Chapter 5
Process Discovery: An Introduction

prof.dr.ir. Wil van der Aalst
www.processmining.org
Overview

Part I: Preliminaries
- Chapter 2: Process Modeling and Analysis
- Chapter 3: Data Mining

Part II: From Event Logs to Process Models
- Chapter 4: Getting the Data
- Chapter 5: Process Discovery: An Introduction
- Chapter 6: Advanced Process Discovery Techniques

Part III: Beyond Process Discovery
- Chapter 7: Conformance Checking
- Chapter 8: Mining Additional Perspectives
- Chapter 9: Operational Support

Part IV: Putting Process Mining to Work
- Chapter 10: Tool Support
- Chapter 11: Analyzing "Lasagna Processes"
- Chapter 12: Analyzing "Spaghetti Processes"

Part V: Reflection
- Chapter 13: Cartography and Navigation
- Chapter 14: Epilogue
Process discovery

"world" business processes
people machines components organizations

models analyzes

software system

specifies configures implements analyzes

records events, e.g., messages, transactions, etc.

(process) model
discovery

conformance

enhancement

event logs
Process discovery = Play-In

Play-In

- event log
- process model

Play-Out

- process model
- event log

Replay

- event log
- process model

- extended model showing times, frequencies, etc.
- diagnostics
- predictions
- recommendations
Event log contains all possible traces of model and vice versa.

$L_1 = [\langle a, b, c, d \rangle^3, \langle a, c, b, d \rangle^2, \langle a, e, d \rangle]$
Another example

Generalization: event log contains only subset of all possible traces of model.

$L_2 = [\langle a, b, c, d \rangle^3, \langle a, c, b, d \rangle^4, \langle a, b, c, e, f, b, c, d \rangle^2, \langle a, b, c, e, f, c, b, d \rangle,\langle a, c, b, e, f, b, c, d \rangle^2, \langle a, c, b, e, f, b, c, e, f, c, b, d \rangle]$
Notation is less relevant (e.g. BPMN)

$L_1 = \left[ \langle a, b, c, d \rangle^3, \langle a, c, b, d \rangle^2, \langle a, e, d \rangle \right]$
Another BPMN example

\[
L_2 = \left[ \langle a, b, c, d \rangle^3, \langle a, c, b, d \rangle^4, \langle a, b, c, e, f, b, c, d \rangle^2, \langle a, b, c, e, f, c, b, d \rangle, \langle a, c, b, e, f, b, c, d \rangle^2, \langle a, c, b, e, f, b, c, e, f, c, b, d \rangle \right]
\]
In general, there is a trade-off between the following four quality criteria:

1. **Fitness**: the discovered model should allow for the behavior seen in the event log.

2. **Precision (avoid underfitting)**: the discovered model should not allow for behavior completely unrelated to what was seen in the event log.

3. **Generalization (avoid overfitting)**: the discovered model should generalize the example behavior seen in the event log.

4. **Simplicity**: the discovered model should be as simple as possible.
Process Discovery: example of algorithm
>,→,||,# relations

$L_1 = [\langle a, b, c, d \rangle^3, \langle a, c, b, d \rangle^2, \langle a, e, d \rangle]$
Basic Idea Used by $\alpha$ Algorithm (1)

(a) sequence pattern: $a \rightarrow b$
Basic Idea Used by Α Algorithm (2)

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

(c) XOR-join pattern: \( a \rightarrow c, b \rightarrow c, \) and \( a \# b \)
Basic Idea Used by Α Algorithm (3)

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

(e) AND-join pattern: 
\[ a \rightarrow c, \ b \rightarrow c, \ and \ a \parallel b \]
Example Revisited

\[ L_1 = [\langle a, b, c, d \rangle^3, \langle a, c, b, d \rangle^2, \langle a, e, d \rangle] \]

Result produced by \( \alpha \) algorithm
Footprint of $L_1$

$L_1 = [\langle a, b, c, d \rangle^3, \langle a, c, b, d \rangle^2, \langle a, e, d \rangle]$
Footprint of $L_2$

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

\[
\begin{array}{ccccccc}
 & a & b & c & d & e & f \\
\hline
 a & \# & \rightarrow & \rightarrow & \# & \# & \# \\
b & \leftarrow & \# & \| & \rightarrow & \rightarrow & \leftarrow \\
c & \leftarrow & \| & \# & \rightarrow & \rightarrow & \leftarrow \\
d & \# & \leftarrow & \leftarrow & \# & \# & \# \\
e & \# & \leftarrow & \leftarrow & \# & \# & \rightarrow \\
f & \# & \rightarrow & \rightarrow & \# & \leftarrow & \# \\
\end{array}
\]
Simple patterns

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

(b) XOR-split pattern: 
\( a \rightarrow b, a \rightarrow c, \text{ and } b \# c \)

(c) XOR-join pattern: 
\( a \rightarrow c, b \rightarrow c, \text{ and } a \# b \)

(d) AND-split pattern: 
\( a \rightarrow b, a \rightarrow c, \text{ and } b \| c \)

(e) AND-join pattern: 
\( a \rightarrow c, b \rightarrow c, \text{ and } a \| b \)
Algorithm

Let $L$ be an event log over $T$. $\alpha(L)$ is defined as follows.

1. $T_L = \{ t \in T \mid \exists \sigma \in L \ t \in \sigma \}$,
2. $T_I = \{ t \in T \mid \exists \sigma \in L \ t = \text{first}(\sigma) \}$,
3. $T_O = \{ t \in T \mid \exists \sigma \in L \ t = \text{last}(\sigma) \}$,
4. $X_L = \{ (A,B) \mid A \subseteq T_L \land A \neq \emptyset \land B \subseteq T_L \land B \neq \emptyset \land \forall a \in A \forall b \in B \ a \rightarrow_L b \land \forall a_1,a_2 \in A \ a_1 \#_L a_2 \land \forall b_1,b_2 \in B \ b_1 \#_L b_2 \}$,
5. $Y_L = \{ (A,B) \in X_L \mid \forall (A',B') \in X_L \ A \subseteq A' \land B \subseteq B' \Rightarrow (A,B) = (A',B') \}$,
6. $P_L = \{ p_{(A,B)} \mid (A,B) \in Y_L \} \cup \{ i_L, o_L \}$,
7. $F_L = \{ (a,p_{(A,B)}) \mid (A,B) \in Y_L \land a \in A \} \cup \{ (p_{(A,B)},b) \mid (A,B) \in Y_L \land b \in B \} \cup \{ (i_L,t) \mid t \in T_I \} \cup \{ (t,o_L) \mid t \in T_O \}$, and
8. $\alpha(L) = (P_L, T_L, F_L)$. 
Key idea: find places

4. $X_L = \{ (A, B) \mid A \subseteq T_L \land A \neq \emptyset \land B \subseteq T_L \land B \neq \emptyset \land \\
\forall a \in A \forall b \in B \ a \rightarrow_L b \land \forall a_1,a_2 \in A \, a_1 \#_L a_2 \land \forall b_1,b_2 \in B \, b_1 \#_L b_2 \}$,

5. $Y_L = \{ (A, B) \in X_L \mid \forall (A',B') \in X_L \ A \subseteq A' \land B \subseteq B' \Rightarrow (A,B) = (A',B') \}$,
### Places as footprints

\[ \text{A} = \{a_1, a_2, \ldots, a_m\} \quad \text{B} = \{b_1, b_2, \ldots, b_n\} \]

\[ p(A,B) \]

<table>
<thead>
<tr>
<th></th>
<th>( a_1 )</th>
<th>( a_2 )</th>
<th>( \ldots )</th>
<th>( a_m )</th>
<th>( b_1 )</th>
<th>( b_2 )</th>
<th>( \ldots )</th>
<th>( b_n )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( a_1 )</td>
<td>( # )</td>
<td>( # )</td>
<td>( \ldots )</td>
<td>( # )</td>
<td>( \rightarrow )</td>
<td>( \rightarrow )</td>
<td>( \ldots )</td>
<td>( \rightarrow )</td>
</tr>
<tr>
<td>( a_2 )</td>
<td>( # )</td>
<td>( # )</td>
<td>( \ldots )</td>
<td>( # )</td>
<td>( \rightarrow )</td>
<td>( \rightarrow )</td>
<td>( \ldots )</td>
<td>( \rightarrow )</td>
</tr>
<tr>
<td>( \ldots )</td>
<td>( \ldots )</td>
<td>( \ldots )</td>
<td>( \ldots )</td>
<td>( \ldots )</td>
<td>( \ldots )</td>
<td>( \ldots )</td>
<td>( \ldots )</td>
<td>( \ldots )</td>
</tr>
<tr>
<td>( a_m )</td>
<td>( # )</td>
<td>( # )</td>
<td>( \ldots )</td>
<td>( # )</td>
<td>( \rightarrow )</td>
<td>( \rightarrow )</td>
<td>( \ldots )</td>
<td>( \rightarrow )</td>
</tr>
<tr>
<td>( b_1 )</td>
<td>( \leftarrow )</td>
<td>( \leftarrow )</td>
<td>( \ldots )</td>
<td>( \leftarrow )</td>
<td>( # )</td>
<td>( # )</td>
<td>( \ldots )</td>
<td>( # )</td>
</tr>
<tr>
<td>( b_2 )</td>
<td>( \leftarrow )</td>
<td>( \leftarrow )</td>
<td>( \ldots )</td>
<td>( \leftarrow )</td>
<td>( # )</td>
<td>( # )</td>
<td>( \ldots )</td>
<td>( # )</td>
</tr>
<tr>
<td>( \ldots )</td>
<td>( \ldots )</td>
<td>( \ldots )</td>
<td>( \ldots )</td>
<td>( \ldots )</td>
<td>( \ldots )</td>
<td>( \ldots )</td>
<td>( \ldots )</td>
<td>( \ldots )</td>
</tr>
<tr>
<td>( b_n )</td>
<td>( \leftarrow )</td>
<td>( \leftarrow )</td>
<td>( \ldots )</td>
<td>( \leftarrow )</td>
<td>( # )</td>
<td>( # )</td>
<td>( \ldots )</td>
<td>( # )</td>
</tr>
</tbody>
</table>
\[ L_1 = \langle a, b, c, d \rangle^3, \langle a, c, b, d \rangle^2, \langle a, e, d \rangle \]

<table>
<thead>
<tr>
<th></th>
<th>(a)</th>
<th>(b)</th>
<th>(c)</th>
<th>(d)</th>
<th>(e)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a)</td>
<td>#_{L_1}</td>
<td>\rightarrow_{L_1}</td>
<td>\rightarrow_{L_1}</td>
<td>#_{L_1}</td>
<td>\rightarrow_{L_1}</td>
</tr>
<tr>
<td>(b)</td>
<td>\leftarrow_{L_1}</td>
<td>#_{L_1}</td>
<td>\parallel_{L_1}</td>
<td>\rightarrow_{L_1}</td>
<td>#_{L_1}</td>
</tr>
<tr>
<td>(c)</td>
<td>\leftarrow_{L_1}</td>
<td>\parallel_{L_1}</td>
<td>#_{L_1}</td>
<td>\rightarrow_{L_1}</td>
<td>#_{L_1}</td>
</tr>
<tr>
<td>(d)</td>
<td>#_{L_1}</td>
<td>\leftarrow_{L_1}</td>
<td>\leftarrow_{L_1}</td>
<td>#_{L_1}</td>
<td>\leftarrow_{L_1}</td>
</tr>
<tr>
<td>(e)</td>
<td>\leftarrow_{L_1}</td>
<td>#_{L_1}</td>
<td>#_{L_1}</td>
<td>\rightarrow_{L_1}</td>
<td>#_{L_1}</td>
</tr>
</tbody>
</table>

\[ X_{L_1} = \{ (\{a\}, \{b\}), (\{a\}, \{c\}), (\{a\}, \{e\}), (\{a\}, \{b, e\}), (\{a\}, \{c, e\}), (\{b\}, \{d\}), (\{c\}, \{d\}), (\{e\}, \{d\}), (\{b, e\}, \{d\}), (\{c, e\}, \{d\}) \} \]

\[ Y_{L_1} = \{ (\{a\}, \{b, e\}), (\{a\}, \{c, e\}), (\{b, e\}, \{d\}), (\{c, e\}, \{d\}) \} \]
Another event log $L_3$

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

<table>
<thead>
<tr>
<th></th>
<th>$a$</th>
<th>$b$</th>
<th>$c$</th>
<th>$d$</th>
<th>$e$</th>
<th>$f$</th>
<th>$g$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a$</td>
<td>#</td>
<td>→</td>
<td>#</td>
<td>#</td>
<td>#</td>
<td>#</td>
<td>#</td>
</tr>
<tr>
<td>$b$</td>
<td>←</td>
<td>#</td>
<td>→</td>
<td>→</td>
<td>#</td>
<td>←</td>
<td>#</td>
</tr>
<tr>
<td>$c$</td>
<td>#</td>
<td>←</td>
<td>#</td>
<td></td>
<td></td>
<td></td>
<td>→</td>
</tr>
<tr>
<td>$d$</td>
<td>#</td>
<td>←</td>
<td></td>
<td></td>
<td></td>
<td>#</td>
<td>→</td>
</tr>
<tr>
<td>$e$</td>
<td>#</td>
<td>#</td>
<td>←</td>
<td>←</td>
<td>#</td>
<td>→</td>
<td>→</td>
</tr>
<tr>
<td>$f$</td>
<td>#</td>
<td>→</td>
<td>#</td>
<td>#</td>
<td>←</td>
<td>#</td>
<td>#</td>
</tr>
<tr>
<td>$g$</td>
<td>#</td>
<td>#</td>
<td>#</td>
<td>#</td>
<td>←</td>
<td>#</td>
<td>#</td>
</tr>
</tbody>
</table>
Model for $L_3$

$L_3 = \langle a, b, c, d, e, f, b, d, c, e, g \rangle,
\langle a, b, d, c, e, g \rangle^2,
\langle a, b, c, d, e, f, b, c, d, e, f, b, d, c, e, g \rangle$
Another event log $L_4$

$L_4 = [\langle a, c, d \rangle^{45}, \langle b, c, d \rangle^{42}, \langle a, c, e \rangle^{38}, \langle b, c, e \rangle^{22}]$
Event log $L_5$

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

<table>
<thead>
<tr>
<th></th>
<th>$a$</th>
<th>$b$</th>
<th>$c$</th>
<th>$d$</th>
<th>$e$</th>
<th>$f$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a$</td>
<td>#</td>
<td>→</td>
<td>#</td>
<td>#</td>
<td>→</td>
<td>#</td>
</tr>
<tr>
<td>$b$</td>
<td>←</td>
<td>#</td>
<td>→</td>
<td>←</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$c$</td>
<td>#</td>
<td>←</td>
<td>#</td>
<td>→</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$d$</td>
<td>#</td>
<td>→</td>
<td>←</td>
<td>#</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$e$</td>
<td>←</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$f$</td>
<td>#</td>
<td>←</td>
<td>#</td>
<td>#</td>
<td>←</td>
<td>#</td>
</tr>
</tbody>
</table>
\[ T_L = \{a, b, c, d, e, f\} \]
\[ T_I = \{a\} \]
\[ T_I = \{f\} \]

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

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

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

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

\[ \alpha(L) = (P_L, T_L, F_L) \]
Discovered model

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

\[
Y_L = \{(\{a\}, \{e\}), (\{c\}, \{d\}), (\{e\}, \{f\}), (\{a,d\}, \{b\}), (\{b\}, \{c,f\})\}
\]
Limitation of $\alpha$ algorithm (implicit places)

$L_6 = [\langle a, c, e, g \rangle^2, \langle a, e, c, g \rangle^3, \langle b, d, f, g \rangle^2, \langle b, f, d, g \rangle^4]$
Limitation of \( \alpha \) algorithm
(loops of length 1)

\[
L_7 = [\langle a, c \rangle^2, \langle a, b, c \rangle^3, \langle a, b, b, c \rangle^2, \langle a, b, b, b, b, c \rangle^1]
\]
Limitation of $\alpha$ algorithm
(loops of length 2)

$$L_8 = [\langle a, b, d \rangle^3, \langle a, b, c, b, d \rangle^2, \langle a, b, c, b, c, b, d \rangle]$$
Limitation of $\alpha$ algorithm (non-local dependencies)

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

Green places are not discovered!

\[
L_4 = [\langle a, c, d \rangle^{45}, \langle b, c, d \rangle^{42}, \langle a, c, e \rangle^{38}, \langle b, c, e \rangle^{22}] 
\]
Difficult constructs for $\alpha$ algorithm
Taking the transactional life-cycle into account
Rediscovering process models

original process model $N$

simulate

event log

discover

discovered process model $N'$

$N = N'$ ?

The rediscovery problem: Is the discovered model $N'$ equivalent to the original model $N$?
Equivalence: trace equivalence, bisimilarity, and branching bisimilarity

Three trace equivalent transition systems: $TS_1$ and $TS_2$ are not bisimilar, but $TS_2$ and $TS_3$ are bisimilar.
Branching bisimilarity defined for YAWL

$TS_1$ and $TS_2$ are not branching bisimilar (although trace equivalent).
Challenge: finding the right representational bias

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

There is no WF-net with unique visible labels that exhibits this behavior.
Another example

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

There is no WF-net with unique visible labels that exhibits this behavior.
Challenge: noise and incompleteness

• To discover a suitable process model it is assumed that the event log contains a representative sample of behavior.

• Two related phenomena:
  − **Noise**: the event log contains rare and infrequent behavior not representative for the typical behavior of the process.
  − **Incompleteness**: the event log contains too few events to be able to discover some of the underlying control-flow structures.
More on incompleteness

To illustrate the relevance of completeness, consider a process consisting of 10 activities that can be executed in parallel and a corresponding log that contains information about 10,000 cases. The total number of possible interleavings in the model with 10 concurrent activities is $10! = 3,628,800$. Hence, it is impossible that each interleaving is present in the log as there are fewer cases (10,000) than potential traces (3,628,800). Even if there are 3,628,800 cases in the log, it is extremely unlikely that all possible variations are present. To motivate this consider the following analogy. In a group of 365 people it is very unlikely that everyone has a different birthdate. The probability is $365!/365^{365} \approx 1.454955 \times 10^{-157} \approx 0$, i.e., incredibly small. The number of atoms in the universe is often estimated to be approximately $10^{79}$ [129].

See also chapter 3 (cross-validation, precision, recall, etc.)
Challenge: Balancing Between Underfitting and Overfitting
Challenge: four competing quality criteria

- **Fitness**: “able to replay event log”
- **Simplicity**: “Occam’s razor”
- **Generalization**: “not overfitting the log”
- **Precision**: “not underfitting the log”

Process Discovery
Flower model
What is the best model?

ACD 99
ACE 0
BCE 85
BCD 0
What is the best model?

ACD 99
ACE 88
BCE 85
BCD 78
What is the best model?

ACD 99
ACE 2
BCE 85
BCD 3
Example: one log four models

```
N1: fitness = +, precision = +, generalization = +, simplicity = +
N2: fitness = +, precision = +, generalization = +, simplicity = +
N3: fitness = +, precision = +, generalization = +, simplicity = +
N4: fitness = +, precision = +, generalization = +, simplicity = +
```

Process discovery

```
# trace
1391 abcdefgh
455 acdeh
191 abdeg
177 acdeh
144 abdeh
111 acdeg
82 acdeg
56 abdeh
47 acdefdbeg
38 abeg
33 acdefbdeh
14 acdefbeg
11 acdefbdeh
9 acdefcdeh
8 acdefdbeg
5 acdefbeg
3 acdefbdeh
2 abcdefd
2 abcdefd
1 dbefbdeh
1 dbefbeg
1 dbefbeg
dbefbeg
```

```
“able to replay event log”
“Occam’s razor”
fitness

“not overfitting the log”
simplicity

generalization

“not underfitting the log”
precision
```
Model $N_1$

$N_1 : fitness = +, precision = +, generalization = +, simplicity = +$
Model $N_2$

$N_2 : \text{fitness} = -, \text{precision} = +, \text{generalization} = -, \text{simplicity} = +$
Model $N_3$

$N_3 : \text{fitness} = +, \text{precision} = -, \text{generalization} = +, \text{simplicity} = +$

<table>
<thead>
<tr>
<th>#</th>
<th>trace</th>
</tr>
</thead>
<tbody>
<tr>
<td>455</td>
<td>acdeh</td>
</tr>
<tr>
<td>191</td>
<td>abdeg</td>
</tr>
<tr>
<td>177</td>
<td>adceh</td>
</tr>
<tr>
<td>144</td>
<td>abdeh</td>
</tr>
<tr>
<td>111</td>
<td>acdeg</td>
</tr>
<tr>
<td>82</td>
<td>adceg</td>
</tr>
<tr>
<td>56</td>
<td>adbeh</td>
</tr>
<tr>
<td>47</td>
<td>acdefdbeh</td>
</tr>
<tr>
<td>38</td>
<td>adbeg</td>
</tr>
<tr>
<td>33</td>
<td>acdefbdeh</td>
</tr>
<tr>
<td>14</td>
<td>acdefbdeg</td>
</tr>
<tr>
<td>11</td>
<td>acdefdbeg</td>
</tr>
<tr>
<td>9</td>
<td>acdefcdeh</td>
</tr>
<tr>
<td>8</td>
<td>acdefdbeh</td>
</tr>
<tr>
<td>5</td>
<td>acdefbdeg</td>
</tr>
<tr>
<td>3</td>
<td>acdefbdefdbeg</td>
</tr>
<tr>
<td>2</td>
<td>acdefdbeg</td>
</tr>
<tr>
<td>2</td>
<td>acdefbdefbdeg</td>
</tr>
<tr>
<td>1</td>
<td>acdefbdefbdeh</td>
</tr>
<tr>
<td>1</td>
<td>adbefbdefdbeg</td>
</tr>
<tr>
<td>1</td>
<td>acdefbdefcdeh</td>
</tr>
<tr>
<td>1391</td>
<td></td>
</tr>
</tbody>
</table>
Model N₄

N₄ : fitness = +, precision = +, generalization = -, simplicity = -
Why is process mining such a difficult problem?

- There are no negative examples (i.e., a log shows what has happened but does not show what could not happen).
- Due to concurrency, loops, and choices the search space has a complex structure and the log typically contains only a fraction of all possible behaviors.
- There is no clear relation between the size of a model and its behavior (i.e., a smaller model may generate more or less behavior although classical analysis and evaluation methods typically assume some monotonicity property).
Creating a 2-D slice of a 3-D reality: the process is viewed from a specific angle, the process is scoped using a frame, and the resolution determines the granularity of the resulting model.