
case 1, task C
case 2, task C
case 4, task A
case 2, task B
case 2, task D
case 5, task A
case 4, task C
case 1, task D
case 3, task C
case 3, task D
case 4, task B
case 5, task E
case 5, task D
case 4, task D
    
```

11

## Literatur:

Wil van der Aalst: Process Mining: Discovery, Conformance and Enhancement of Business Processes

- Kap. 5.1: S. 126

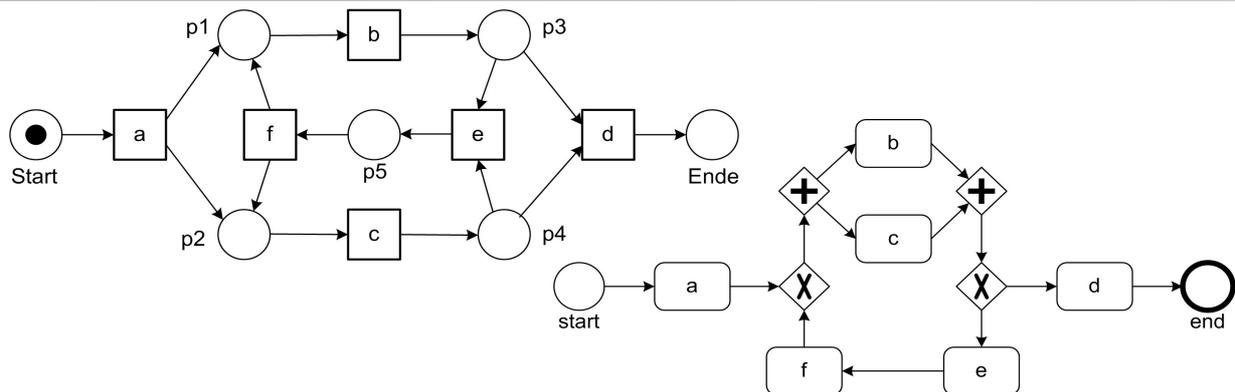


12

## Literatur:

Wil van der Aalst: Process Mining: Discovery, Conformance and Enhancement of Business Processes

- Kap. 5.1: S. 128, Fig. 5.3 (b)



$$L_2 = [\langle a, b, c, d \rangle^3, \langle a, c, b, d \rangle^4, \langle a, b, c, e, f, b, c, d \rangle^2, \langle a, b, c, e, f, c, b, d \rangle, \langle a, c, b, e, f, b, c, d \rangle^2, \langle a, c, b, e, f, b, c, e, f, c, b, d \rangle]$$

Superscript von **Event-Folge**: Häufigkeit des Auftretens.

Hier: Event-Log enthält **Teilmenge von möglichen Traces** des Modells

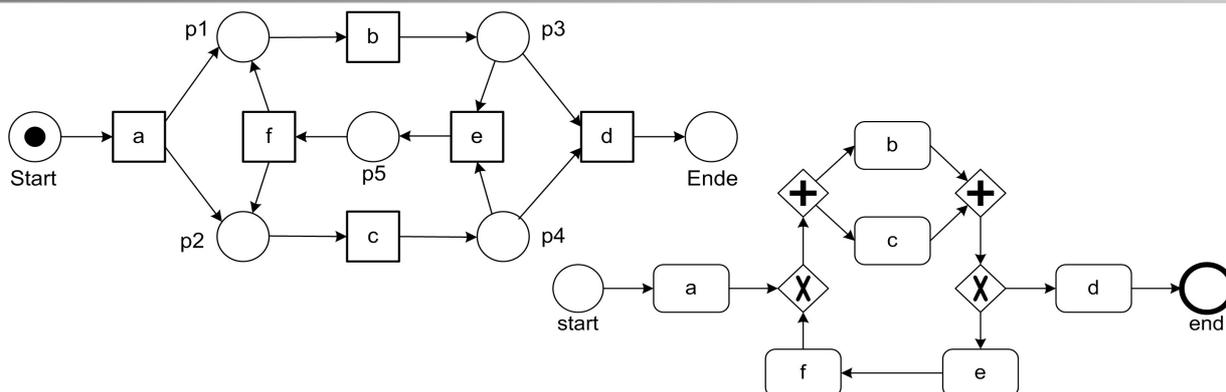
D.h. Modell **Verallgemeinerung der Log-Daten**, lässt weiteres Verhalten zu.  
(Welches z.B. ?)

13

## Literatur:

Wil van der Aalst: Process Mining: Discovery, Conformance and Enhancement of Business Processes

- Kap. 5.1: Seite 126-128, Fig. 5.2, 5.3 (b)



$$L_2 = [\langle a, b, c, d \rangle^3, \langle a, c, b, d \rangle^4, \langle a, b, c, e, f, b, c, d \rangle^2, \langle a, b, c, e, f, c, b, d \rangle, \langle a, c, b, e, f, b, c, d \rangle^2, \langle a, c, b, e, f, b, c, e, f, c, b, d \rangle]$$

Superscript von **Event-Folge**: Häufigkeit des Auftretens.

Hier: Event-Log enthält **Teilmenge von möglichen Traces** des Modells

D.h. Modell **Verallgemeinerung der Log-Daten**, lässt weiteres Verhalten zu.  
(Welches z.B. ?) => Beliebige Anzahl Iterationen der Schleife !

14

## Literatur:

Wil van der Aalst: Process Mining: Discovery, Conformance and Enhancement of Business Processes

- Kap. 5.1: Seite 126-128, Fig. 5.2, 5.3 (b)



**Trade-off** zwischen Qualitätskriterien: **Entdecktes Modell soll ...**

- **Angemessenheit (Fitness):**  
... Verhalten des Event-Logs zulassen.
- **Genauigkeit (Precision) - vermeide Unteranpassung (underfitting):**  
... kein Verhalten ohne Bezug zu Inhalt des Event-Logs zulassen.
- **Verallgemeinerung (Generalization) - vermeide Überanpassung (overfitting):**  
... Verallgemeinerung des Beispiel-Verhaltens im Event-Log sein.
- **Einfachheit (Simplicity):**  
... so einfach wie möglich sein.

15

## Literatur:

Wil van der Aalst: Process Mining: Discovery, Conformance and Enhancement of Business Processes

- Kap. 5.1: S. 128 letzter Abschnitt

# Überblick

## 2.4 Prozessextraktion



- Einführung und Beispiel
- **$\alpha$ -Algorithmus**
  - **Idee und Vorbereitungen**
  - Formalisierung
  - Beispiel
  - Weitere Beispiele
  - Einschränkungen
- Allgemeine Herausforderungen beim Process-Mining



**$\alpha$ -Algorithmus:** aus Menge von Folgen von Log-Daten GP-Modell als Petrinetz extrahieren.

Idee:

- Informationen über **Abfolge der Aktivitäten** durch Verwendung geeigneter **Ordnungs-Relationen** sammeln.
  - Ob Aktivitäten in verschiedenen Folgen in **Kausal-Beziehung** stehen, oder **parallel** bzw. **unabhängig** voneinander sind.
- Damit Petrinetz konstruieren.

Korrektheit des Algorithmus und benötigte Eigenschaften der Ordnungsrelationen nicht-trivial; hier nicht betrachtet.

17

## Literatur:

Wil van der Aalst: Process Mining: Discovery, Conformance and Enhancement of Business Processes

- Kap. 5.2: S. 129 letzter Abschnitt – S. 133



## Annahmen an zu erzeugende Petrinetze:

- **Ausführung:** jede **Transition erzeugt** jeweils ein **Ereignis** in **Log-Datei** mit zugehöriger Bezeichnung.
- Verschiedene Transitionen haben verschiedene Bezeichnungen.
- Keine Bogen-Vielfachheiten.

## Annahmen an Folge von Log-Daten:

- Log-Daten alle vom relevanten GP erzeugt.
- GP durch Petrinetz modellierbar.
- Aufteilung der Log-Daten auf Folgen in gegebener Menge entsprechen verschiedenen GP-Instanzen.
- Abstrahieren von mehrfachen GP-Instanzen mit demselben Verlauf (und von GP-Instanznummern).



Für eine Menge  $W$  von Folgen  $\sigma$  von Log-Daten (Folge  $\equiv$  GP-Instanz) und Elemente  $x, y$  in diesen Folgen (mit  $x \neq y$ ) definieren wir:

- **Direkte Nachfolge:**  $x >_w y$   
 $\exists \sigma \in W$  worin  $y$  direkt auf  $x$  folgt
- **Kausalität:**  $x \rightarrow_w y$   
 $x >_w y$  und nicht  $y >_w x$
- **Parallelität:**  $x \parallel_w y$   
 $x >_w y$  und  $y >_w x$
- **Unabhängigkeit:**  $x \#_w y$   
nicht  $x >_w y$  und nicht  $y >_w x$

Im Beispiel  
{ABCD, ACBD, AED}:

[Index  $W$  kann weggelassen werden, wenn klar ist, welche Log-Datei gemeint ist.]

19

## Literatur:

Wil van der Aalst: Process Mining: Discovery, Conformance and Enhancement of Business Processes

- Kap. 5.2: S.130-131



Für eine Menge  $W$  von Folgen  $\sigma$  von Log-Daten (Folge  $\equiv$  GP-Instanz) und Elemente  $x, y$  in diesen Folgen (mit  $x \neq y$ ) definieren wir:

- **Direkte Nachfolge:**  $x >_w y$   
 $\exists \sigma \in W$  worin  $y$  direkt auf  $x$  folgt
- **Kausalität:**  $x \rightarrow_w y$   
 $x >_w y$  und nicht  $y >_w x$
- **Parallelität:**  $x \parallel_w y$   
 $x >_w y$  und  $y >_w x$
- **Unabhängigkeit:**  $x \#_w y$   
nicht  $x >_w y$  und nicht  $y >_w x$

Im Beispiel  
{ABCD, ACBD, AED}:

- $A > B, B > C, C > D, A > C,$   
 $C > B, B > D, A > E, E > D$

[Index  $W$  kann weggelassen werden, wenn klar ist, welche Log-Datei gemeint ist.]

20

## Literatur:

Wil van der Aalst: Process Mining: Discovery, Conformance and Enhancement of Business Processes

- Kap. 5.2: S.130-131



Für eine Menge  $W$  von Folgen  $\sigma$  von Log-Daten (Folge  $\equiv$  GP-Instanz) und Elemente  $x, y$  in diesen Folgen (mit  $x \neq y$ ) definieren wir:

- **Direkte Nachfolge:**  $x >_w y$   
 $\exists \sigma \in W$  worin  $y$  direkt auf  $x$  folgt
- **Kausalität:**  $x \rightarrow_w y$   
 $x >_w y$  und nicht  $y >_w x$
- **Parallelität:**  $x \parallel_w y$   
 $x >_w y$  und  $y >_w x$
- **Unabhängigkeit:**  $x \#_w y$   
nicht  $x >_w y$  und nicht  $y >_w x$

Im Beispiel  
{ABCD, ACBD, AED}:

- $A > B, B > C, C > D, A > C,$   
 $C > B, B > D, A > E, E > D$
- $A \rightarrow B, A \rightarrow C, A \rightarrow E, B \rightarrow D,$   
 $C \rightarrow D, E \rightarrow D$

[Index  $W$  kann weggelassen werden, wenn klar ist, welche Log-Datei gemeint ist.]

21

## Literatur:

Wil van der Aalst: Process Mining: Discovery, Conformance and Enhancement of Business Processes

- Kap. 5.2: S.130-131



Für eine Menge  $W$  von Folgen  $\sigma$  von Log-Daten (Folge  $\equiv$  GP-Instanz) und Elemente  $x, y$  in diesen Folgen (mit  $x \neq y$ ) definieren wir:

- **Direkte Nachfolge:**  $x >_w y$   
 $\exists \sigma \in W$  worin  $y$  direkt auf  $x$  folgt
- **Kausalität:**  $x \rightarrow_w y$   
 $x >_w y$  und nicht  $y >_w x$
- **Parallelität:**  $x \parallel_w y$   
 $x >_w y$  und  $y >_w x$
- **Unabhängigkeit:**  $x \#_w y$   
nicht  $x >_w y$  und nicht  $y >_w x$

Im Beispiel  
{ABCD, ACBD, AED}:

- $A > B, B > C, C > D, A > C, C > B, B > D, A > E, E > D$
- $A \rightarrow B, A \rightarrow C, A \rightarrow E, B \rightarrow D, C \rightarrow D, E \rightarrow D$
- $B \parallel C$  (und symm.)

[Index  $W$  kann weggelassen werden, wenn klar ist, welche Log-Datei gemeint ist.]

22

## Literatur:

Wil van der Aalst: Process Mining: Discovery, Conformance and Enhancement of Business Processes

- Kap. 5.2: S.130-131



Für eine Menge  $W$  von Folgen  $\sigma$  von Log-Daten (Folge  $\equiv$  GP-Instanz) und Elemente  $x, y$  in diesen Folgen (mit  $x \neq y$ ) definieren wir:

- **Direkte Nachfolge:**  $x >_w y$   
 $\exists \sigma \in W$  worin  $y$  direkt auf  $x$  folgt
- **Kausalität:**  $x \rightarrow_w y$   
 $x >_w y$  und nicht  $y >_w x$
- **Parallelität:**  $x \parallel_w y$   
 $x >_w y$  und  $y >_w x$
- **Unabhängigkeit:**  $x \#_w y$   
nicht  $x >_w y$  und nicht  $y >_w x$

Im Beispiel

$\{ABCD, ACBD, AED\}$ :

- $A > B, B > C, C > D, A > C, C > B, B > D, A > E, E > D$
- $A \rightarrow B, A \rightarrow C, A \rightarrow E, B \rightarrow D, C \rightarrow D, E \rightarrow D$
- $B \parallel C$  (und symm.)
- $A \# D, B \# E, C \# E$  (und symm.)

[Index  $W$  kann weggelassen werden, wenn klar ist, welche Log-Datei gemeint ist.]

23

## Literatur:

Wil van der Aalst: Process Mining: Discovery, Conformance and Enhancement of Business Processes

- Kap. 5.2: S.130-131



I) (antisymmetrische) **Kausalität**, sowie (symmetrische) **Parallelität** und **Unabhängigkeit** definieren die 4 Möglichkeiten, wie 2 Elemente durch  $>$  in Relation stehen können.

II) **Korrektheit** des  $\alpha$ -Algorithmus: Unter gegebenen Annahmen an Log-Daten und Petrinetz sind...

- 1) ... Folgen von Log-Daten durch oben definierte Ordnungs-Relationen **charakterisierbar** (ggf. inkl. Verallgemeinerung).
- 2) ... aus Ordnungs-Relationen Petrinetz **ableitbar**, sodass die vom Petrinetz zur Laufzeit gefeuerten Folgen von Transitionen ursprünglich gegebenen Folgen von Log-Daten enthalten (evtl. auch andere !).

(Verzichten auf (nicht-trivialen) Beweis...)

## Literatur:

Wil van der Aalst: Process Mining: Discovery, Conformance and Enhancement of Business Processes

- Kap. 5.2: S. 129-130



Ist aus diesen Relationen Petrinetz ableitbar, das diese Aussagen bestätigt ?

Einfachster Fall: **Kausalität**  $x \rightarrow y$  („sequence pattern“).

Welches („minimale“) (Teil-)Petrinetz erzeugt bei Ausführung Folgen von Transitionen, die durch diese eine Relation charakterisiert sind ?

25

## Literatur:

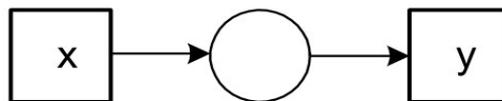
Wil van der Aalst: Process Mining: Discovery, Conformance and Enhancement of Business Processes

- Kap. 5.2: S. 129-130

Ist aus diesen Relationen Petrinetz ableitbar, das diese Aussagen bestätigt ?

Einfachster Fall: **Kausalität**  $x \rightarrow y$  („sequence pattern“)

Welches („minimale“) (Teil-)Petrinetz erzeugt bei Ausführung Folgen von Transitionen, die durch diese eine Relation charakterisiert sind ?



## Literatur:

Wil van der Aalst: Process Mining: Discovery, Conformance and Enhancement of Business Processes

- Kap. 5.2: S. 130-131, Fig. 5.4 (a)



Welches („minimale“) (Teil-)Petrinetz erzeugt Folgen von Transitionen, die jeweils durch folgende Relationen charakterisiert sind ?

$x \rightarrow y, x \rightarrow z, y || z$

$x \rightarrow y, x \rightarrow z, y \# z$

$x \rightarrow z, y \rightarrow z, x || y$

$x \rightarrow z, y \rightarrow z, x \# y$

27

## Literatur:

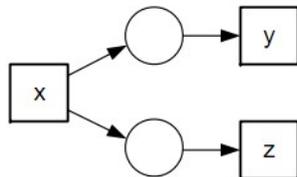
Wil van der Aalst: Process Mining: Discovery, Conformance and Enhancement of Business Processes

- Kap. 5.2: S. 130-131, Fig. 5.4

Welches („minimale“) (Teil-)Petrinetz erzeugt Folgen von Transitionen, die jeweils durch folgende Relationen charakterisiert sind ?

$x \rightarrow y, x \rightarrow z, y \parallel z$

$x \rightarrow y, x \rightarrow z, y \# z$



**AND-split  
pattern**

$x \rightarrow z, y \rightarrow z, x \parallel y$

$x \rightarrow z, y \rightarrow z, x \# y$

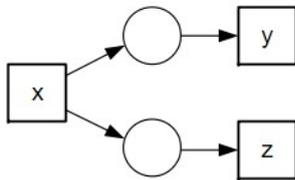
## Literatur:

Wil van der Aalst: Process Mining: Discovery, Conformance and Enhancement of Business Processes

- Kap. 5.2: S. 130-131, Fig. 5.4

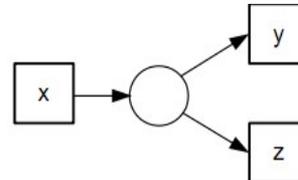
Welches („minimale“) (Teil-)Petrinetz erzeugt Folgen von Transitionen, die jeweils durch folgende Relationen charakterisiert sind ?

$x \rightarrow y, x \rightarrow z, y || z$



**AND-split  
pattern**

$x \rightarrow y, x \rightarrow z, y \# z$



**XOR-split  
pattern**

$x \rightarrow z, y \rightarrow z, x || y$

$x \rightarrow z, y \rightarrow z, x \# y$

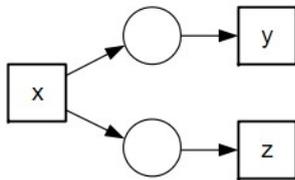
## Literatur:

Wil van der Aalst: Process Mining: Discovery, Conformance and Enhancement of Business Processes

- Kap. 5.2: S. 130-131, Fig. 5.4

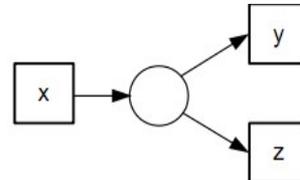
Welches („minimale“) (Teil-)Petrinetz erzeugt Folgen von Transitionen, die jeweils durch folgende Relationen charakterisiert sind ?

$x \rightarrow y, x \rightarrow z, y \parallel z$



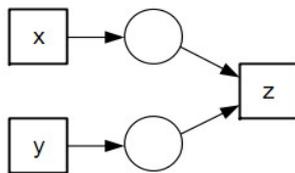
**AND-split  
pattern**

$x \rightarrow y, x \rightarrow z, y \# z$



**XOR-split  
pattern**

$x \rightarrow z, y \rightarrow z, x \parallel y$



**AND-join  
pattern**

$x \rightarrow z, y \rightarrow z, x \# y$

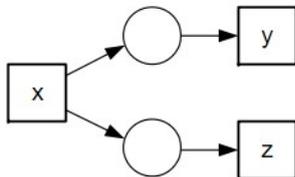
## Literatur:

Wil van der Aalst: Process Mining: Discovery, Conformance and Enhancement of Business Processes

- Kap. 5.2: S. 130-131, Fig. 5.4

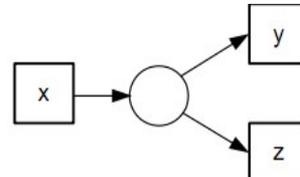
Welches („minimale“) (Teil-)Petrinetz erzeugt Folgen von Transitionen, die jeweils durch folgende Relationen charakterisiert sind ?

$x \rightarrow y, x \rightarrow z, y || z$



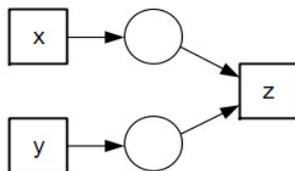
**AND-split  
pattern**

$x \rightarrow y, x \rightarrow z, y \# z$



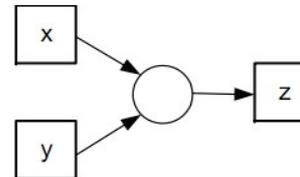
**XOR-split  
pattern**

$x \rightarrow z, y \rightarrow z, x || y$



**AND-join  
pattern**

$x \rightarrow z, y \rightarrow z, x \# y$



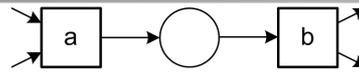
**XOR-join  
pattern**

31

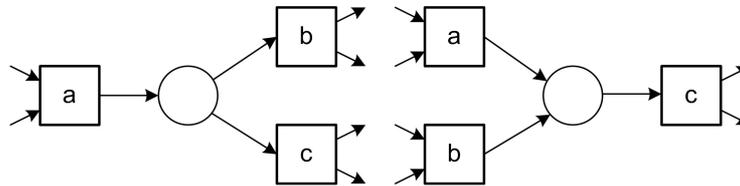
## Literatur:

Wil van der Aalst: Process Mining: Discovery, Conformance and Enhancement of Business Processes

- Kap. 5.2: S. 130-131, Fig. 5.4

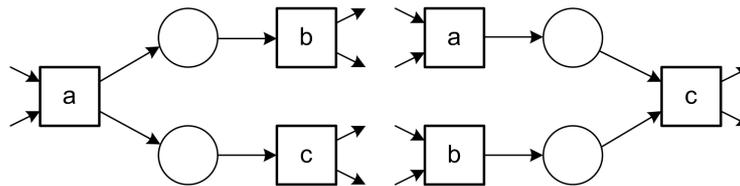


(a) sequence pattern:  $a \rightarrow b$



(b) XOR-split pattern:  
 $a \rightarrow b$ ,  $a \rightarrow c$ , and  $b \# c$

(c) XOR-join pattern:  
 $a \rightarrow c$ ,  $b \rightarrow c$ , and  $a \# b$



(d) AND-split pattern:  
 $a \rightarrow b$ ,  $a \rightarrow c$ , and  $b \parallel c$

(e) AND-join pattern:  
 $a \rightarrow c$ ,  $b \rightarrow c$ , and  $a \parallel b$

## Literatur:

Wil van der Aalst: Process Mining: Discovery, Conformance and Enhancement of Business Processes

- Kap. 5.2: S. 130-131, Fig. 5.4

# Überblick

## 2.4 Prozessextraktion



- Einführung und Beispiel
- **$\alpha$ -Algorithmus**
  - Idee und Vorbereitungen
  - **Formalisierung**
  - Beispiel
  - Weitere Beispiele
  - Einschränkungen
- Allgemeine Herausforderungen beim Process-Mining



**Gegeben:** **Workflow-Log**  $W$  = Menge von Folgen  $\sigma$  von Log-Daten  
(z.B. ABCD).

**Gesucht:** Workflow-Schema als **Petrinetz**

$\alpha(W) = (\text{Stellen } P_W, \text{ Transitionen } T_W, \text{ Verbindungen } F_W)$

## Schritte 1-3 (Transitionen)

Annahme (zur Erinnerung):

- Ausführung: jede Transition erzeugt ein Ereignis in Log-Datei;  
Bezeichnung wie im Petrinetz.

→ Wie demnach Menge der Transitionen im Petrinetz definieren ?

34

## Literatur:

Wil van der Aalst: Process Mining: Discovery,  
Conformance and Enhancement of Business  
Processes

- Kap. 5.2: S. 133



**Gegeben:** Workflow-Log  $W$  = Menge von Folgen  $\sigma$  von Log-Daten  
(z.B. ABCD).

**Gesucht:** Workflow-Schema als **Petrinetz**

$\alpha(W) = (\text{Stellen } P_W, \text{ Transitionen } T_W, \text{ Verbindungen } F_W)$

**Schritte 1-3 (Transitionen):**

1.  $T_W = \{t \in T \mid \exists \sigma \in W t \in \sigma\}$  **Transitionen** (= Vereinigung der Transitions-  
mengen der Log-Daten)
2.  $T_I = \{t \in T \mid \exists \sigma \in W t = \text{first}(\sigma)\}$  **Start-Transitionen**
3.  $T_O = \{t \in T \mid \exists \sigma \in W t = \text{last}(\sigma)\}$  **End-Transitionen**

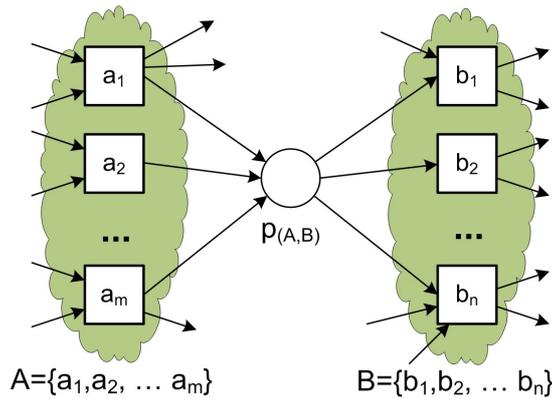
35

## Literatur:

Wil van der Aalst: Process Mining: Discovery,  
Conformance and Enhancement of Business  
Processes

- Kap. 5.2: S. 133

# Der $\alpha$ -Algorithmus Schritt 4: Kandidaten für Stellen (Idee)



	$a_1$	$a_2$	...	$a_m$	$b_1$	$b_2$	...	$b_n$
$a_1$	Welche Relationen (wie vorher definiert) zwischen Transitionen $a_1, \dots, a_m, b_1, \dots, b_n$ sind in dieser Situation in der Logdatei zu erwarten ?							
$a_2$								
...								
$a_m$								
$b_1$	Wie könnte man dann die Stelle $p_{(A,B)}$ aus der Logdatei ableiten ?							
$b_2$								
...								
$b_n$								

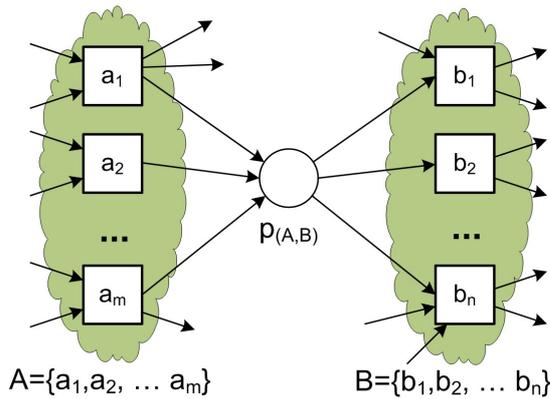


## Literatur:

Wil van der Aalst: Process Mining: Discovery, Conformance and Enhancement of Business Processes

- Kap. 5.2: S. 134, Fig.5.7, Tabelle 5.4

# Der $\alpha$ -Algorithmus Schritt 4: Kandidaten für Stellen (Idee)



→  $p_{(A,B)} = (A,B)$  Repräsentant für **Stellen**,  
die Paare von **kausal aufeinander-  
folgenden Transitionen** verbinden  
(Unabhängigkeit: gleichzeitig möglich).

Via Transitionsmengen (A,B), weil Stelle  
nicht direkt in Logdaten beobachtbar.

Wie allgemein definieren ?

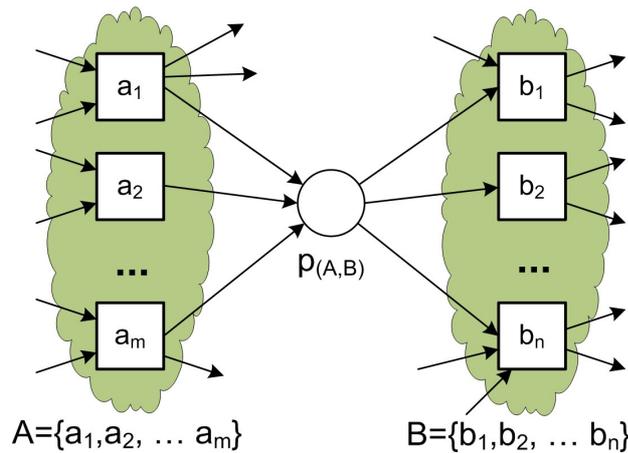
	$a_1$	$a_2$	...	$a_m$	$b_1$	$b_2$	...	$b_n$
$a_1$	#	#	...	#	→	→	...	→
$a_2$	#	#	...	#	→	→	...	→
...	...	...	...	...	...	...	...	...
$a_m$	#	#	...	#	→	→	...	→
$b_1$	←	←	...	←	#	#	...	#
$b_2$	←	←	...	←	#	#	...	#
...	...	...	...	...	...	...	...	...
$b_n$	←	←	...	←	#	#	...	#



## Literatur:

Wil van der Aalst: Process Mining: Discovery,  
Conformance and Enhancement of Business  
Processes

- Kap. 5.2: S. 134, Fig.5.7, Tabelle 5.4



$A = \{a_1, a_2, \dots, a_m\}$

$B = \{b_1, b_2, \dots, b_n\}$

$X_W$ : „verallgemeinerte

Kausalitätsrelationen“:

Kandidaten für **Stellen**, um Paare von **kausal aufeinanderfolgenden Transitionen** zu verbinden.

Beispiele:

$(\{a\}, \{b\}) \in X_W$  für alle  $a \rightarrow b$

„Kandidaten“ weil teilweise gegenseitig redundant (Teilmengen (A', B') auch dabei).

Wie könnte man das reparieren ?

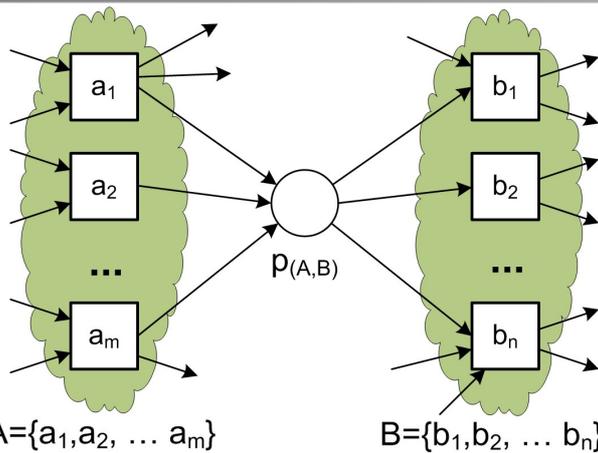
$$4. X_W = \{ (A, B) \mid A \subseteq T_W \wedge A \text{ 开 } \emptyset \wedge B \subseteq T_W \wedge B \text{ 开 } \emptyset \wedge \forall_{a \in A} \forall_{b \in B} a \rightarrow_W b \wedge \forall_{a_1, a_2 \in A} a_1 \#_W a_2 \wedge \forall_{b_1, b_2 \in B} b_1 \#_W b_2 \}$$

38

## Literatur:

Wil van der Aalst: Process Mining: Discovery, Conformance and Enhancement of Business Processes

- Kap. 5.2: S. 134, Fig.5.7, Tabelle 5.4



$Y_W$ : „Maximale  
Kausalitätsrelationen“.

Maximalität: soweit möglich  
keine unnötigen Stellen  
erzeugen, um Paare von kausal  
aufeinander folgenden  
Transitionen zu verbinden.

$$5. Y_W = \{(A, B) \in X_W \mid \forall (A', B') \in X_W \quad A \subseteq A' \wedge B \subseteq B' \Rightarrow (A, B) = (A', B')\}$$

$$6. P_W = \{(p_{(A,B)}) \mid (A, B) \in Y_W\} \cup \{i_W, o_W\}$$

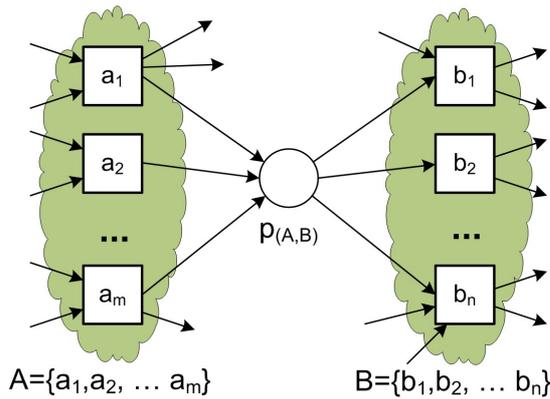
**Stellen:** Jede maximale Kausalitätsrelation definiert genau eine Stelle.  
Zusätzlich:  $i_W, o_W$ : eine Start- und eine Endstelle

39

## Literatur:

Wil van der Aalst: Process Mining: Discovery,  
Conformance and Enhancement of Business  
Processes

- Kap. 5.2: S. 134, Fig.5.7, Tabelle 5.4, S.135



Haben Transitionen und Stellen.  
Was fehlt für Petrinetz und wie  
könnte man das aus Logdatei  
ableiten ?

	$a_1$	$a_2$	...	$a_m$	$b_1$	$b_2$	...	$b_n$
$a_1$	#	#	...	#	→	→	...	→
$a_2$	#	#	...	#	→	→	...	→
...	...	...	...	...	...	...	...	...
$a_m$	#	#	...	#	→	→	...	→
$b_1$	←	←	...	←	#	#	...	#
$b_2$	←	←	...	←	#	#	...	#
...	...	...	...	...	...	...	...	...
$b_n$	←	←	...	←	#	#	...	#

## Literatur:

Wil van der Aalst: Process Mining: Discovery, Conformance and Enhancement of Business Processes

- Kap. 5.2: S. 134, Fig.5.7, Tabelle 5.4



$$7. F_W = \{(a, p_{(A,B)}) \mid (A,B) \in Y_W \wedge a \in A\} \cup \\ \{p_{(A,B)}, b \mid (A,B) \in Y_W \wedge b \in B\} \cup \\ \{(i_W, t) \mid t \in T_I\} \cup \\ \{(t, o_W) \mid t \in T_O\}$$

**Verbindungen:** Durch Kausalitätsrelationen definierte Verbindungen.  
Zusätzlich: Verbindungen zwischen Start- bzw. Endstellen und  
Start- bzw. Endtransitionen.

$$8. \alpha(W) = (P_W, T_W, F_W) \quad \text{Resultierendes Petrinetz.}$$

## Literatur:

Wil van der Aalst: Process Mining: Discovery,  
Conformance and Enhancement of Business  
Processes

- Kap. 5.2: S.135



## Frage:

Wie würde man **Korrektheit** des Algorithmus zeigen ?



## Frage:

Wie würde man **Korrektheit** des Algorithmus zeigen ?

## Antwort:

Z.z.: erhaltenes Petrinetz gibt bei Ausführung **ursprüngliche Folgen von Logdaten** aus.

Gilt i.A. nur unter o.g. Annahmen (z.B. Logdaten von einem Petrinetz generiert).

Petrinetz kann i.A. noch andere Folgen von Logdaten ausgeben !

# Überblick

## 2.4 Prozessextraktion



- Einführung und Beispiel
- **$\alpha$ -Algorithmus**
  - Idee und Vorbereitungen
  - Formalisierung
  - **Beispiel**
  - Weitere Beispiele
  - Einschränkungen
- Allgemeine Herausforderungen beim Process-Mining



1.  $T_W = \{$

Ursprungs-Log:  
{ABCD, ACBD, AED}

$$T_W = \{t \in T \mid \exists \sigma \in W, t \in \sigma\}$$

Extrahiere alle  
Transitionen  
aus dem Log



1.  $T_W = \{A, B, C, D\}$

Ursprungs-Log:  
 $\{ABCD, ACBD, AED\}$

$$T_W = \{t \in T \mid \exists \sigma \in W, t \in \sigma\}$$

Extrahiere alle  
Transitionen  
aus dem Log



1.  $T_W = \{A, B, C, D, E\}$

Ursprungs-Log:  
 $\{ABCD, ACBD, AED\}$

$$T_W = \{t \in T \mid \exists \sigma \in W, t \in \sigma\}$$

Extrahiere alle  
Transitionen  
aus dem Log



1.  $T_W = \{A, B, C, D, E\}$

2.  $T_I = \{$

Ursprungs-Log:  
 $\{ABCD, ACBD, AED\}$

$$T_I = \{t \in T \mid \exists \sigma \in W \ t = \text{first}(\sigma)\}$$

Samme  
Starttransitionen aus  
einzelnen Logs



1.  $T_W = \{A, B, C, D, E\}$

2.  $T_I = \{A\}$

Ursprungs-Log:  
 $\{ABCD, ACBD, AED\}$

$$T_I = \{t \in T \mid \exists \sigma \in W \ t = \text{first}(\sigma)\}$$

Samme  
Starttransitionen aus  
einzelnen Logs



1.  $T_W = \{A, B, C, D, E\}$

2.  $T_I = \{A\}$

3.  $T_O = \{$

$T_I = \{t \in T \mid \exists \sigma \in W, t = \text{last}(\sigma)\}$   
Samme  
Endtransitionen aus  
einzelnen Logs

Ursprungs-Log:  
 $\{ABCD, ACBD, AED\}$

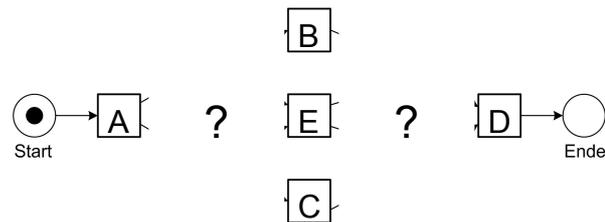
1.  $T_W = \{A, B, C, D, E\}$

2.  $T_I = \{A\}$

3.  $T_O = \{D\}$

Ursprungs-Log:  
 $\{ABC\color{red}D, AC\color{red}B\color{red}D, A\color{red}E\color{red}D\}$

$T_I \cup \{t \in T \mid \exists \sigma \in W \ t = \text{last}(\sigma)\}$   
Samme  
Endtransitionen aus  
einzelnen Logs



1.  $T_W =$

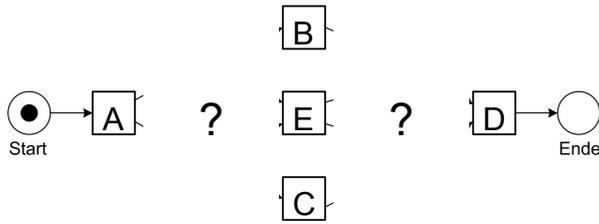
2.  $T_I = \{$

3.  $T_O =$

Bilde Kandidaten  
für Stellen.

Ursprungs-Log:  
{ABCD, ACBD, AED}

4.  $X_W = \{$



NB:  
A: Event / Transition  
A: Menge von Events / Transitionen

$$X_W \sqsubseteq \{ (A, B) \mid A \subseteq T_W \wedge$$

$$B \subseteq T_W \wedge$$

$$\forall a \in A \forall b \in B a \rightarrow_W b \wedge$$

$$\forall a_1, a_2 \in A a_1 \#_W a_2 \wedge$$

$$\forall b_1, b_2 \in B b_1 \#_W b_2 \}$$

Bilde Kandidaten für Stellen:  
Bilde Teilmengen mit unabhängigen Elementen  
als Kandidaten für Mengen **A** und **B**

Ursprungs-Log:  
{ABCD, ACBD, AED}

4.  $X_W = \{$

a>b  
a>c  
a>e  
b>c  
b>d  
c>b  
c>d  
e>d

a→b  
a→c  
a→e  
b→d  
c→d  
e→d

b||c  
c||b

b#e  
e#b  
c#e  
a#d  
...

$X_W \models \{ (A, B) \mid A \subseteq T_W \wedge$   
 $B \subseteq T_W \wedge$   
 $\forall a \in A \forall b \in B a \rightarrow_W b \wedge$   
 $\forall a_1, a_2 \in A a_1 \#_W a_2 \wedge$   
 $\forall b_1, b_2 \in B b_1 \#_W b_2 \}$



Bilde Kandidaten für Stellen:  
Bilde Teilmengen mit unabhängigen Elementen  
als Kandidaten für Mengen **A** und **B**

$\{A\}, \{A, D\}$

Ursprungs-Log:  
 $\{\overline{A}BCD, A\overline{C}BD, A\overline{E}D\}$

Alle andere Transitionen  
abhängig von A, bis auf D

3.  $T_o = \{D\}$

4.  $X_w = \{$

a>b	a→b	b  c	b#e
a>c	a→c	c  b	e#b
a>e	a→e		c#e
b>c	b→d		a#d
b>d	c→d		...
c>b	e→d		
c>d			
e>d			

$$X_w \sqsubseteq \{(A, B) \mid A \subseteq T_w \wedge B \subseteq T_w \wedge \forall a \in A \forall b \in B a \rightarrow_w b \wedge \forall a_1, a_2 \in A a_1 \#_w a_2 \wedge \forall b_1, b_2 \in B b_1 \#_w b_2\}$$



Bilde Kandidaten für Stellen:  
Bilde Teilmengen mit unabhängigen Elementen  
als Kandidaten für Mengen **A** und **B**  
 $\{A\}, \{A,D\}, \{B\}, \{B,E\}$

Ursprungs-Log:  
 $\{A \overline{B} C D, A C \overline{B} D, A \overline{A} E D\}$

A, C, D abhängig von B  
E unabhängig

3.  $T_o = \{D\}$

4.  $X_w = \{$

a>b  
a>c  
a>e  
b>c  
b>d  
c>b  
c>d  
e>d

a→b  
a→c  
a→e  
b→d  
c→d  
e→d

b||c  
c||b

b#e  
e#b  
c#e  
a#d  
...

$X_w \models \{(A, B) \mid A \subseteq T_w \wedge B \subseteq T_w \wedge \forall a \in A \forall b \in B a \rightarrow_w b \wedge \forall a_1, a_2 \in A a_1 \#_w a_2 \wedge \forall b_1, b_2 \in B b_1 \#_w b_2\}$

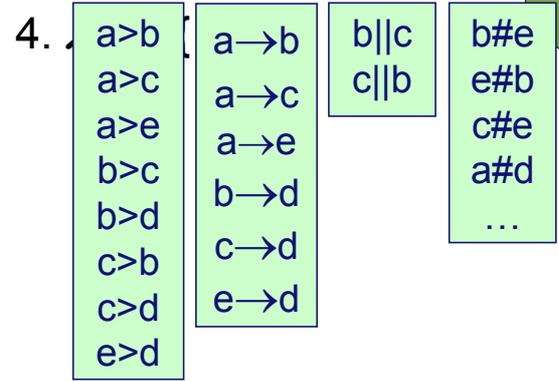


Bilde Kandidaten für Stellen:  
Bilde Teilmengen mit unabhängigen Elementen  
als Kandidaten für Mengen **A** und **B**  
{A}, {A,D}, {B}, {B,E}, {C}, {C,E}, {D}, {E}

Ursprungs-Log:  
{ABCD, ACBD, AED}

3.  $T_0 = \{D\}$

{A,D} nicht relevant  
obwohl unabhängig,  
weil kein gemeinsamer  
Vorgänger / Nachfolger



$$X_W = \{(A, B) \mid A \subseteq T_W \wedge B \subseteq T_W \wedge \forall a \in A \forall b \in B a \rightarrow_W b \wedge \forall a_1, a_2 \in A a_1 \#_W a_2 \wedge \forall b_1, b_2 \in B b_1 \#_W b_2\}$$



1. Bilde Kandidaten für Stellen:  
Bilde Paare  $(A, B)$ , sodass Menge  $A$  nur kausale  
2. Vorgänger der Elemente in Menge  $B$  enthält:  
 $\{A\}, \{A, D\}, \{B\}, \{B, E\}, \{C\}, \{C, E\}, \{D\}, \{E\}$

Ursprungs-Log:  
 $\{ABCD, ACBD, AED\}$

3.  $T_0 = \{D\}$

4.  $X_W = \{(\{A\}, \{B\}), (\{A\}, \{C\}), (\{A\}, \{E\}),$   
 $(\{A\}, \{B, E\}), (\{A\}, \{C, E\}),$

$X_W = \{(A, B) \mid A \subseteq T_W \wedge$   
 $B \subseteq T_W \wedge$   
 $\forall a \in A \forall b \in B a \rightarrow_W b \wedge$   
 $\forall a_1, a_2 \in A a_1 \#_W a_2 \wedge$   
 $\forall b_1, b_2 \in B b_1 \#_W b_2\}$

a>b	a→b	b  c	b#e
a>c	a→c	c  b	e#b
a>e	a→e		c#e
b>c	b→d		a#d
b>d	c→d		...
c>b	e→d		
c>d			
e>d			

1. Bilde Kandidaten für Stellen:  
Bilde Paare  $(A, B)$ , sodass Menge  $A$  nur kausale  
2. Vorgänger der Elemente in Menge  $B$  enthält:  
 $\{A\}, \{A, D\}, \{B\}, \{B, E\}, \{C\}, \{C, E\}, \{D\}$

Ursprungs-Log:  
 $\{ABCD, ACBD, AED\}$

$\{C\}$  nicht wegen  
BC und CB:  
Abfolge nicht eindeutig

3.  $T_0 = \{D\}$

4.  $X_W = \{(\{A\}, \{B\}), (\{A\}, \{C\}), (\{A\}, \{E\}), (\{B\}, \{D\}),$   
 $(\{A\}, \{B, E\}), (\{A\}, \{C, E\}),$

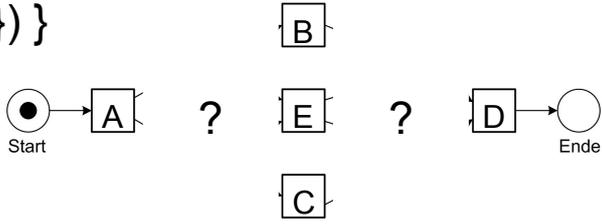
$X_W = \{(A, B) \mid A \subseteq T_W \wedge B \subseteq T_W \wedge$   
 $\forall a \in A \forall b \in B a \rightarrow_W b \wedge$   
 $\forall a_1, a_2 \in A a_1 \#_W a_2 \wedge$   
 $\forall b_1, b_2 \in B b_1 \#_W b_2\}$

a > b	a → b	b    c	b # e
a > c	a → c	c    b	e # b
a > e	a → e		c # e
b > c	b → d		a # d
b > d	c → d		...
c > b	e → d		
c > d			
e > d			

1  
2  
3.  $X_w = \{D\}$   
 $X_w$ : Kandidaten für Stellen.

Ursprungs-Log:  
{ABCD, ACBD, AED}

4.  $X_w = \{(\{A\}, \{B\}), (\{A\}, \{C\}), (\{A\}, \{E\}), (\{B\}, \{D\}), (\{C\}, \{D\}), (\{E\}, \{D\}), (\{A\}, \{B, E\}), (\{A\}, \{C, E\}), (\{B, E\}, \{D\}), (\{C, E\}, \{D\})\}$



$X_w \models \{ (A, B) \mid A \subseteq T_w \wedge B \subseteq T_w \wedge \forall a \in A \forall b \in B a \rightarrow_w b \wedge \forall a_1, a_2 \in A a_1 \#_w a_2 \wedge \forall b_1, b_2 \in B b_1 \#_w b_2 \}$



1.  $T_W = \{A, B, C, D, E\}$

Ursprungs-Log:  
{ABCD, ACBD, AED}

2.  $T_I = \{A\}$

3.  $T_O = \{ (A, B) \in X_W \mid \exists A', B' \in X_W, A \subseteq A' \wedge B \subseteq B' \Rightarrow (A, B) = (A', B') \}$   
Entferne alle Paare  $(A, B)$  aus  $X_W$  die Teilmenge eines anderen Paares sind.

4.  $X_W = \{(\{A\}, \{B\}), (\{A\}, \{C\}), (\{A\}, \{E\}), (\{B\}, \{D\}), (\{C\}, \{D\}), (\{E\}, \{D\}), (\{A\}, \{B, E\}), (\{A\}, \{C, E\}), (\{B, E\}, \{D\}), (\{C, E\}, \{D\})\}$

5.  $Y_W = \{$



$$1. T_W = \{A, B, C, D, E\}$$

Ursprungs-Log:  
{ABCD, ACBD, AED}

$$2. T_I = \{A\}$$

$$3. T_O = \{ (A, B) \in X_W \mid \exists A', B' \in X_W, A \subseteq A' \wedge B \subseteq B' \Rightarrow (A, B) = (A', B') \}$$

Entferne alle Paare (A, B) aus  $X_W$  die Teilmenge eines anderen Paares sind.

$$4. X_W = \{ \cancel{(\{A\}, \{B\})}, (\{A\}, \{C\}), \cancel{(\{A\}, \{E\})}, (\{B\}, \{D\}), (\{C\}, \{D\}), (\{E\}, \{D\}), (\{A\}, \{B, E\}), (\{A\}, \{C, E\}), (\{B, E\}, \{D\}), (\{C, E\}, \{D\}) \}$$

$$5. Y_W = \{ (\{A\}, \{B, E\}) \}$$



$$1. T_W = \{A, B, C, D, E\}$$

Ursprungs-Log:  
{ABCD, ACBD, AED}

$$2. T_I = \{A\}$$

$$3. T_O = \{ (A, B) \in X_W \mid \exists A', B' \in X_W, A \subseteq A' \wedge B \subseteq B' \Rightarrow (A, B) = (A', B') \}$$

Entferne alle Paare (A, B) aus  $X_W$  die Teilmenge eines anderen Paares sind.

$$4. X_W = \{ \{A\}, \{B\}, \{C\}, \{D\}, \{E\}, \{A, B\}, \{A, C\}, \{A, E\}, \{B, D\}, \{C, D\}, \{E, D\}, \{A, \{B, E\}\}, \{A, \{C, E\}\}, \{B, E, \{D\}\}, \{C, E, \{D\}\} \}$$

$$5. Y_W = \{ \{A, \{B, E\}\}, \{A, \{C, E\}\}, \{B, E, \{D\}\}, \{C, E, \{D\}\} \}$$



1.  $T_W = \{A, B, C, D, E\}$

Ursprungs-Log:  
{ABCD, ACBD, AED}

2.  $T_I = \{A\}$

3.  $T_O = \{ (A, B) \in X_W \mid \exists A', B' \in X_W, A \subseteq A' \wedge B \subseteq B' \Rightarrow (A, B) = (A', B') \}$   
Entferne alle Paare  $(A, B)$  aus  $X_W$  die Teilmenge eines anderen Paares sind.

4.  $X_W = \{(\{A\}, \{B\}), (\{A\}, \{C\}), (\{A\}, \{E\}), (\{B\}, \{D\}), (\{C\}, \{D\}),$   
 $(\{E\}, \{D\}), (\{A\}, \{B, E\}), (\{A\}, \{C, E\}), (\{B, E\}, \{D\}),$   
 $(\{C, E\}, \{D\})\}$

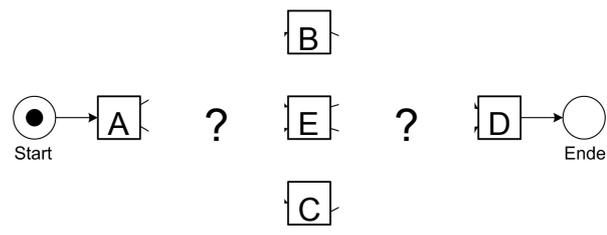
5.  $Y_W = \{(\{A\}, \{B, E\}), (\{A\}, \{C, E\}), (\{B, E\}, \{D\}),$   
 $(\{C, E\}, \{D\})\}$



$$5. \mathcal{Y}_W = \{(\{A\}, \{B, E\}), (\{A\}, \{C, E\}), (\{B, E\}, \{D\}), (\{C, E\}, \{D\})\}$$

$$6. \mathcal{P}_W = \{$$

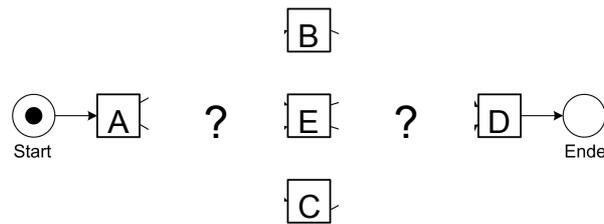
$\mathcal{P}_W \sqsupseteq \{(p_{(A,B)}) \mid (A,B) \in \mathcal{Y}_W\} \cup \{i_W, o_W\}$   
 Füge für jedes Paar  $(A,B)$  eine Stelle, sowie eine Start- und Endstelle ein.



$$5. Y_W = \{(\{A\}, \{B, E\}), (\{A\}, \{C, E\}), (\{B, E\}, \{D\}), (\{C, E\}, \{D\})\}$$

$$6. P_W = \{i_W, o_W\}$$

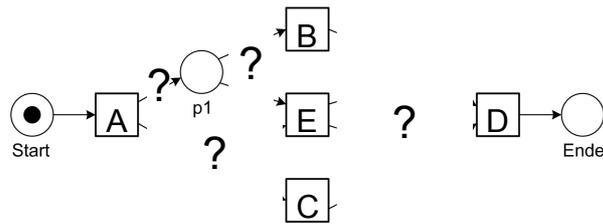
$P_W = \{p_{(A,B)} \mid (A,B) \in Y_W\} \cup \{i_W, o_W\}$   
 Füge für jedes Paar  $(A,B)$  eine Stelle, sowie eine Start- und Endstelle ein.



$$5. Y_W = \{(\{A\}, \{B, E\}), (\{A\}, \{C, E\}), (\{B, E\}, \{D\}), (\{C, E\}, \{D\})\}$$

$$6. P_W = \{i_W, o_W, p(\{A\}, \{B, E\})\}$$

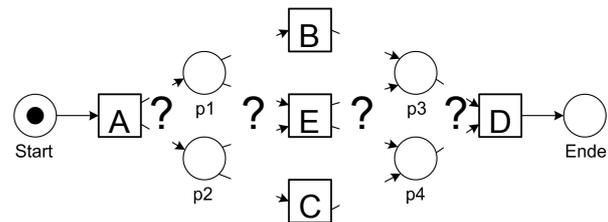
$P_W = \{(p_{(A,B)} | \{A, B\} \in Y_W) \cup \{i_W, o_W\}$   
 Füge für jedes Paar  $(A, B)$  eine Stelle, sowie eine Start- und Endstelle ein.



$$5. \mathcal{Y}_W = \{(\{A\}, \{B, E\}), (\{A\}, \{C, E\}), (\{B, E\}, \{D\}), (\{C, E\}, \{D\})\}$$

$$6. \mathcal{P}_W = \{i_W, o_W, p(\{A\}, \{B, E\}), p(\{A\}, \{C, E\}), p(\{B, E\}, \{D\}), p(\{C, E\}, \{D\})\},$$

$\mathcal{P}_W \sqsupseteq \{p_{(A,B)} \mid (A,B) \in \mathcal{Y}_W\} \cup \{i_W, o_W\}$   
Füge für jedes Paar  $(A,B)$  eine Stelle, sowie eine Start- und Endstelle ein.



1.  $T_W = \{A, B, C, D, E\}$

2.  $T_I = \{A\}$       3.  $T_O = \{D\}$

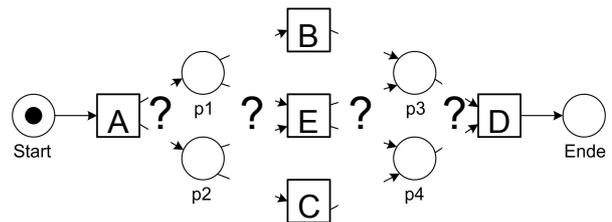
5.  $Y_W = \{(\{A\}, \{B, E\}), (\{A\}, \{C, E\}), (\{C, E\}, \{D\})\}$

6.  $P_W = \{i_W, o_W, p(\{A\}, \{B, E\}), p(\{A\}, \{C, E\}), p(\{B, E\}, \{D\}), p(\{C, E\}, \{D\})\}$

7.  $F_W = \{$

Füge die zugehörigen Bögen ein.

$$F_W = \{(a, p_{(A,B)}) \mid (A, B) \in Y_W \wedge a \in A\} \cup \{p_{(A,B)}, b \mid (A, B) \in Y_W \wedge b \in B\} \cup \{(i_W, t) \mid t \in T_I\} \cup \{(t, o_W) \mid t \in T_O\}$$



$$1. T_W = \{A, B, C, D, E\}$$

Füge die zugehörigen Bögen ein  
Verbinde Startstelle mit Starttransitionen  
Verbinde Endstelle mit Starttransitionen

$$F_W = \{(a, p_{(A,B)}) \mid (A, B) \in Y_W \wedge a \in A\} \cup$$

$$\{p_{(A,B)}, b \mid (A, B) \in Y_W \wedge b \in B\} \cup$$

$$\{(i_W, t) \mid t \in T_I\} \cup$$

$$\{(t, o_W) \mid t \in T_O\}$$

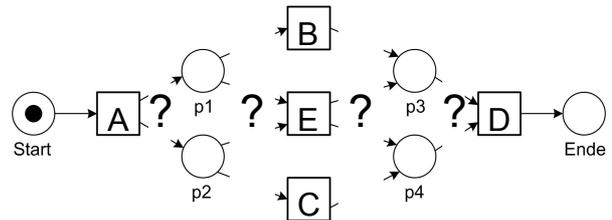
$$(\{C, E\}, \{D\})$$

$$6. P_W = \{i_W, o_W, p(\{A\}, \{B, E\}), p(\{A\}, \{C, E\}), p(\{B, E\}, \{D\}),$$

$$p(\{C, E\}, \{D\})\},$$

$$7. F_W = \{(i_W, A),$$

$$\dots, (D, o_W)$$



69

1.  $T_W = \{A, B, C, D, E\}$

2.  $T_I = \{A\}$       3.  $T_O = \{D\}$

$$F_W = \{(a, p_{(A,B)}) \mid (A, B) \in Y_W \wedge a \in A\} \cup$$

$$\{p_{(A,B)}, b \mid (A, B) \in Y_W \wedge b \in B\} \cup$$

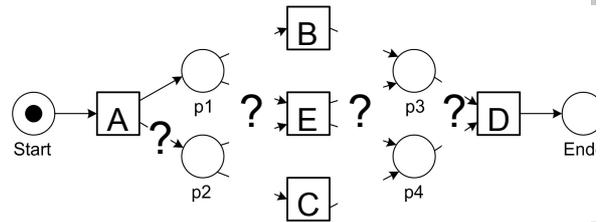
$$\{(i_W, t) \mid t \in T_I\} \cup$$

$$\{(t, o_W) \mid t \in T_O\}$$

Füge die zugehörigen Bögen ein:  
*Verbinde Elemente aus A eines  $Y_W$ -Tupels mit der zugehörigen Stelle*

6.  $P_W = \{i_W, o_W, p(\{A\}, \{B, E\}), p(\{A\}, \{C, E\}), p(\{B, E\}, \{D\}), p(\{C, E\}, \{D\})\}$ ,

7.  $F_W = \{(i_W, A), (A, p(\{A\}, \{B, E\}))$   
...,  $(D, o_W)$



1.  $T_W = \{A, B, C, D, E\}$

2.  $T_I = \{A\}$       3.  $T_O = \{D\}$

Füge die zugehörigen Bögen ein:  
Verbinde zugehörigen Stelle eines  
 $Y_W$  – Tupels mit den Elementen aus  $B$

$$F_W = \{(a, p_{(A,B)}) \mid (A, B) \in Y_W \wedge a \in A\} \cup$$

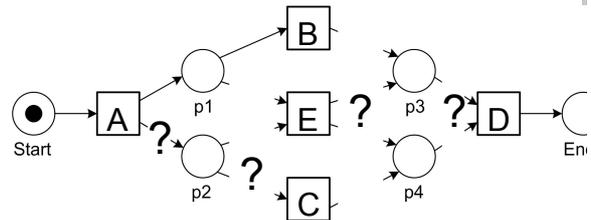
$$\{p_{(A,B)}, b \mid (A, B) \in Y_W \wedge b \in B\} \cup$$

$$\{(i_W, t) \mid t \in T_I\} \cup$$

$$\{(t, o_W) \mid t \in T_O\}$$

6.  $P_W = \{i_W, o_W, p(\{A\}, \{B, E\}), p(\{A\}, \{C, E\}), p(\{B, E\}, \{D\}), p(\{C, E\}, \{D\})\}$ ,

7.  $F_W = \{(i_W, A), (A, p(\{A\}, \{B, E\})), (p(\{A\}, \{B, E\}), B), \dots, (D, o_W)\}$



1.  $T_W = \{A, B, C, D, E\}$

2.  $T_I = \{A\}$       3.  $T_O = \{D\}$

5. ...  
Füge die zugehörigen Bögen ein: Etc.

$$F_W = \{(a, p_{(A,B)}) \mid (A, B) \in Y_W \wedge a \in A\} \cup$$

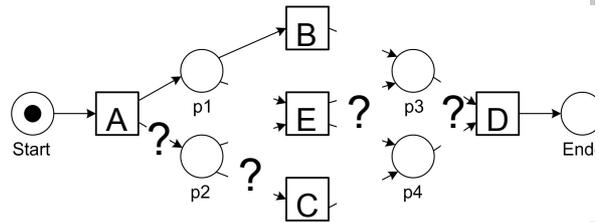
$$\{p_{(A,B)}, b \mid (A, B) \in Y_W \wedge b \in B\} \cup$$

$$\{(i_W, t) \mid t \in T_I\} \cup$$

$$\{(t, o_W) \mid t \in T_O\}$$

6.  $P_W = \{i_W, o_W, p(\{A\}, \{B, E\}), p(\{A\}, \{C, E\}), p(\{B, E\}, \{D\}), p(\{C, E\}, \{D\})\}$ ,

7.  $F_W = \{(i_W, A), (A, p(\{A\}, \{B, E\})), (p(\{A\}, \{B, E\}), B), \dots, (D, o_W)\}$



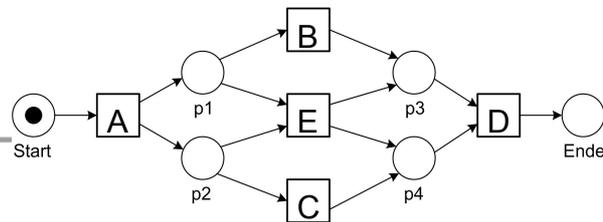
$$5. \mathcal{Y}_W = \{(\{A\}, \{B, E\}), (\{A\}, \{C, E\}), (\{B, E\}, \{D\}), (\{C, E\}, \{D\})\}$$

$$6. \mathcal{P}_W = \{i_W, o_W, p(\{A\}, \{B, E\}), p(\{A\}, \{C, E\}), p(\{B, E\}, \{D\}), p(\{C, E\}, \{D\})\},$$

$$7. \mathcal{F}_W = \{(i_W, A), (A, p(\{A\}, \{B, E\})), (p(\{A\}, \{B, E\}), B), \dots, (D, o_W)\}$$

$$8. \alpha(W) = (\mathcal{P}_W, \mathcal{T}_W, \mathcal{F}_W)$$

$\alpha(W) = (\mathcal{P}_W, \mathcal{T}_W, \mathcal{F}_W)$   
Das extrahierte Petrinetz ist nun definiert.



# Überblick

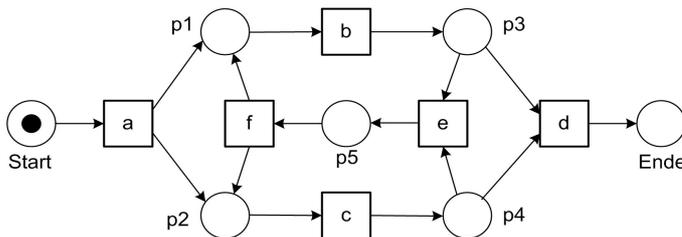
## 2.4 Prozessextraktion



- Einführung und Beispiel
- **$\alpha$ -Algorithmus**
  - Idee und Vorbereitungen
  - Formalisierung
  - Beispiel
  - **Weitere Beispiele**
  - Einschränkungen
- Allgemeine Herausforderungen beim Process-Mining

$$L_2 = [\langle a, b, c, d \rangle^3, \langle a, c, b, d \rangle^4, \langle a, b, c, e, f, b, c, d \rangle^2, \langle a, b, c, e, f, c, b, d \rangle, \langle a, c, b, e, f, b, c, d \rangle^2, \langle a, c, b, e, f, b, c, e, f, c, b, d \rangle]$$

	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
<i>a</i>	#	→	→	#	#	#
<i>b</i>	←	#		→	→	←
<i>c</i>	←		#	→	→	←
<i>d</i>	#	←	←	#	#	#
<i>e</i>	#	←	←	#	#	→
<i>f</i>	#	→	→	#	←	#



Welches **zusätzliche Verhalten** ergibt sich aus Petrinetz im Vergleich zu ursprünglichen Logdaten ?

75

### Literatur:

Wil van der Aalst: Process Mining: Discovery, Conformance and Enhancement of Business Processes

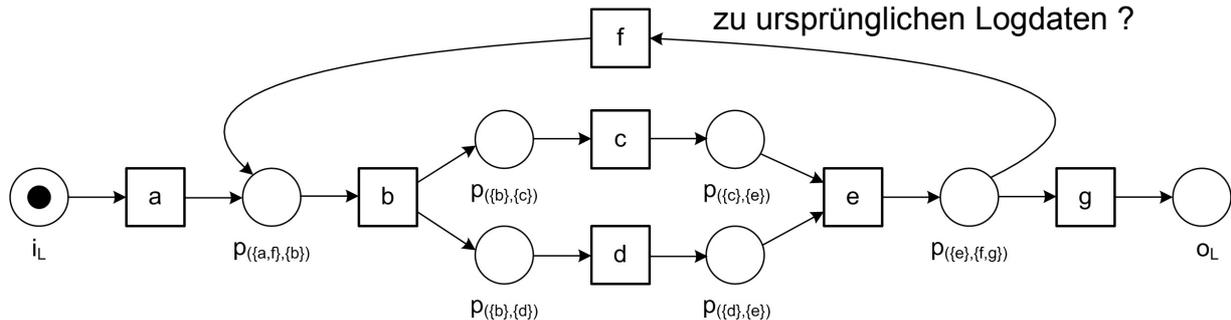
- Obere Abbildung: Kap. 5.1: S.126
- Mittlere Abbildung: Kap. 5.2: S.131, Tabelle 5.2
- Untere Abbildung: Kap. 5.1: S.127, Fig. 5.2



$L_3 = [\langle a, b, c, d, e, f, b, d, c, e, g \rangle,$   
 $\langle a, b, d, c, e, g \rangle^2,$   
 $\langle a, b, c, d, e, f, b, c, d, e, f, b, d, c, e, g \rangle]$

	a	b	c	d	e	f	g
a	#	→	#	#	#	#	#
b	←	#	→	→	#	←	#
c	#	←	#		→	#	#
d	#	←		#	→	#	#
e	#	#	←	←	#	→	→
f	#	→	#	#	←	#	#
g	#	#	#	#	←	#	#

Welches **zusätzliche Verhalten** ergibt sich aus Petrinetz im Vergleich zu ursprünglichen Logdaten ?



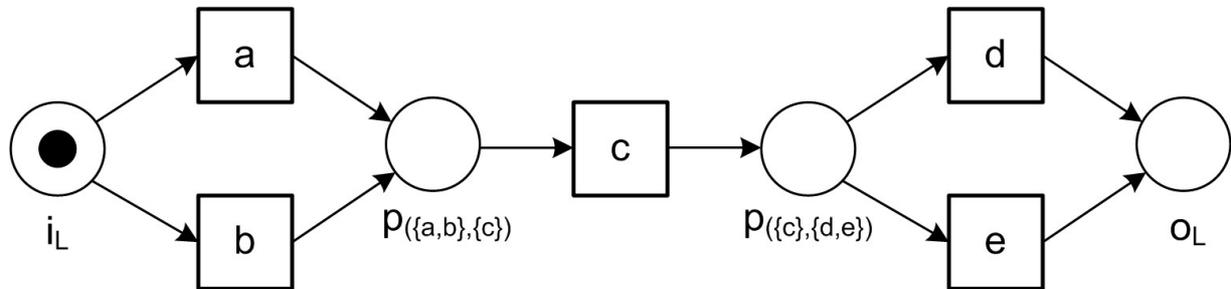
76

## Literatur:

Wil van der Aalst: Process Mining: Discovery, Conformance and Enhancement of Business Processes

- Kap. 5.2: S.132, Fig. 5.5

$$L_4 = [\langle a, c, d \rangle^{45}, \langle b, c, d \rangle^{42}, \langle a, c, e \rangle^{38}, \langle b, c, e \rangle^{22}]$$



77

**Literatur:**

Wil van der Aalst: Process Mining: Discovery, Conformance and Enhancement of Business Processes

- Kap. 5.2: S.132, Fig. 5.6

## 5. Beispiel $L_5$ : Event-Log und Relationen



$$L_5 = [\langle a, b, e, f \rangle^2, \langle a, b, e, c, d, b, f \rangle^3, \langle a, b, c, e, d, b, f \rangle^2, \langle a, b, c, d, e, b, f \rangle^4, \langle a, e, b, c, d, b, f \rangle^3]$$

	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
<i>a</i>	#	→	#	#	→	#
<i>b</i>	←	#	→	←		→
<i>c</i>	#	←	#	→		#
<i>d</i>	#	→	←	#		#
<i>e</i>	←				#	→
<i>f</i>	#	←	#	#	←	#

78

### Literatur:

Wil van der Aalst: Process Mining: Discovery, Conformance and Enhancement of Business Processes

- Kap. 5.2: S.135, Tabelle 5.5

## 5. Beispiel $L_5$ :

### Durchführung $\alpha$ -Algorithmus

$$T_L = \{a, b, c, d, e, f\}$$

$$T_I = \{a\}$$

$$T_I = \{f\}$$

$$X_L = \{(\{a\}, \{b\}), (\{a\}, \{e\}), (\{b\}, \{c\}), (\{b\}, \{f\}), (\{c\}, \{d\}), \\ (\{d\}, \{b\}), (\{e\}, \{f\}), (\{a, d\}, \{b\}), (\{b\}, \{c, f\})\}$$

$$Y_L = \{(\{a\}, \{e\}), (\{c\}, \{d\}), (\{e\}, \{f\}), (\{a, d\}, \{b\}), (\{b\}, \{c, f\})\}$$

$$P_L = \{p(\{a\}, \{e\}), p(\{c\}, \{d\}), p(\{e\}, \{f\}), p(\{a, d\}, \{b\}), p(\{b\}, \{c, f\}), i_L, o_L\}$$

$$F_L = \{(a, p(\{a\}, \{e\})), (p(\{a\}, \{e\}), e), (c, p(\{c\}, \{d\})), (p(\{c\}, \{d\}), d), \\ (e, p(\{e\}, \{f\})), (p(\{e\}, \{f\}), f), (a, p(\{a, d\}, \{b\})), (d, p(\{a, d\}, \{b\})), \\ (p(\{a, d\}, \{b\}), b), (b, p(\{b\}, \{c, f\})), (p(\{b\}, \{c, f\}), c), (p(\{b\}, \{c, f\}), f), \\ (i_L, a), (f, o_L)\}$$

$$\alpha(L) = (P_L, T_L, F_L)$$

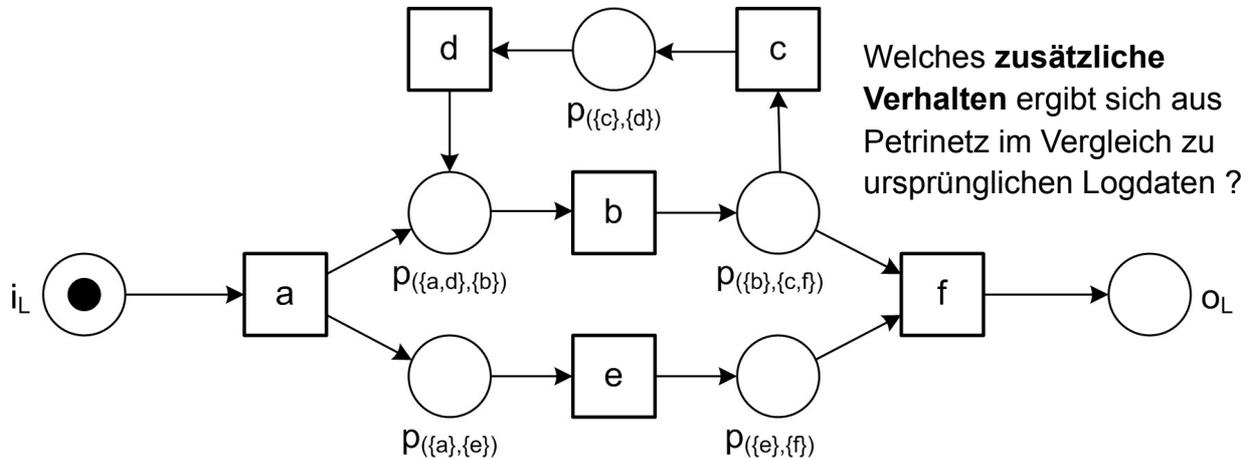
### Literatur:

Wil van der Aalst: Process Mining: Discovery, Conformance and Enhancement of Business Processes

- Kap. 5.2: S.135-136

## 5. Beispiel $L_5$ : Extrahiertes Petrinetz

$$L_5 = [\langle a, b, e, f \rangle^2, \langle a, b, e, c, d, b, f \rangle^3, \langle a, b, c, e, d, b, f \rangle^2, \langle a, b, c, d, e, b, f \rangle^4, \langle a, e, b, c, d, b, f \rangle^3]$$



80

### Literatur:

Wil van der Aalst: Process Mining: Discovery, Conformance and Enhancement of Business Processes

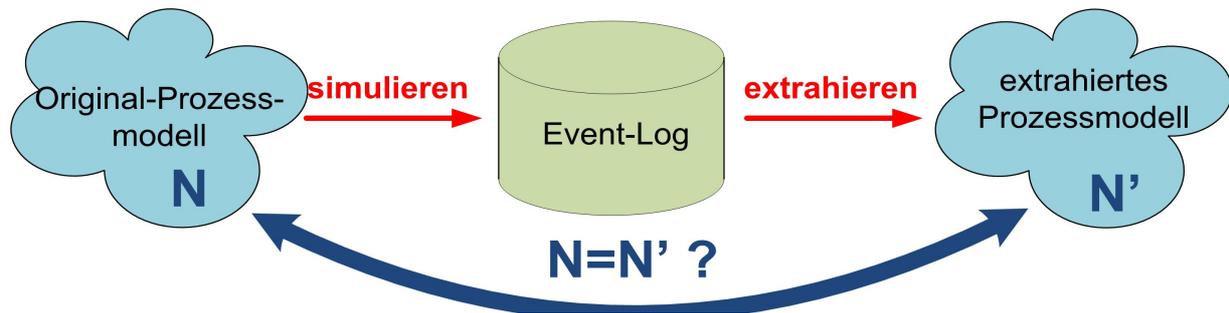
- Kap. 5.2: S.135-136, Fig. 5.8

# Überblick

## 2.4 Prozessextraktion



- Einführung und Beispiel
- **$\alpha$ -Algorithmus**
  - Idee und Vorbereitungen
  - Formalisierung
  - Beispiel
  - Weitere Beispiele
  - **Einschränkungen**
- Allgemeine Herausforderungen beim Process-Mining



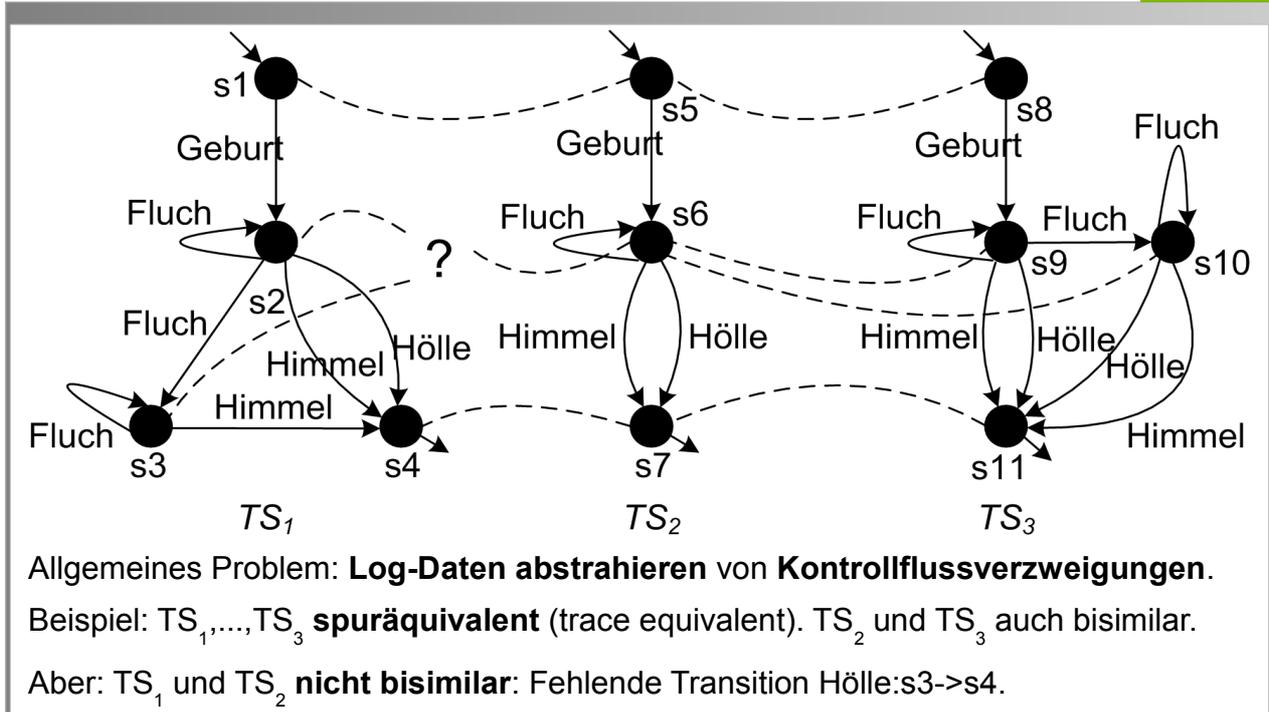
Re-Extraktionsproblem:

Entdeckte Modell  $N'$  äquivalent zum Original-Modell  $N$ ?

## Literatur:

Wil van der Aalst: Process Mining: Discovery, Conformance and Enhancement of Business Processes

- Kap. 5.3: S.141, Fig. 5.17



Allgemeines Problem: **Log-Daten abstrahieren** von **Kontrollflussverzweigungen**.

Beispiel:  $TS_1, \dots, TS_3$  **spuräquivalent** (trace equivalent).  $TS_2$  und  $TS_3$  auch bisimilar.

Aber:  $TS_1$  und  $TS_2$  **nicht bisimilar**: Fehlende Transition  $Hölle:s_3 \rightarrow s_4$ .

## Literatur:

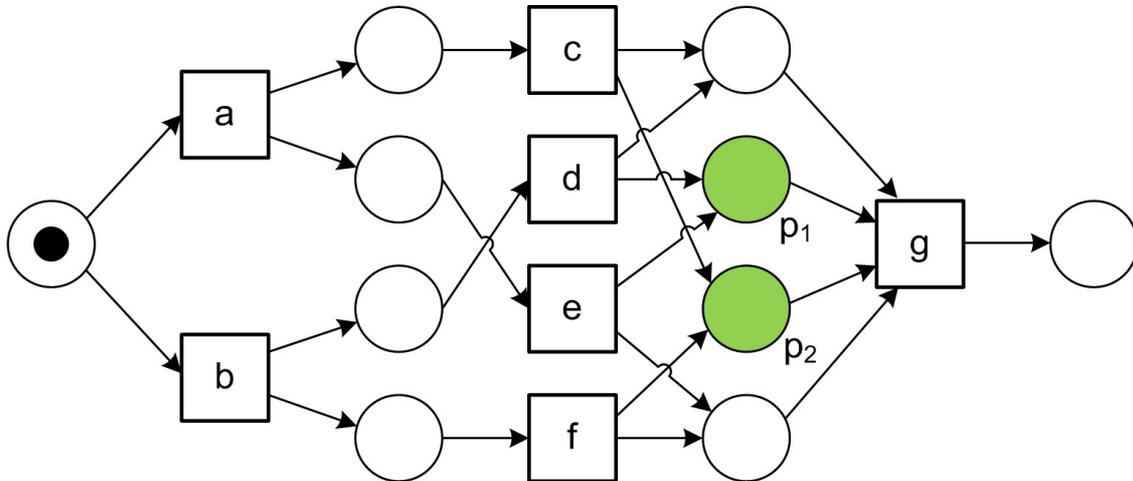
Wil van der Aalst: Process Mining: Discovery, Conformance and Enhancement of Business Processes

- Kap. 5.3: S.142, Fig. 5.18



- → i.A. Log-Daten von **mehr als einem Petrinetz** erzeugbar.
- Nur „**beobachtbare**“ Netzwerke werden erzeugt.
  - **Splits** und **Joins** nicht immer direkt beobachtbar.
  - Beobachtetes Verhalten korrekt wiedergegeben.

$$L_6 = [\langle a, c, e, g \rangle^2, \langle a, e, c, g \rangle^3, \langle b, d, f, g \rangle^2, \langle b, f, d, g \rangle^4]$$



Grüne Stellen **implizit**: Werden im Petrinetz nicht benötigt, aufgrund Symmetrie der Logdaten erzeugt.

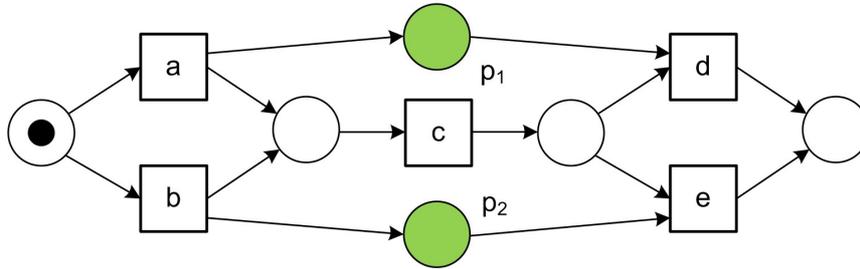
85

## Literatur:

Wil van der Aalst: Process Mining: Discovery, Conformance and Enhancement of Business Processes

- Kap. 5.2: S. 136-137, Fig. 5.9

$$L_9 = [\langle a, c, d \rangle^{45}, \langle b, c, e \rangle^{42}]$$



Grüne Stellen werden **nicht entdeckt** !  
Extrahiertes Petrinetz also dasselbe wie für:

$$L_4 = [\langle a, c, d \rangle^{45}, \langle b, c, d \rangle^{42}, \langle a, c, e \rangle^{38}, \langle b, c, e \rangle^{22}]$$

## Literatur:

Wil van der Aalst: Process Mining: Discovery, Conformance and Enhancement of Business Processes

- Kap. 5.2: S. 138-139, Fig. 5.14



- **Eindeutigkeit der Bezeichner** nach o.g. Annahme:  
Jeder Transitionsname taucht nur einmal im Petrinetz auf.
- **Nicht jede Folge** von Transitionen aus wohlgeformten Petrinetz  
**erzeugbar !**

NB: Ohne o.g. Annahme Petrinetz immer als XOR-Verknüpfung über  
Transitionen, die je eine Sequenz erzeugen, darstellbar.

### **Mehrfach auftretende Aktivitäten:**

- Schwierig;  $\alpha$ -Algorithmus konstruiert nur eine Transition pro Aktivität.

87

## **Literatur:**

Wil van der Aalst: Process Mining: Discovery,  
Conformance and Enhancement of Business  
Processes

- Kap. 5.2: S. 138-139



$$L_{10} = [\langle a, a \rangle^{55}]$$

Wie durch Petrinetz darstellen, das jeden Transitionsbezeichner jeweils **nur einmal** enthält ?

88

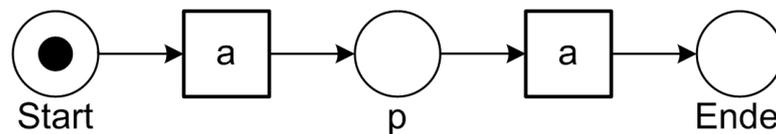
## Literatur:

Wil van der Aalst: Process Mining: Discovery, Conformance and Enhancement of Business Processes

- Kap. 5.4: S. 145

$$L_{10} = [\langle a, a \rangle^{55}]$$

Naheliegend (nicht mit eindeutigen Transitionsbezeichnungen):



Gibt **kein WF-net** mit eindeutigen Transitionsbezeichnungen, welches (exakt) dieses Verhalten darstellt.

## Literatur:

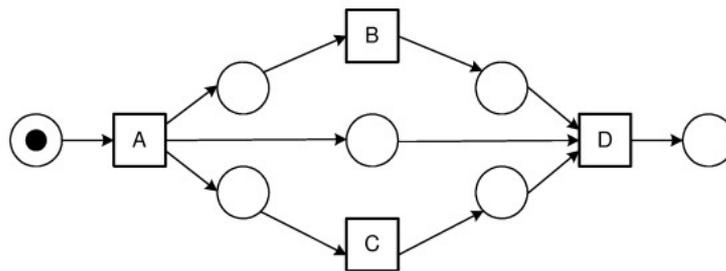
Wil van der Aalst: Process Mining: Discovery, Conformance and Enhancement of Business Processes

- Kap. 5.4: S. 145, Fig. 5.20

Abstraktion („unsichtbar machen“) von Transitionen:

**Wegabstrahieren** der Transition E in **Log-Daten** im obigen Beispiel führt zu **nicht** durch **Petrinetz** darstellbaren Log-Daten.

Z.B. erzeugt folgendes Petrinetz nur [ $\langle ABCD \rangle, \langle ACBD \rangle$ ] (nicht  $\langle AD \rangle$ , wie durch Abstrahieren von E aus  $\langle AED \rangle$  erhalten):



90

## Literatur:

Wil van der Aalst: Process Mining: Discovery, Conformance and Enhancement of Business Processes

- Kap. 5.4: S. 145

(a)

(b)

(c)

$L_{11} = [\langle a, b, c \rangle^{20}, \langle a, c \rangle^{30}]$

Welches Petrinetz extrahiert  $\alpha$ -Algorithmus aus  $L_{11}$  ?

Welche Logdaten erzeugt das Petrinetz bei Ausführung ?

## Literatur:

Wil van der Aalst: Process Mining: Discovery, Conformance and Enhancement of Business Processes

- Kap. 5.4: S. 145-146, Fig. 5.21

(a)

(b)

(c)

$L_{11} = [\langle a, b, c \rangle^{20}, \langle a, c \rangle^{30}]$

Gibt **kein WF-net** mit **eindeutigen** und **sichtbaren** Bezeichnungen, welches (genau) dieses Verhalten darstellt.

(a) **Unsichtbares** Ereignis  $\tau$

(b) Ereignis **a mehrfach**

(c) Extrahiert mit  **$\alpha$ -Algorithmus** aus  $L_{11}$ .  
Erzeugt nur  $[\langle a, b, c \rangle]$ .

## Literatur:

Wil van der Aalst: Process Mining: Discovery, Conformance and Enhancement of Business Processes

- Kap. 5.4: S. 145-146, Fig. 5.21



$$L_7 = [\langle a, c \rangle^2, \langle a, b, c \rangle^3, \langle a, b, b, c \rangle^2, \langle a, b, b, b, b, c \rangle^1]$$

Welches Petrinetz extrahiert  **$\alpha$ -Algorithmus** aus o.g. Logdaten ?

Welches **Petrinetz** generiert o.g. Logdaten ?

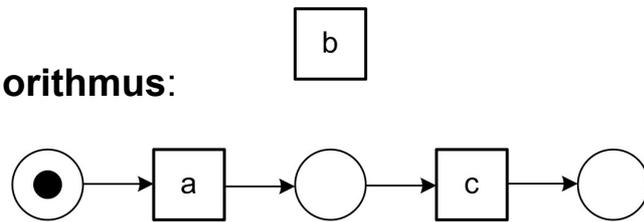
## Literatur:

Wil van der Aalst: Process Mining: Discovery,  
Conformance and Enhancement of Business  
Processes

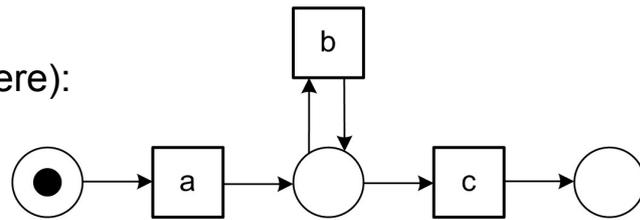
- Kap. 5.2: S. 137

$$L_7 = [\langle a, c \rangle^2, \langle a, b, c \rangle^3, \langle a, b, b, c \rangle^2, \langle a, b, b, b, b, c \rangle^1]$$

Petrinetz **extrahiert** von  $\alpha$ -Algorithmus:



Petrinetz, das tatsächlich o.g.  
**Logdaten generiert** (und weitere):



94

## Literatur:

Wil van der Aalst: Process Mining: Discovery, Conformance and Enhancement of Business Processes

- Kap. 5.2: S. 137, Fig. 5.10, 5.11



## Iterationen:

- Führen zu mehrfach auftretenden Aktivitäten.
- Müssen ggf. extra erkannt werden.
  - Mitunter sehr schwierig !
- Keine Beschränkung des Algorithmus; **Schleifen** durch **endliche Logdatenmengen** nur **unvollständig** charakterisierbar.

95

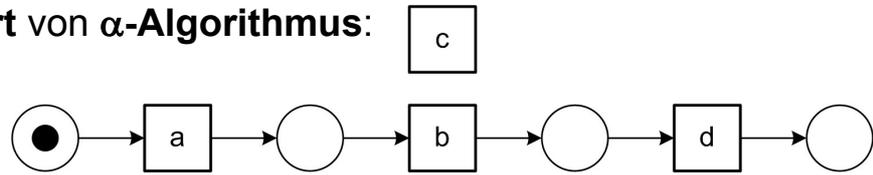
## Literatur:

Wil van der Aalst: Process Mining: Discovery, Conformance and Enhancement of Business Processes

- Kap. 5.2: S. 137

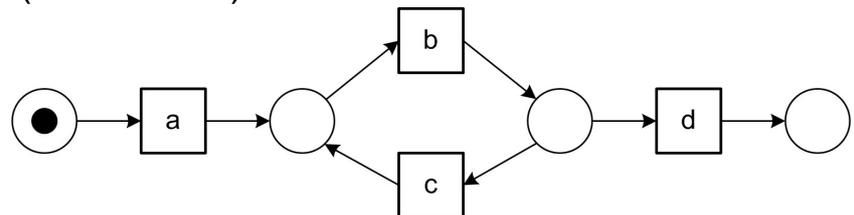
$$L_8 = [\langle a, b, d \rangle^3, \langle a, b, c, b, d \rangle^2, \langle a, b, c, b, c, b, d \rangle]$$

Petrinetz extrahiert von  $\alpha$ -Algorithmus:



Petrinetz, das tatsächlich o.g.

Logdaten generiert (und weitere):



96

## Literatur:

Wil van der Aalst: Process Mining: Discovery, Conformance and Enhancement of Business Processes

- Kap. 5.2: S. 137 letzter Abschnitt – S 138, Fig. 5.12, 5.13

# Überblick

## 2.4 Prozessextraktion



- Einführung und Beispiel
- $\alpha$ -Algorithmus
  - Idee und Vorbereitungen
  - Formalisierung
  - Beispiel
  - Weitere Beispiele
  - Einschränkungen
- **Allgemeine Herausforderungen beim Process-Mining**



Annahme passendes Prozessmodell zu finden: Event-Log enthält **repräsentative Verhaltensmenge**.

Zwei verwandte Phänomene:

- **Rauschen**: Event-Log enthält seltenes und unregelmäßiges Verhalten. → **Nicht typisches Verhalten** des Prozesses.
  - Lösung z.B. durch Betrachtung der **Häufigkeit**.
- **Unvollständigkeit**: Event-Log enthält **zu wenig Events**, um alle Kontrollflüsse zu erfassen.
  - Siehe auch Teil 2.2 (cross-validation, precision, recall, etc.).

98

## Literatur:

Wil van der Aalst: Process Mining: Discovery, Conformance and Enhancement of Business Processes

- Kap. 5.4: Seite 147



Bsp. zu Relevanz von **Vollständigkeit**: Betrachte Prozess aus 10 Aktivitäten.

- Annahme: Parallel ausführbar.

Gegeben: **Log mit Informationen** über 10.000 Fälle.

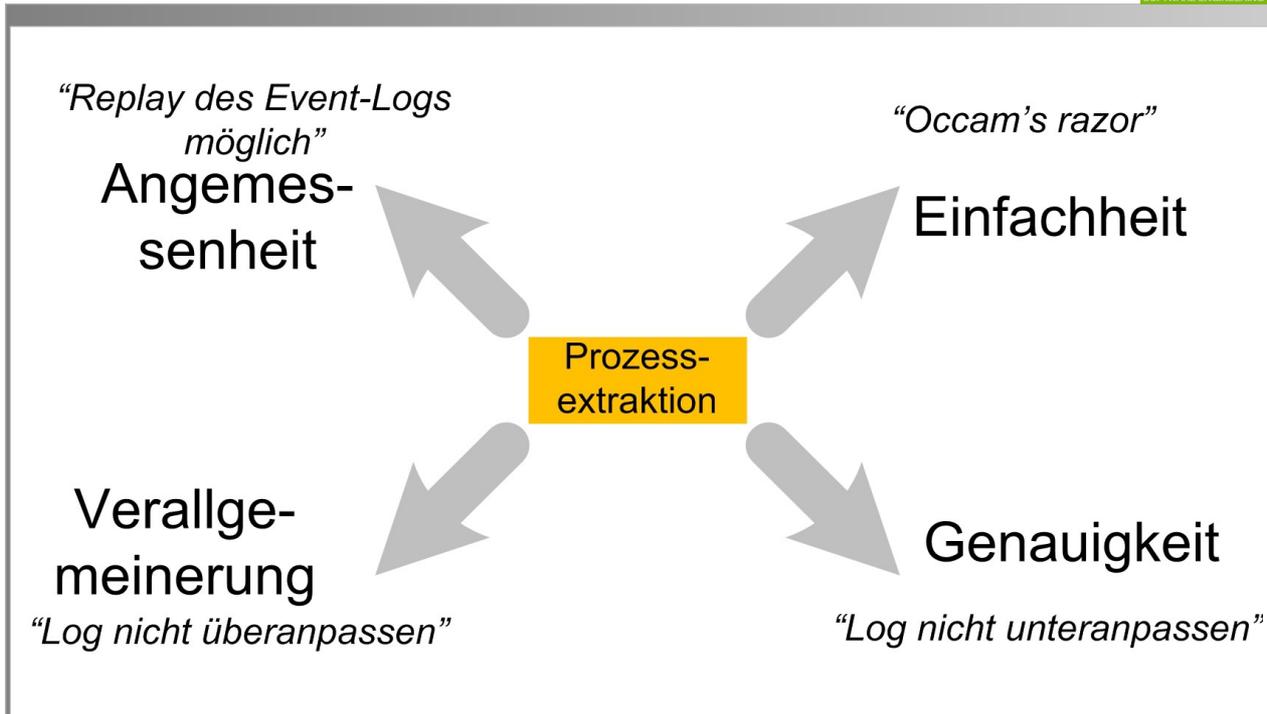
- Anzahl mögliche Kombinationen (10 nebenläufige Aktivitäten):  
 $10! = 3.628.800$ .
- Unmöglich jede Kombination im Log enthalten:
  - Im Log weniger Fälle (10.000) als mögliche Traces (3.628.800).
- Selbst bei 3.628.800 Fällen im Log:
  - **Alle Variationen vorhanden → Unwahrscheinlich**
  - Analogie: 365 Menschen, jeder hat am anderen Tag Geburtstag.  
→ Unwahrscheinlich  
 $W'keit \text{ bei } 365!/365^{365} \approx 1.454955 \times 10^{-157}$ .
  - Anzahl von Atomen im Universum ca.  $10^{79}$ .

99

## Literatur:

Wil van der Aalst: Process Mining: Discovery, Conformance and Enhancement of Business Processes

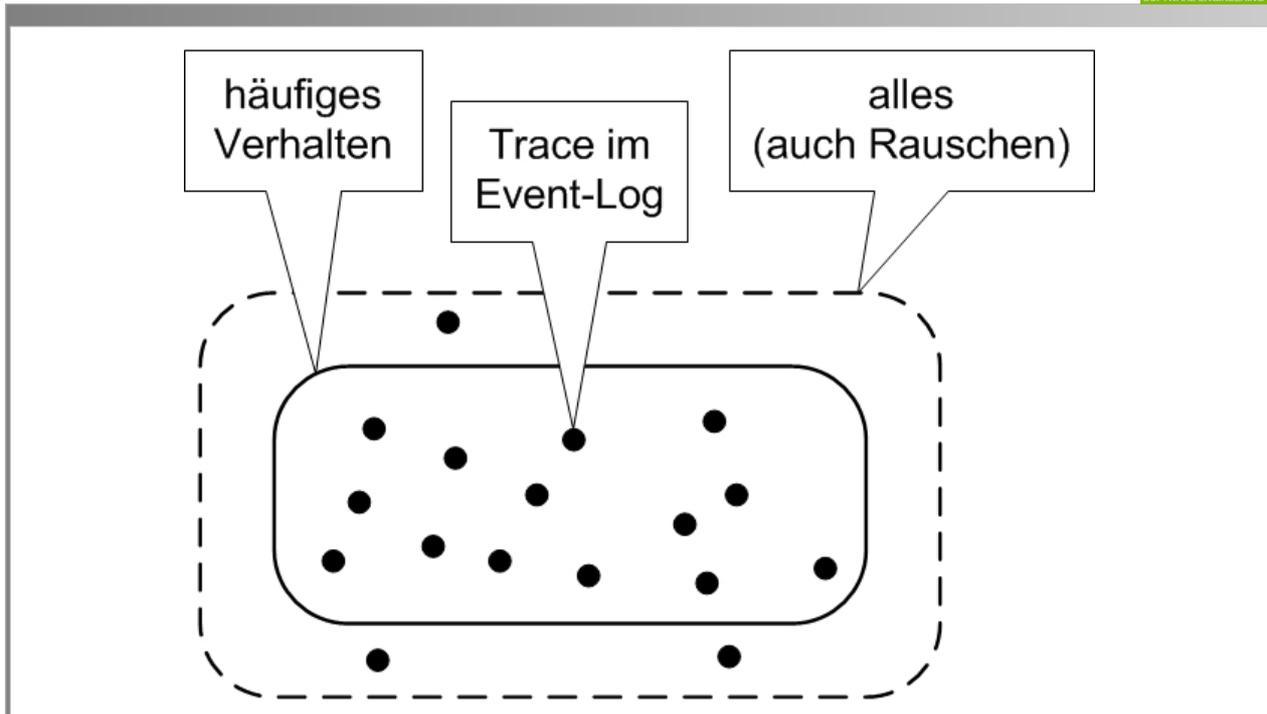
- Kap. 5.4: Seite 149



## Literatur:

Wil van der Aalst: Process Mining: Discovery, Conformance and Enhancement of Business Processes

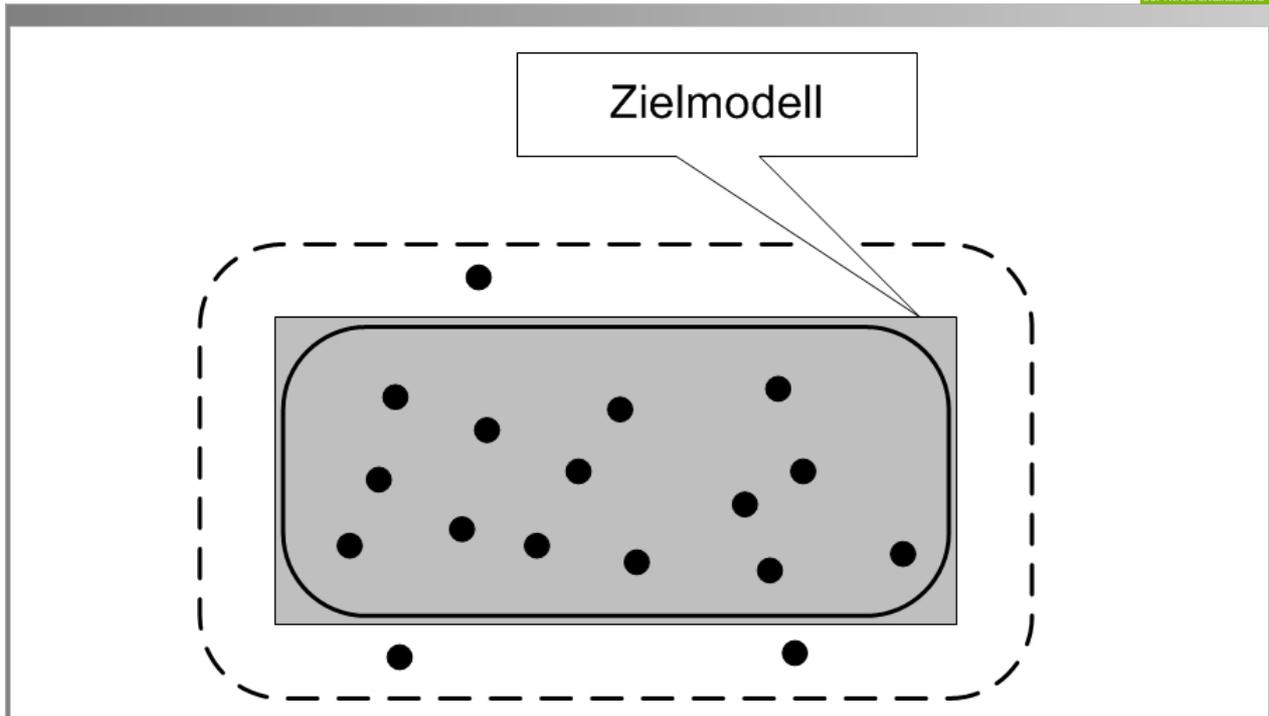
- Kap. 5.4: S. 151, Fig. 5.22



## Literatur:

Wil van der Aalst: Process Mining: Discovery, Conformance and Enhancement of Business Processes

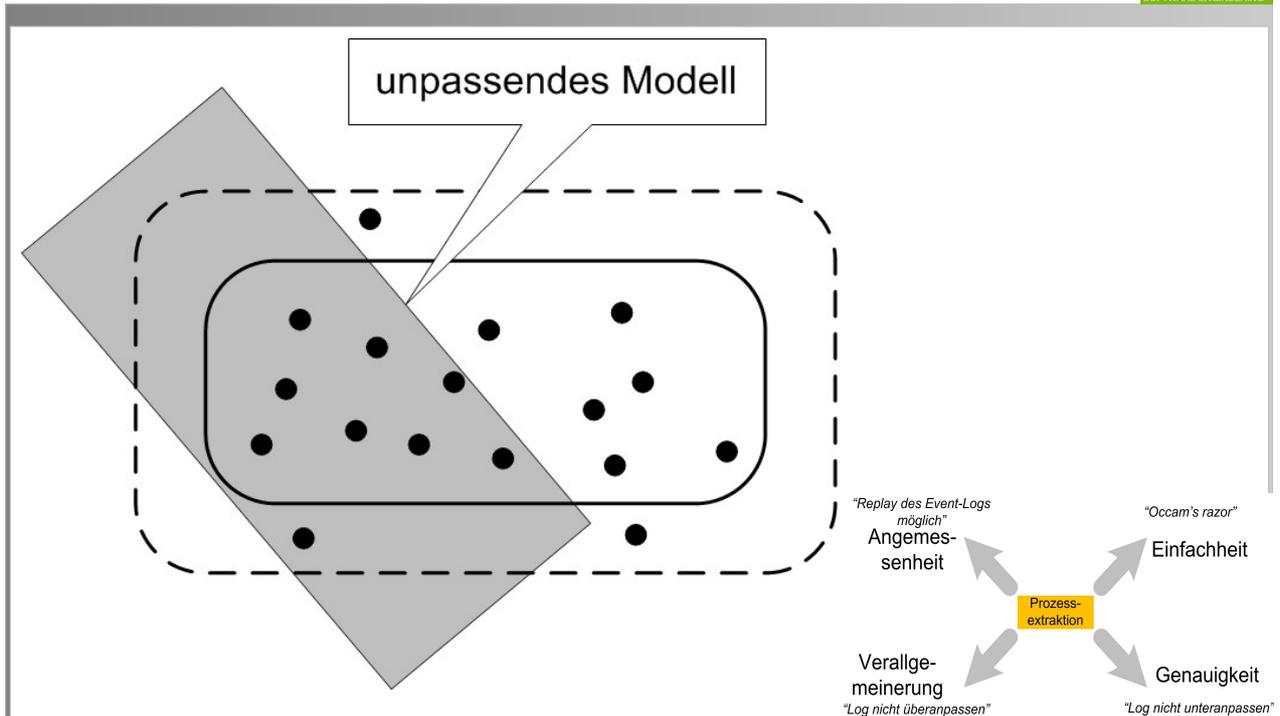
- Kap. 6.1: S. 158, Fig. 6.1



## Literatur:

Wil van der Aalst: Process Mining: Discovery, Conformance and Enhancement of Business Processes

- Kap. 6.1: S. 158, Fig. 6.1



## Literatur:

Wil van der Aalst: Process Mining: Discovery, Conformance and Enhancement of Business Processes

- Kap. 6.1: S. 158, Fig. 6.1

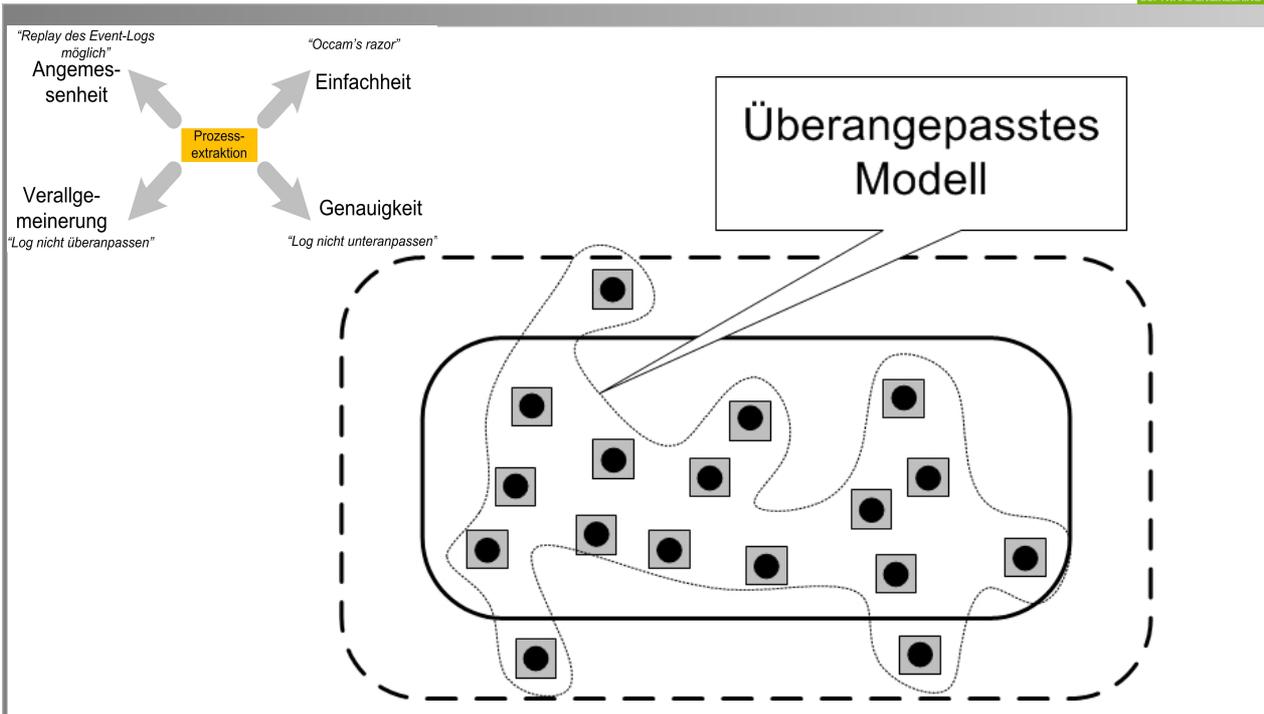


**Besondere Herausforderung:  
Balance zwischen  
Unteranpassung und  
Überanpassung**

### **Literatur:**

Wil van der Aalst: Process Mining: Discovery, Conformance and Enhancement of Business Processes

- Kap. 6.1: S. 159 ersten beiden Abschnitte

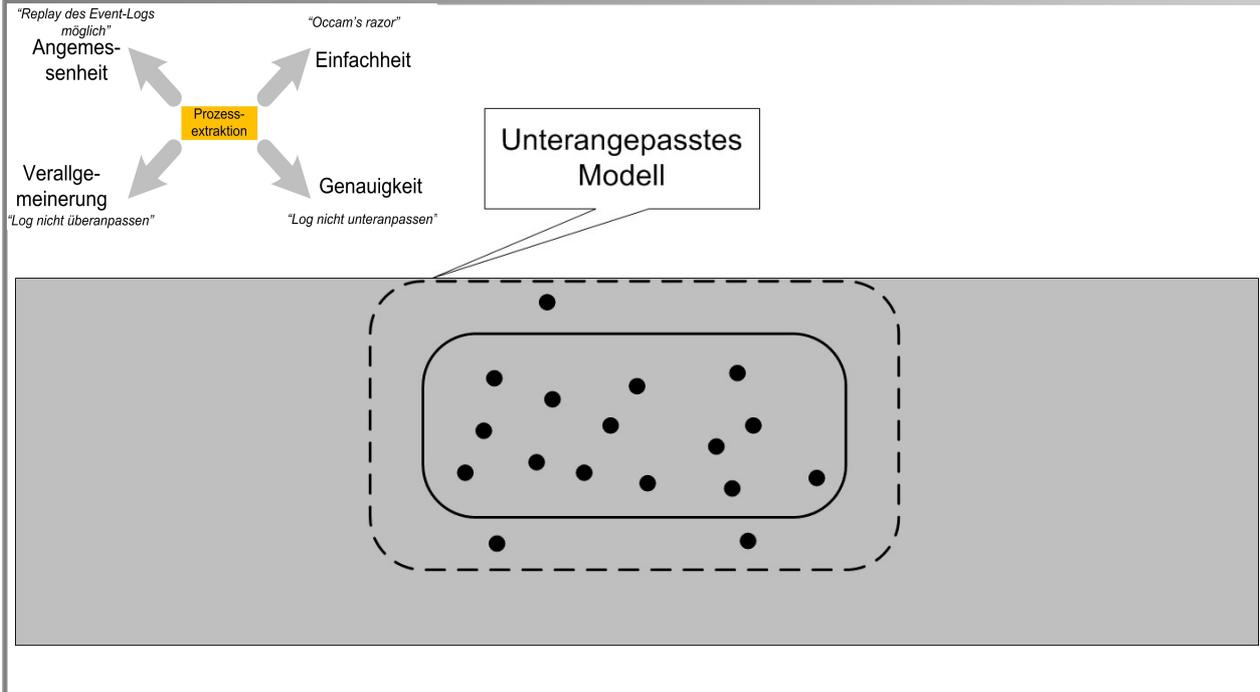


105

## Literatur:

Wil van der Aalst: Process Mining: Discovery, Conformance and Enhancement of Business Processes

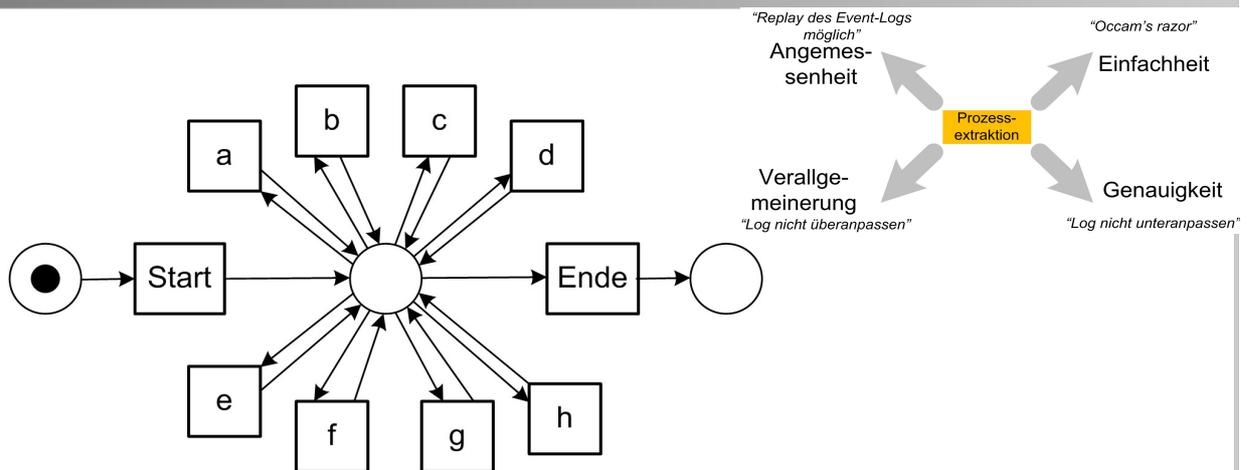
- Kap. 6.1: S. 158, Fig. 6.1



## Literatur:

Wil van der Aalst: Process Mining: Discovery, Conformance and Enhancement of Business Processes

- Kap. 6.1: S. 158, Fig. 6.1



"Replay des Event-Logs  
möglich"

Angemes-  
senheit

"Occam's razor"

Einfachheit

Prozess-  
extraktion

Verallge-  
meinerung

"Log nicht überanpassen"

Genauigkeit

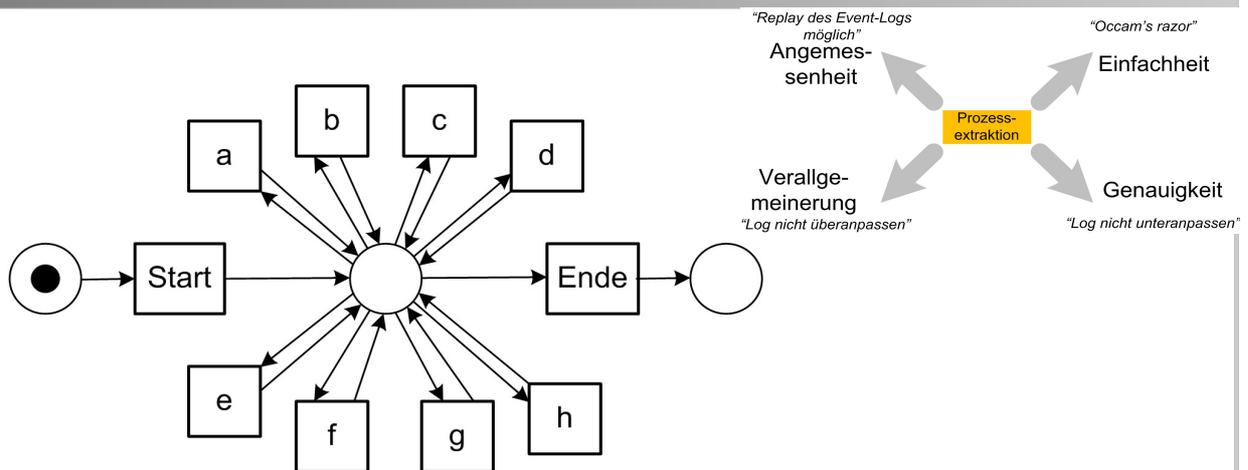
"Log nicht unteranpassen"

Welche **Qualitätskriterien** erfüllt dieses Modell in besonderer Weise ?

## Literatur:

Wil van der Aalst: Process Mining: Discovery, Conformance and Enhancement of Business Processes

- Kap. 5.4: S. 152, Fig. 5.23



Welche **Qualitätskriterien** erfüllt dieses Modell in besonderer Weise ?

Paradebeispiel für **unterangepasstes Modell**:

jedes Verhalten auf Basis der Ereignisse a, ..., h zulässig.

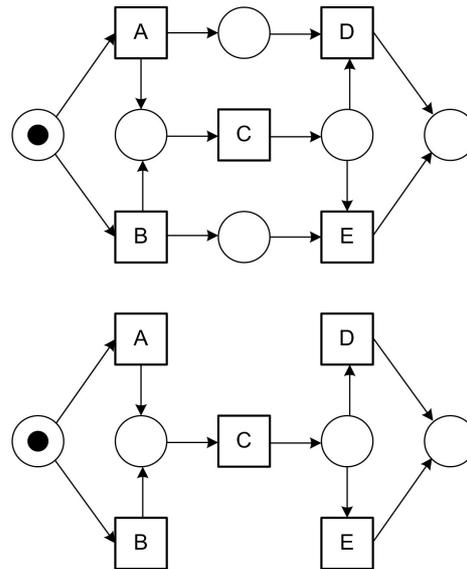
## Literatur:

Wil van der Aalst: Process Mining: Discovery, Conformance and Enhancement of Business Processes

- Kap. 5.4: S. 152, Fig. 5.23

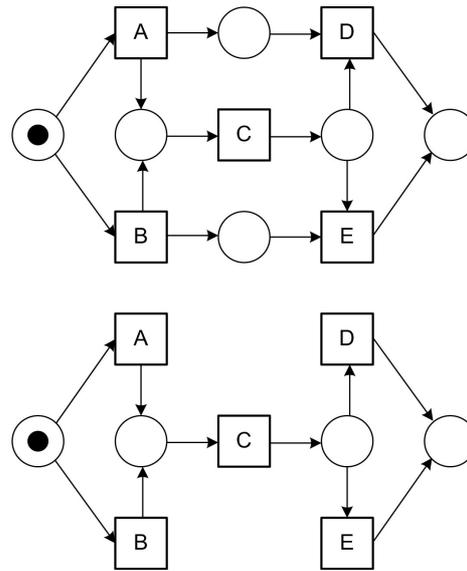
# Welches ist das beste Modell?

ACD	99
ACE	0
BCE	85
BCD	0



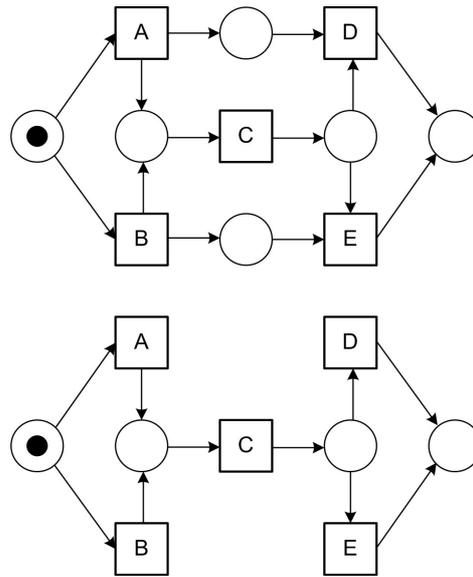
# Welches ist das beste Modell?

ACD	99
ACE	0
BCE	85
BCD	0



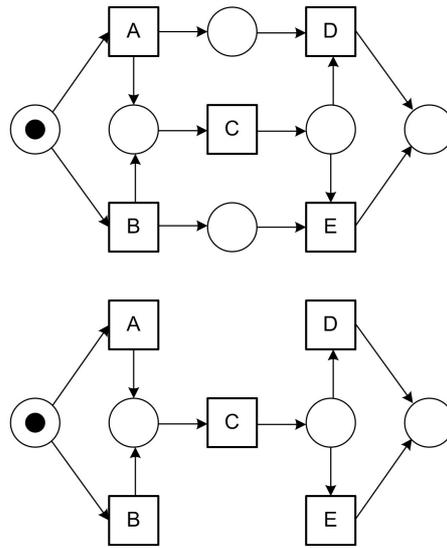
# Welches ist das beste Modell?

ACD	99
ACE	88
BCE	85
BCD	78



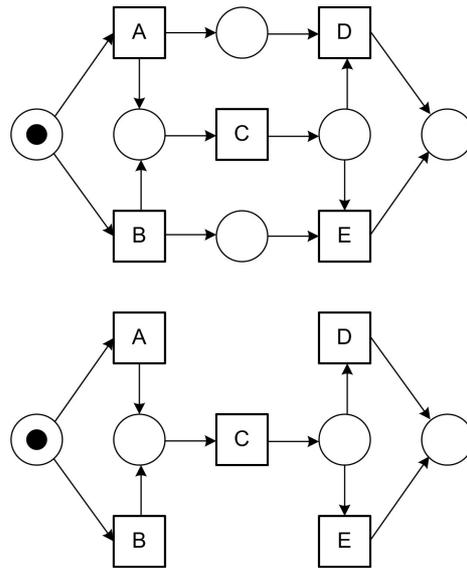
# Welches ist das beste Modell?

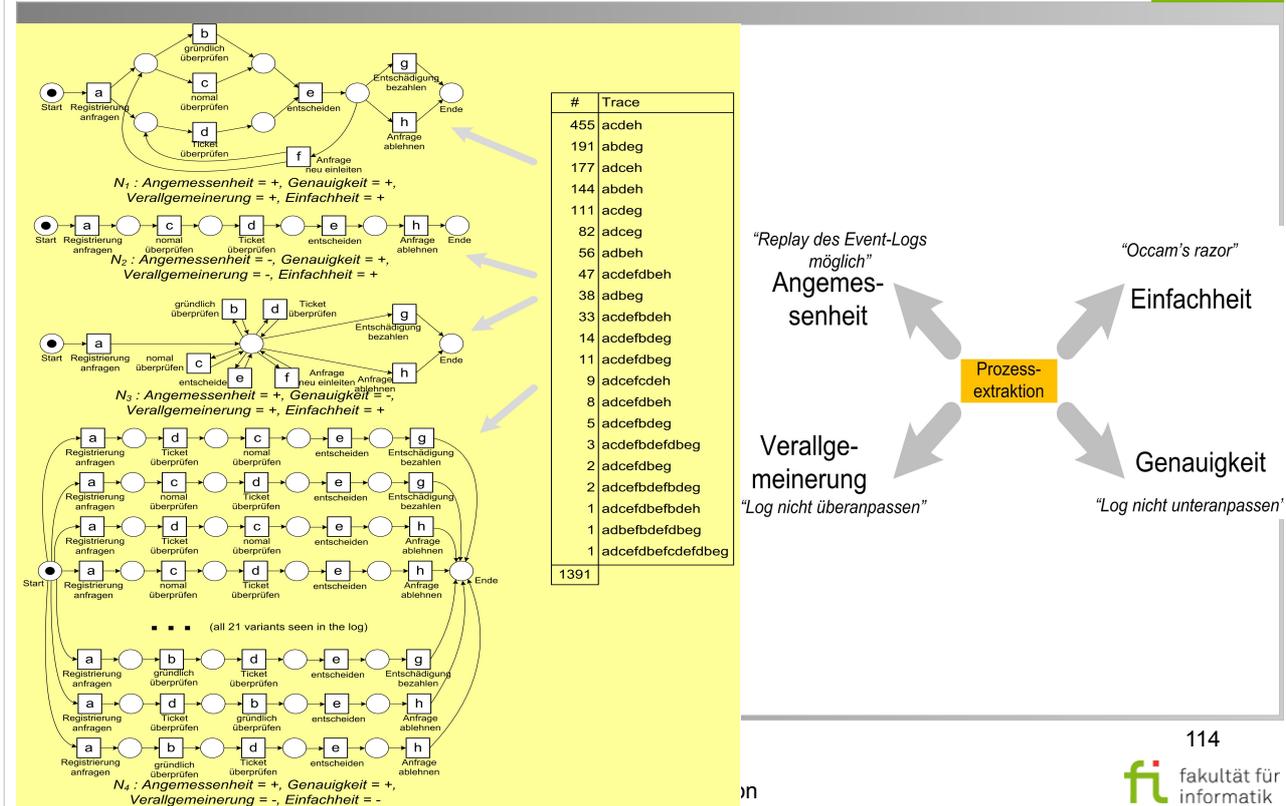
- ACD **99**
- ACE **88**
- BCE **85**
- BCD **78**



# Welches ist das beste Modell?

ACD	99
ACE	2
BCE	85
BCD	3





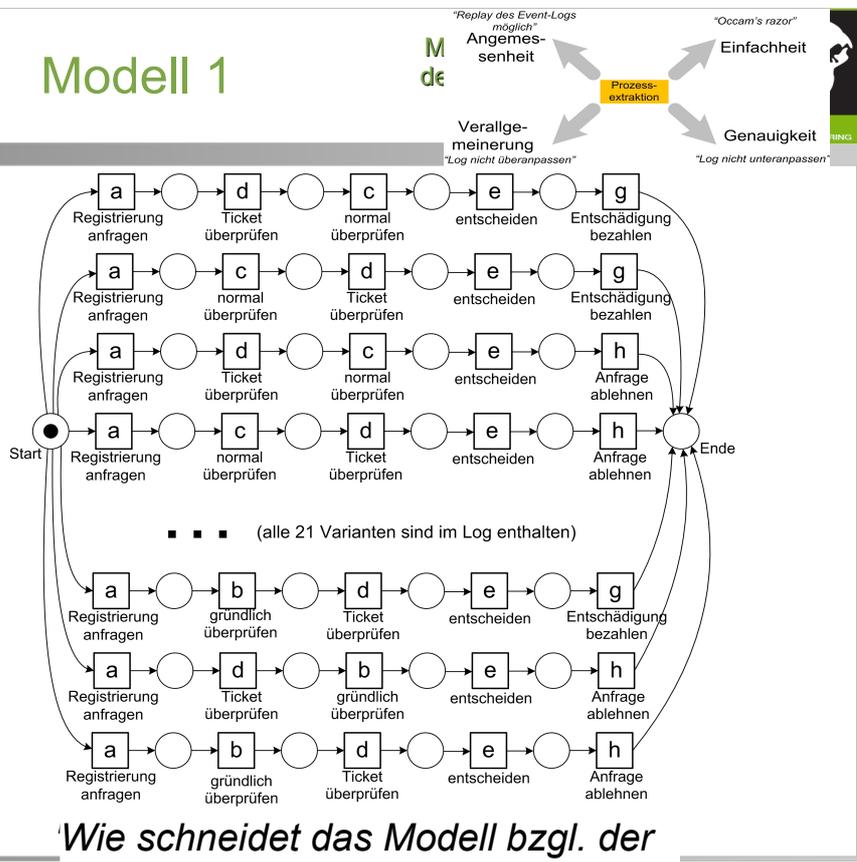
## Literatur:

Wil van der Aalst: Process Mining: Discovery, Conformance and Enhancement of Business Processes

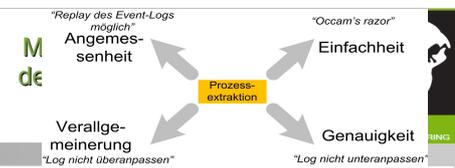
- Kap. 5.4: S. 154, Fig. 5.24

#	trace
455	acdeh
191	abdeg
177	adceh
144	abdeh
111	acdeg
82	adceg
56	adbeh
47	acdefdbeh
38	adbeg
33	acdefbdeh
14	acdefbdeg
11	acdefdbeg
9	adcefcdeh
8	adcefdbeh
5	adcefbdeg
3	acdefbdefdbeg
2	adcefbdeg
2	adcefbdefdbeg
1	adcefdbefbdeh
1	adbefbdefdbeg
1	adcefdbefcdefdbeg
1391	

# Modell 1



Wie schneidet das Modell bzgl. der 4 Qualitätskriterien ab ?



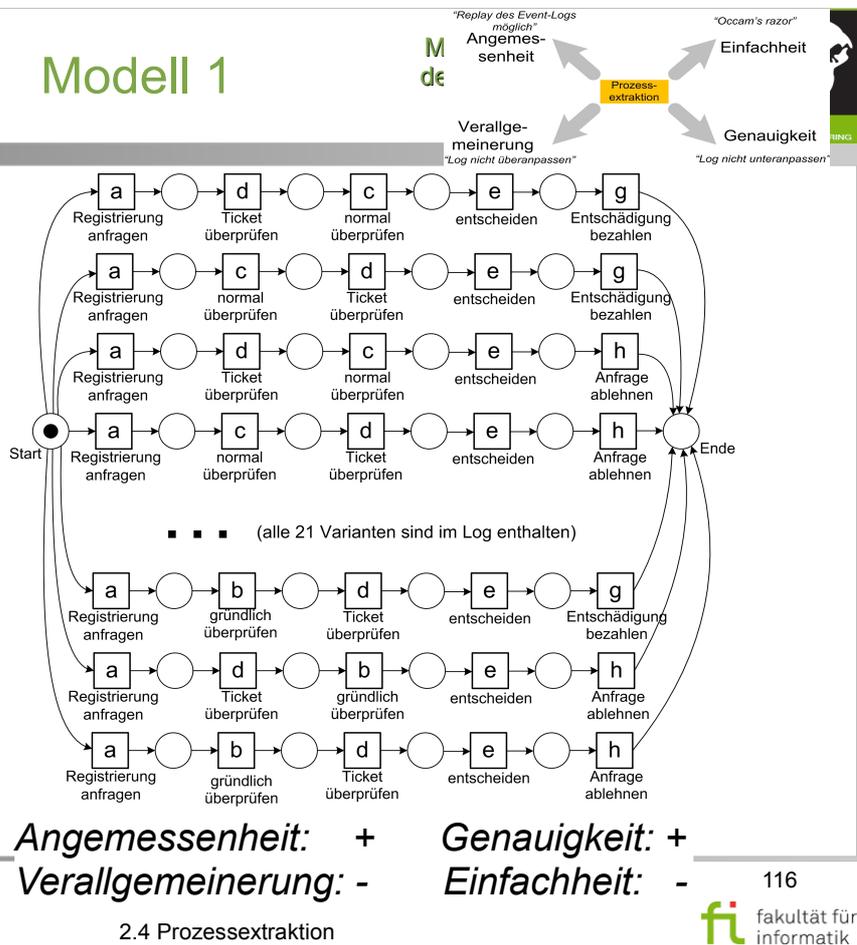
## Literatur:

Wil van der Aalst: Process Mining: Discovery, Conformance and Enhancement of Business Processes

- Kap. 5.4: S. 154, Fig. 5.24

#	trace
455	acdeh
191	abdeg
177	adceh
144	abdeh
111	acdeg
82	adceg
56	adbeh
47	acdefdbeh
38	adbeg
33	acdefbdeh
14	acdefbdeg
11	acdefdbeg
9	adcefcdeh
8	adcefdbeh
5	adcefbdeg
3	acdefbdefdbeg
2	adcefbdeg
2	adcefbdefdbeg
1	adcefdbefbdeh
1	adbefbdefdbeg
1	adcefdbefcdefdbeg
1391	

# Modell 1



## Literatur:

Wil van der Aalst: Process Mining: Discovery, Conformance and Enhancement of Business Processes

- Kap. 5.4: S. 154, Fig. 5.24

#	trace
455	acdeh
191	abdeg
177	adceh
144	abdeh
111	acdeg
82	adceg
56	adbeh
47	acdefdbeh
38	adbeg
33	acdefdbeh
14	acdefbdeg
11	acdefdbeg
9	adcefcdeh
8	adcefdbeh
5	adcefbdeg
3	acdefbdefdbeg
2	adcefdbeg
2	adcefbdefdbeg
1	adcefdbefbdeh
1	adbefbdefdbeg
1	adcefdbefcdefdbeg
1391	

## Modell 2

*Wie schneidet das Modell bzgl. der 4 Qualitätskriterien ab ?*

"Replay des Event-Logs möglich"  
 Angemessenheit

"Occam's razor"  
 Einfachheit

Verallgemeinerung  
 "Log nicht überanpassen"

Genauigkeit  
 "Log nicht unteranpassen"

Prozess-extraktion

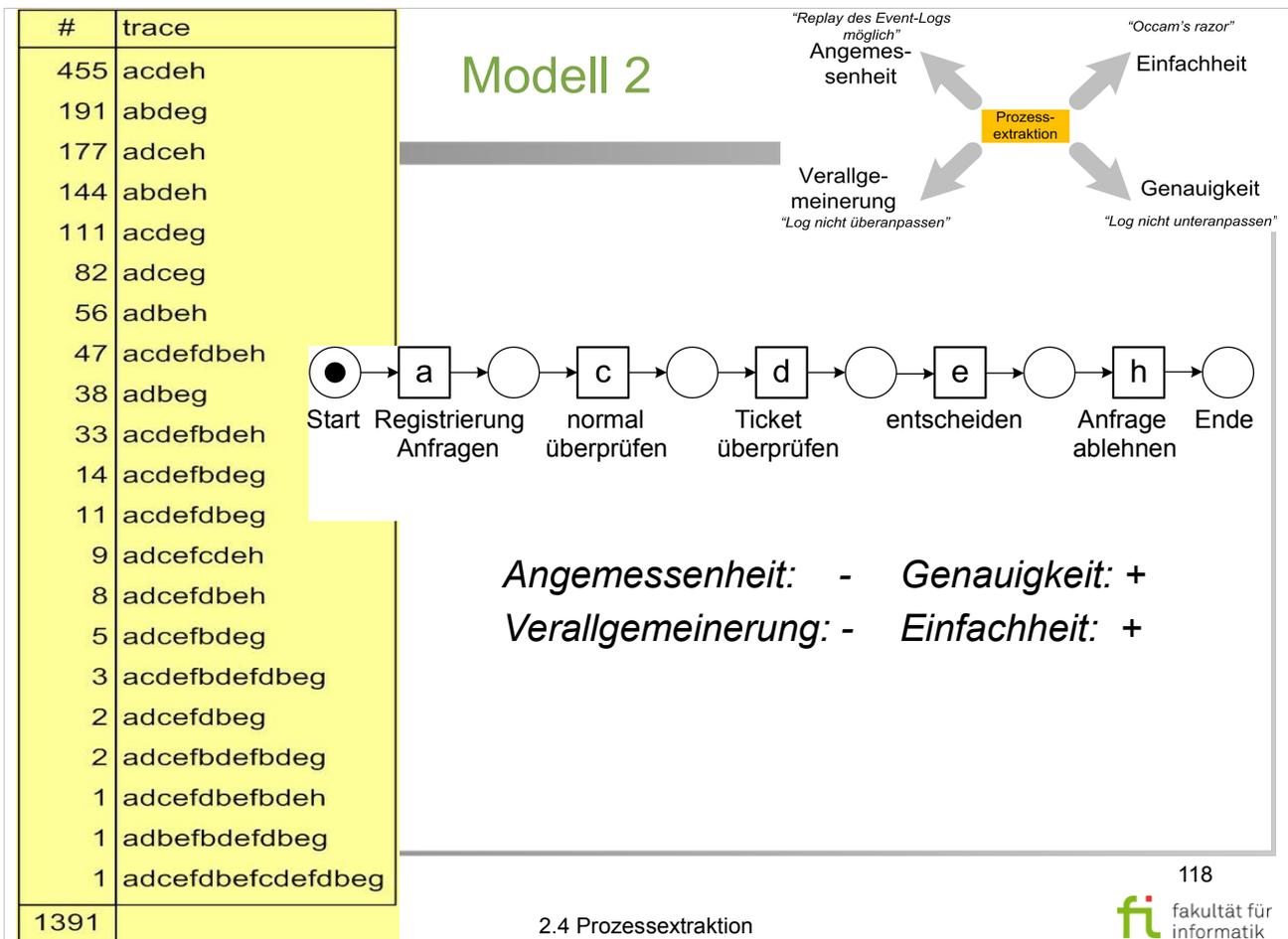
2.4 Prozessextraktion

117  
 fakultät für informatik

### Literatur:

Wil van der Aalst: Process Mining: Discovery, Conformance and Enhancement of Business Processes

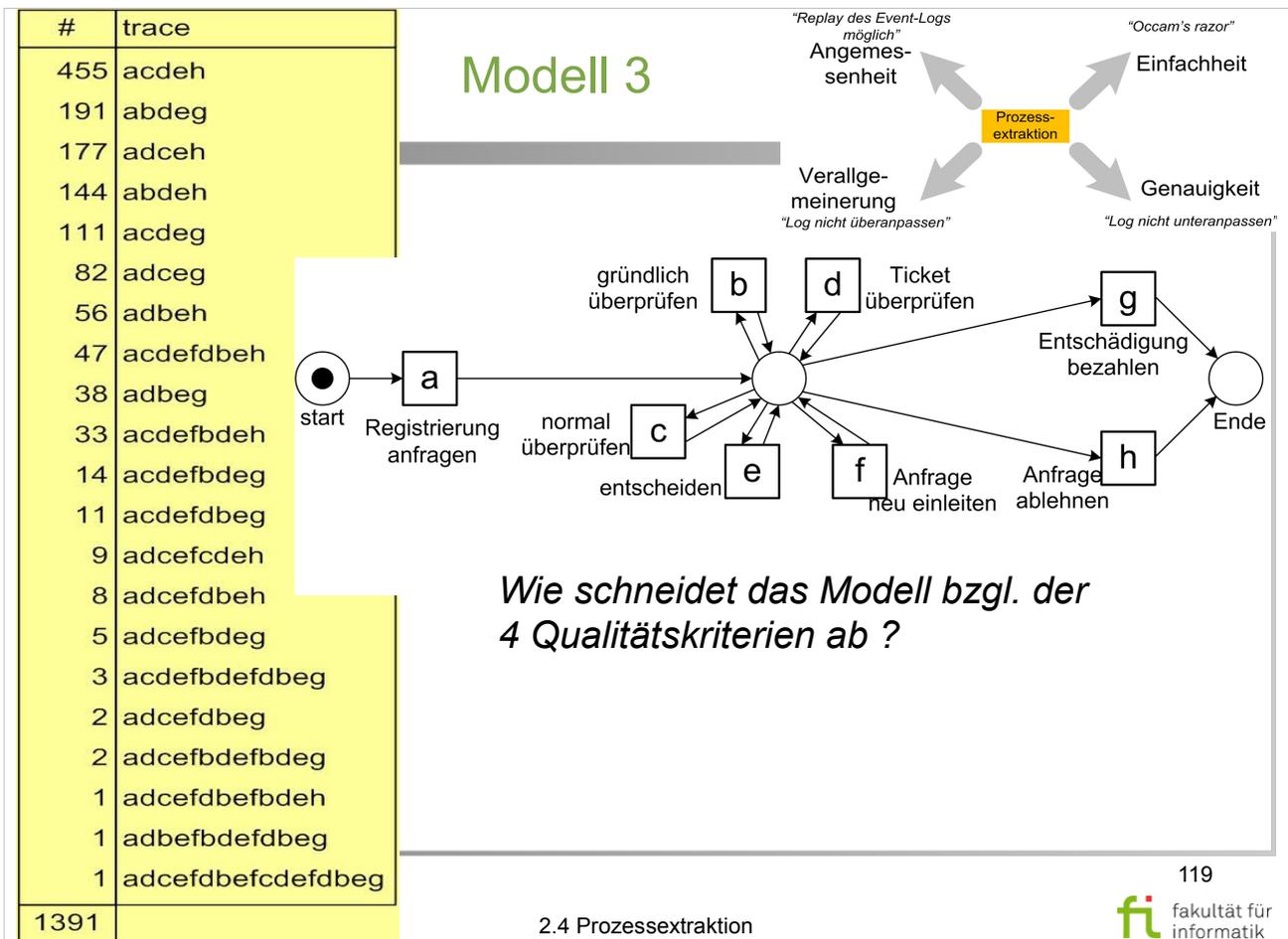
- Kap. 5.4: S. 154, Fig. 5.24



## Literatur:

Wil van der Aalst: Process Mining: Discovery, Conformance and Enhancement of Business Processes

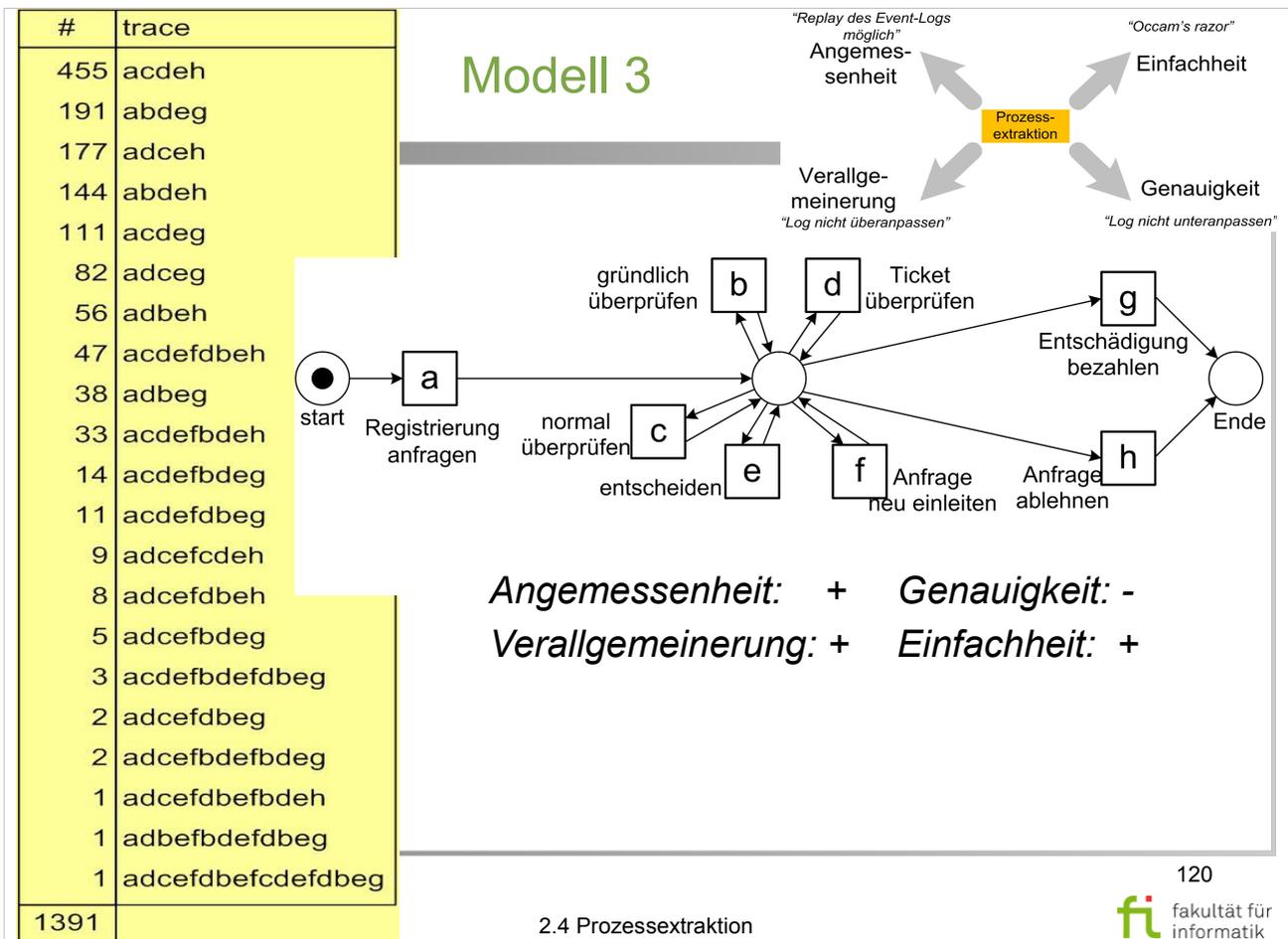
- Kap. 5.4: S. 154, Fig. 5.24



### Literatur:

Wil van der Aalst: Process Mining: Discovery, Conformance and Enhancement of Business Processes

- Kap. 5.4: S. 154, Fig. 5.24



## Literatur:

Wil van der Aalst: Process Mining: Discovery, Conformance and Enhancement of Business Processes

- Kap. 5.4: S. 154, Fig. 5.24

#	trace
455	acdeh
191	abdeg
177	adceh
144	abdeh
111	acdeg
82	adceg
56	adbeh
47	acdefdbeh
38	adbeg
33	acdefdbeh
14	acdefbdeg
11	acdefdbeg
9	adcefcdeh
8	adcefdbeh
5	adcefbdeg
3	acdefbdefdbeg
2	adcefbdeg
2	adcefbdefdbeg
1	adcefdbefdbeh
1	adbefbdefdbeg
1	adcefdbefcdefdbeg
1391	

## Modell 4

*Wie schneidet das Modell bzgl. der 4 Qualitätskriterien ab ?*

2.4 Prozessextraktion

121  
 fakultät für informatik

“Replay des Event-Logs möglich”  
Angemessenheit

“Occam’s razor”  
Einfachheit

Verallgemeinerung  
“Log nicht überanpassen”

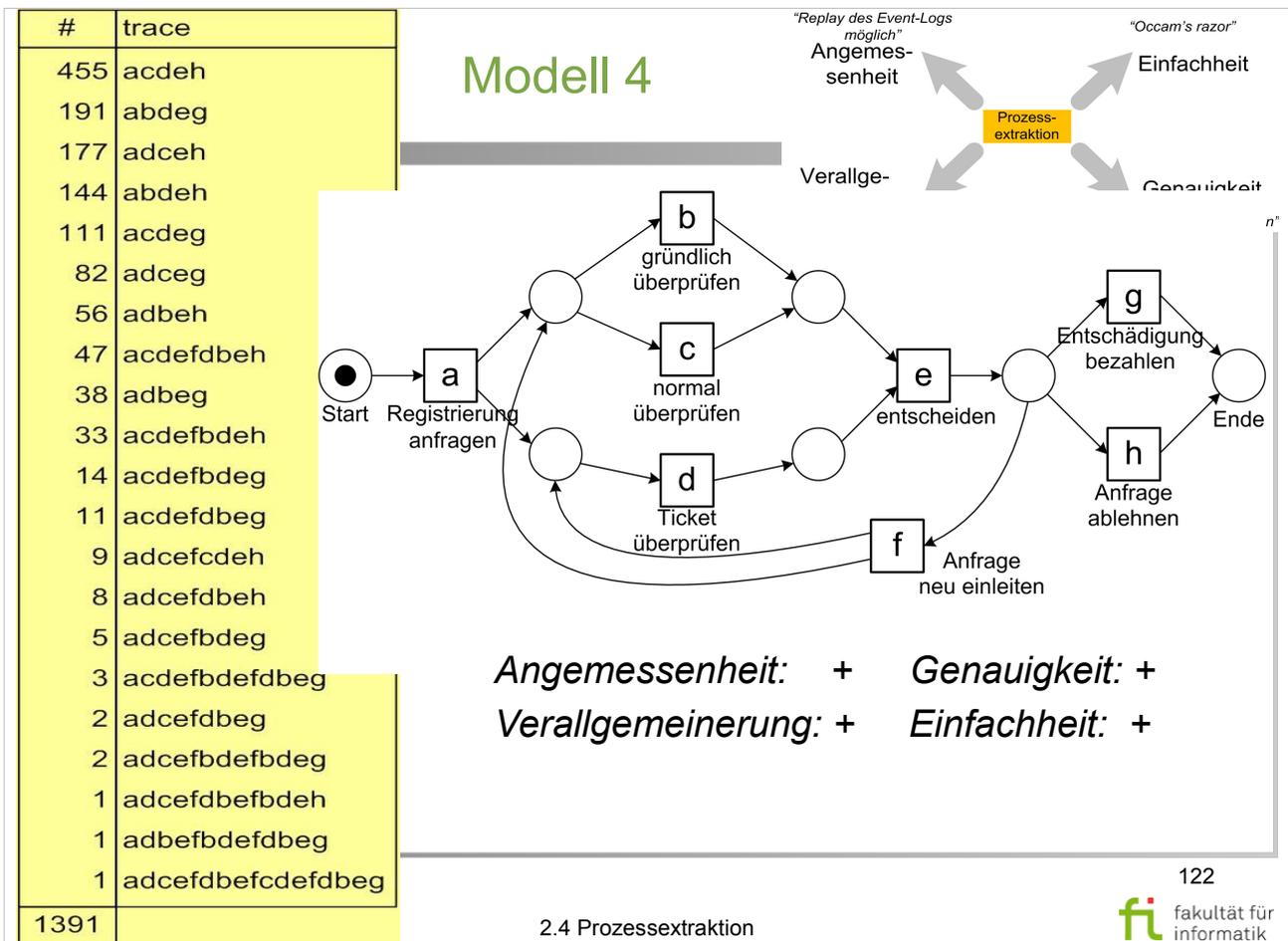
Genauigkeit  
“Log nicht unteranpassen”

Prozessextraktion

### Literatur:

Wil van der Aalst: Process Mining: Discovery, Conformance and Enhancement of Business Processes

- Kap. 5.4: S. 154, Fig. 5.24



### Literatur:

Wil van der Aalst: Process Mining: Discovery, Conformance and Enhancement of Business Processes

- Kap. 5.4: S. 154, Fig. 5.24



- Keine negativen Beispiele:
  - Logs zeigen, was geschah. **Nicht:** was nicht passieren kann.
- Durch Nebenläufigkeit, Schleifen, Verzweigungen:
  - Suchraum hat komplexe Struktur.
  - Log enthält **nur Teil** des möglichen Verhaltens.
- Kein Zusammenhang zwi. **Größe des Modells** und seinem **Verhalten:**
  - Kleines Modell kann mehr/weniger Verhalten generieren.
  - Klassische Analyse- und Evaluations-Methoden erwarten Monotonie-Eigenschaften.

## Literatur:

Wil van der Aalst: Process Mining: Discovery, Conformance and Enhancement of Business Processes

- Kap. 5.4: S. 154 – S. 155 erster Abschnitt



### In diesem Abschnitt:

- Einführung und Beispiel
- $\alpha$ -Algorithmus
  - Idee und Vorbereitungen
  - Formalisierung
  - Beispiel
  - Weitere Beispiele
  - Einschränkungen
- Allgemeine Herausforderungen beim Process-Mining

### Im nächsten Abschnitt:

- Konformanzanalyse (mit Hilfe von Replay).