Process Mining: Control-Flow Mining Algorithms

Lecturer: Marlon Dumas

(using slides by Ana Karla Alves de Medeiros, Eindhoven University of Technology – www.processmining.org)
Types of Process Mining

- "world" (business processes, people, components, organizations)
- models analyzes
- specifies configures implements
- supports/controls
- information system
- analyzes
- records events, e.g., messages, transactions, etc.
- (process) model
- event logs
- Process Mining Tools

Diagram:
- Process Model
- Organizational Model
- Social Network
Types of Mining Algorithms

"world" business processes
people components
organizations

models analyzes
specifies configures implements

information system
supports/controls
records events, e.g., messages, transactions, etc.

(process) model

event logs

discovery conformance extension

Process Mining Tools

Compliance Process Model

Auditing/Security
Types of Mining Algorithms

“world”
- business processes
- people
- components
- machines
- organizations

models analyzes

(process) model

specifies configures implements

discovery conformance extension

Process Mining Tools

supports/controls

information system

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

analyze

event logs

Bottlenecks/Business Rules Process Model

Performance Analysis
Control-Flow Mining

Event Log

Discovery Techniques: Control-Flow Mining

Mined Model
Mining Common Constructs

- Sequence
- Splits
- Joins
- Loops
- Non-Free Choice
- Invisible Tasks
- Duplicate Tasks
Mining Common Constructs

- Sequence
- Splits
- Joins
- Loops
- Non-Free Choice
- Invisible Tasks
- Duplicate Tasks

+ noise in logs!
Process Mining Algorithms

- $\alpha$-algorithm
- Heuristics Miner
- Genetic Miner
- Fuzzy Miner
- LP miner (ProM 6)
α-algorithm

1. Read a log
2. Get the set of tasks
3. Infer the ordering relations \textcolor{blue}{Core Step!}
4. Build the net based on inferred relations
5. Output the net
• **Direct succession:** $x > y$ iff for some case $x$ is directly followed by $y$.
• **Causality:** $x \rightarrow y$ iff $x > y$ and not $y > x$.
• **Parallel:** $x \parallel y$ iff $x > y$ and $y > x$.
• **Unrelated:** $x \# y$ iff not $x > y$ and not $y > x$. 

\(\alpha\)-algorithm - Ordering Relations >, →, ||, #
α-algorithm - Insight
<table>
<thead>
<tr>
<th>Case ID</th>
<th>Task Name</th>
<th>Originator</th>
<th>Timestamp</th>
<th>Case ID</th>
<th>Task Name</th>
<th>Originator</th>
<th>Timestamp</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>File Fine</td>
<td>Anne</td>
<td>20-07-2004 14:00:00</td>
<td>3</td>
<td>Reminder</td>
<td>John</td>
<td>21-08-2004 10:00:00</td>
</tr>
<tr>
<td>2</td>
<td>File Fine</td>
<td>Anne</td>
<td>20-07-2004 15:00:00</td>
<td>2</td>
<td>Process Payment</td>
<td>system</td>
<td>22-08-2004 09:05:00</td>
</tr>
<tr>
<td>1</td>
<td>Send Bill</td>
<td>system</td>
<td>20-07-2004 15:05:00</td>
<td>2</td>
<td>Close case</td>
<td>system</td>
<td>22-08-2004 09:06:00</td>
</tr>
<tr>
<td>2</td>
<td>Send Bill</td>
<td>system</td>
<td>20-07-2004 15:07:00</td>
<td>4</td>
<td>Reminder</td>
<td>John</td>
<td>22-08-2004 15:10:00</td>
</tr>
<tr>
<td>3</td>
<td>File Fine</td>
<td>Anne</td>
<td>21-07-2004 10:00:00</td>
<td>4</td>
<td>Reminder</td>
<td>Mary</td>
<td>22-08-2004 17:10:00</td>
</tr>
<tr>
<td>3</td>
<td>Send Bill</td>
<td>system</td>
<td>21-07-2004 14:00:00</td>
<td>4</td>
<td>Process Payment</td>
<td>system</td>
<td>29-08-2004 14:01:00</td>
</tr>
<tr>
<td>4</td>
<td>File Fine</td>
<td>Anne</td>
<td>22-07-2004 11:00:00</td>
<td>4</td>
<td>Close Case</td>
<td>system</td>
<td>29-08-2004 17:30:00</td>
</tr>
<tr>
<td>4</td>
<td>Send Bill</td>
<td>system</td>
<td>22-07-2004 11:10:00</td>
<td>3</td>
<td>Reminder</td>
<td>John</td>
<td>21-09-2004 10:00:00</td>
</tr>
<tr>
<td>1</td>
<td>Process Payment</td>
<td>system</td>
<td>24-07-2004 15:05:00</td>
<td>3</td>
<td>Reminder</td>
<td>John</td>
<td>21-10-2004 10:00:00</td>
</tr>
<tr>
<td>1</td>
<td>Close Case</td>
<td>system</td>
<td>24-07-2004 15:06:00</td>
<td>3</td>
<td>Process Payment</td>
<td>system</td>
<td>25-10-2004 14:00:00</td>
</tr>
<tr>
<td>2</td>
<td>Reminder</td>
<td>Mary</td>
<td>20-08-2004 10:00:00</td>
<td>3</td>
<td>Close Case</td>
<td>system</td>
<td>25-10-2004 14:01:00</td>
</tr>
</tbody>
</table>

www.processmining.org
α-algorithm – Log properties + target nets

- If log is complete with respect to relation >, it can be used to mine SWF-net without short loops
- *Structured Workflow Nets* (SWF-nets) have no implicit places and the following two constructs cannot be used:
α-algorithm – No short loops

B>B and not B>B implies B→B (impossible!)

A>B and B>A implies A||B and B||A instead of A→B and B→A
α-algorithm – Common Constructs

- No invisible tasks, non-free-choice or duplicate tasks
- No noisy logs

Why no duplicate tasks?
Why not invisible tasks?
Why noise-free logs?
Process Mining Algorithms

• $\alpha$-algorithm
• Heuristics Miner
• Genetic Algorithm
• Fuzzy Miner
Heuristics Miner

1. Read a log
2. Get the set of tasks
3. Infer the ordering relations based on their frequencies
4. Build the net based on inferred relations
5. Output the net
Heuristics Miner

Let $W$ be an event log over $T$, and $a, b \in T$:

- $|a >_W b|$ is the number of times $a >_W b$ occurs in $W$,
- $a \Rightarrow_w b = \left( \frac{|a >_W b| - |b >_W a|}{|a >_W b| + |b >_W a| + 1} \right)$

- The more frequently a task $A$ directly follows another task $B$, and the less frequently the opposite occurs, the higher the probability that $A$ causally follows $B$!
- Robust to invisible tasks and noisy logs
- No non-free-choice or duplicate tasks
Process Mining Algorithms

- \( \alpha \)-algorithm
- Heuristics Miner
- Genetic Miner
- Fuzzy Miner
Genetic Process Mining (GPM)

• Genetic Algorithms + Process Mining

• Genetic Algorithms
  – Search technique that mimics the process of evolution in biological systems

• Advantages
  – Tackle all common structural constructs
  – Robust to noise

• Disadvantages
  – Computational Time
Genetic Process Mining (GPM)

Algorithm:

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>Read event log</td>
</tr>
<tr>
<td>II</td>
<td>Build the initial population</td>
</tr>
<tr>
<td>III</td>
<td>Calculate fitness of the individuals in the population</td>
</tr>
<tr>
<td>IV</td>
<td>Stop and return the fittest individuals?</td>
</tr>
<tr>
<td>V</td>
<td>Create next population — use elitism and genetic operators</td>
</tr>
</tbody>
</table>

Internal Representation
Fitness Measure
Genetic Operators
GPM – Fitness Measure

- Guides the search!
GPM – Fitness Measure
GPM – Fitness Measure

Overgeneral solution

Punish for the amount of enabled tasks during the parsing!
GPM – Fitness Measure

Overspecific solution

Punish for the amount of duplicate tasks with common input/output tasks!
Process Mining Algorithms

- α-algorithm
- Heuristics Miner
- Genetic Miner
- Fuzzy Miner
Fuzzy Miner - Motivation

Mine less structured processes!
Fuzzy Miner - Motivation

- More significant nodes are emphasized
- Highlights more important paths
Fuzzy Miner

More to learn from maps...

**Aggregation**
Clustering of coherent, less significant structures

**Abstraction**
Removing isolated, less significant structures
No “Ask Question” or “Give Talk”!

Abstracting even more from details!

All details!