Process Mining: Control-Flow Mining Algorithms

Ana Karla Alves de Medeiros

Eindhoven University of Technology
Department of Information Systems
a.k.medeiros@tue.nl
Process Mining

- Short Recap
- Types of Process Mining Algorithms
- Common Constructs
- Input Format
- $\alpha$-algorithm
- Heuristics Miner
- Genetic Miner
- Fuzzy Miner
Process Mining

- **Short Recap**
- Types of Process Mining Algorithms
- Common Constructs
- Input Format
- $\alpha$-algorithm
- Heuristics Miner
- Genetic Miner
- Fuzzy Miner
Process Mining

Event Log

Mining Techniques

Process Model

Organizational Model

Social Network

Performance Analysis

Auditing/Security

Mined Models

/faculteit technologie management
Event Logs are Everywhere!

Machines, Municipalities, Airports, Internet, Hospitals, etc.
Tools

- www.processmining.org
- ProM 4.2
- ProMimport
- Free tools!
Process Mining

• Short Recap
• Types of Process Mining Algorithms
• Common Constructs
• Input Format
• $\alpha$-algorithm
• Heuristics Miner
• Genetic Miner
• Fuzzy Miner
Process Mining

• Short Recap
• **Types of Process Mining Algorithms**
• Common Constructs
• Input Format
• $\alpha$-algorithm
• Heuristics Miner
• Genetic Miner
• Fuzzy Miner
Types of Algorithms

The diagram illustrates the relationship between "world" components (business processes, people, machines, organizations) and an information system. The system supports/controls these components. "World" models and (process) models analyze and model these components. The (process) model specifies, configures, implements, and analyzes processes. The information system records events, e.g., messages, transactions, etc. Process Mining Tools discover conformance and extension of processes.
Types of Algorithms

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

Process Model
Organizational Model
Social Network

/ faculteit technologie management
Types of Algorithms

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

models
- analyzes
- configures
- implements

information system
- supports/controls
- analyzes
- specifies

(process) model
- (process) model

event logs
- records, e.g., events, messages, transactions, etc.

Process Mining Tools
- discovery
- conformance
- extension

Compliance Process Model

Auditing/Security

Facade technologie management
Types of Algorithms

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

supports/controls

information system

models analyzes

specifies configures implements

discovery

conformance

extension

(event) model

(event) logs

Performance Analysis

Bottlenecks/Business Rules Process Model

Process Mining Tools

/receive payment
- contact customer

archive order

(ship goods)

prepare shipment

register order

start
Process Mining

Discovery Techniques: Control-Flow Mining

//faculteit technologie management
Process Mining

- Short Recap
- Types of Process Mining Algorithms
- Common Constructs
- Input Format
- \( \alpha \)-algorithm
- Heuristics Miner
- Genetic Miner
- Fuzzy Miner
Process Mining

• Short Recap
• Types of Process Mining Algorithms
• Common Constructs
• Input Format
• $\alpha$-algorithm
• Heuristics Miner
• Genetic Miner
• Fuzzy Miner
Common Constructs

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

/ faculteit technologie management
Common Constructs

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

+ noise in logs!
Process Mining

• Short Recap
• Types of Process Mining Algorithms
• Common Constructs
• Input Format
• $\alpha$-algorithm
• Heuristics Miner
• Genetic Miner
• Fuzzy Miner
Process Mining

• Short Recap
• Types of Process Mining Algorithms
• Common Constructs
• Input Format
• α-algorithm
• Heuristics Miner
• Genetic Miner
• Fuzzy Miner
The notion of which tasks belong to a same instance is crucial for applying process mining techniques!
Event Log: Mining XML (MXML)

```xml
<WorkflowLog>
  <Process>
    <ProcessInstance>
      <AuditTrailEntry/>
      <AuditTrailEntry/>
      <AuditTrailEntry/>
    </ProcessInstance>
    <ProcessInstance/>
    <ProcessInstance/>
  </Process>
</WorkflowLog>
```
Event Log: Mining XML (MXML)

```
<WorkflowLog>
  <Process>
    <AuditTrailEntry>
      <WorkflowModelElement> Task A </WorkflowModelElement>
      <EventType> complete </EventType>
      <TimeStamp> 2005-10-26T12:37:33... </TimeStamp>
      <Originator> John Doe </Originator>
      <Data>
        <Attribute name="x"> 1 </Attribute>
        <Attribute name="y"> whatever </Attribute>
      </Data>
      <AuditTrailEntry>
  </Process>
</WorkflowLog>
```

Compulsory fields!
Process Mining

• Short Recap
• Types of Process Mining Algorithms
• Common Constructs
• Input Format
• $\alpha$-algorithm
• Heuristics Miner
• Genetic Miner
• Fuzzy Miner
Process Mining

• Short Recap
• Types of Process Mining Algorithms
• Common Constructs
• Input Format
• $\alpha$-algorithm
• Heuristics Miner
• Genetic Miner
• Fuzzy Miner
$\alpha$-algorithm

1. Read a log
2. Get the set of tasks
3. Infer the ordering relations ◼️ 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:** \( x || y \) iff \( x > y \) and \( y > x \)

**Unrelated:** \( x \# y \) iff not \( x > y \) and not \( y > x \).

### Ordering Relations

- \( > \)
- \( \rightarrow \)
- \( || \)
- \( \# \)

<table>
<thead>
<tr>
<th>Case</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A</td>
</tr>
<tr>
<td>2</td>
<td>A</td>
</tr>
<tr>
<td>3</td>
<td>A</td>
</tr>
<tr>
<td>4</td>
<td>A</td>
</tr>
<tr>
<td>1</td>
<td>B</td>
</tr>
<tr>
<td>2</td>
<td>B</td>
</tr>
<tr>
<td>1</td>
<td>C</td>
</tr>
<tr>
<td>2</td>
<td>C</td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
</tbody>
</table>

\[ \alpha \text{-algorithm - } \]

- \( A \rightarrow B \)
- \( A \rightarrow C \)
- \( B \rightarrow D \)
- \( C \rightarrow D \)
- \( E \rightarrow F \)

\( \text{ABCD} \)
\( \text{ACBD} \)
\( \text{EF} \)
Let $W$ be a workflow log over $T$. $\alpha(W)$ is defined as follows.

1. $T_W = \{ t \in T \mid \exists \sigma \in W \ t \in \sigma \}$,
2. $T_I = \{ t \in T \mid \exists \sigma \in W \ t = first(\sigma) \}$,
3. $T_O = \{ t \in T \mid \exists \sigma \in W \ t = last(\sigma) \}$,
4. $X_W = \{ (A,B) \mid A \subseteq T_W \land B \subseteq T_W \land \forall a \in A \forall b \in B \ a \rightarrow_W b \land \forall a_1,a_2 \in A \ a_1 \#_W a_2 \land \forall b_1,b_2 \in B \ b_1 \#_W b_2 \}$,
5. $Y_W = \{ (A,B) \in X \mid \forall (A',B') \in X \ A \subseteq A' \land B \subseteq B' \Rightarrow (A,B) = (A',B') \}$,
6. $P_W = \{ p_{(A,B)} \mid (A,B) \in Y_W \} \cup \{ i_W,o_W \}$,
7. $F_W = \{ (a,p_{(A,B)}) \mid (A,B) \in Y_W \land a \in A \} \cup \{ (p_{(A,B)},b) \mid (A,B) \in Y_W \land b \in B \} \cup \{ (i_W,t) \mid t \in T_I \} \cup \{ (t,o_W) \mid t \in T_O \}$, and
8. $\alpha(W) = (P_W,T_W,F_W)$. 
α-algorithm - Insight
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:
\[ B \succ B \text{ and not } B \succ B \text{ implies } B \rightarrow B \text{ (impossible!)} \]

\[ A \succ B \text{ and } B \succ A \text{ implies } A \parallel B \text{ and } B \parallel A \text{ instead of } A \rightarrow B \text{ and } B \rightarrow A \]

Why no short loops?
Why no duplicate tasks?

Why not invisible tasks?

Why noise-free logs?

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

• Short Recap
• Types of Process Mining Algorithms
• Common Constructs
• Input Format
• $\alpha$-algorithm
• Heuristics Miner
• Genetic Algorithm
• Fuzzy Miner
Process Mining

• Short Recap
• Types of Process Mining Algorithms
• Common Constructs
• Input Format
• $\alpha$-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 $W$,
- $a \Rightarrow_W b = \left( \frac{|a >_W b| - |b >_W a|}{|a >_W b| + |b >_W a| + 1} \right)$

Insight:

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 $B$!
\(\alpha\)-algorithm – Common Constructs

- No non-free-choice or duplicate tasks
- Robust to invisible tasks and noisy logs

Why no non-free-choice?

Why no guarantee whole net will be correct?
Process Mining

• Short Recap
• Types of Process Mining Algorithms
• Common Constructs
• Input Format
• $\alpha$-algorithm
• Heuristics Miner
• Genetic Miner
• Fuzzy Miner
Process Mining

• Short Recap
• Types of Process Mining Algorithms
• Common Constructs
• Input Format
• $\alpha$-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!

 faculteit technologie management
GPM – DGA ProM Plug-in

Why does the GA Miner takes so much time?

How could we improve its running time without changing the code?
Process Mining

- Short Recap
- Types of Process Mining Algorithms
- Common Constructs
- Input Format
- $\alpha$-algorithm
- Heuristics Miner
- Genetic Miner
- Fuzzy Miner
Process Mining

• Short Recap
• Types of Process Mining Algorithms
• Common Constructs
• Input Format
• $\alpha$-algorithm
• Heuristics Miner
• Genetic Miner
• Fuzzy Miner
Fuzzy Miner - Motivation

Mine less structured processes!

/faculteit technologie management
Fuzzy Miner - Motivation

More significant nodes are emphasized

Highlights more important paths

/faculteit technologie management
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!
Conclusions

• The notion of a process instance is crucial!
• Ordering of tasks is the basic information
• Frequencies are important to handle noise
• Local approaches
  – $\alpha$-algorithm, Heuristics Miner
• Global approaches
  – Genetic Miner and Fuzzy Miner