



University of Tirana

Faculty of Natural Science

Master of Science in Information System Engineering

Process Mining with Event Logs

This report provides a basic analysis of a process derived from an event log. The analysis was done using the Alpha Miner algorithm, which discovered a process model (Petri Net) that illustrates the sequence and flow of activities in the process.

Process Overview

The event log used in this analysis captures the following activities:

- Submit Application
- Review Application
- Accept Application
- Reject Application

Each activity in the log is associated with a timestamp, and the event log provides information on when each activity occurred within different cases.

Process Flow

Using the Alpha Miner algorithm, a Petri Net was generated to visualize the process flow. The Petri Net clearly defines how the activities are interconnected, showing the main path from submitting the application to either accepting or rejecting it.

Key Observations

- Main Activities:

The primary activities in the process include submitting the application, reviewing it, and the decision of either accepting or rejecting the application.

- Possible Outcomes:

There are two possible outcomes for each application: it can either be accepted or rejected. This branching behavior is clearly depicted in the Petri Net.

- Process Flow Consistency:

The process may not always follow the same sequence, as applications can be rejected after submission, leading to different paths in the workflow. This variability is captured in the Petri Net.

Conclusion

The analysis shows that the process is relatively simple, with clear branching based on whether the application is accepted or rejected. The Alpha Miner algorithm successfully captured the main activities and possible outcomes in the event log, providing a clear understanding of how the process works.

```
!pip install pm4py

import pandas as pd

df = pd.read_csv("proceset.csv")

print(df.head())

df['Timestamp'] = pd.to_datetime(df['Timestamp'], errors='coerce')

print(df.head())

df = df.rename(columns={
    'CaseID': 'case:concept:name',
    'Activity': 'concept:name',
    'Timestamp': 'time:timestamp',
    'InitialStatus': 'initial_status',
    'FinalStatus': 'final_status',
    'ProcessFlow': 'process_flow'
})

print(df.columns)
```

```
from pm4py.objects.conversion.log import converter as log_converter

from pm4py.objects.log.util import dataframe_utils

df = dataframe_utils.convert_timestamp_columns_in_df(df)

log = log_converter.apply(df)

print(log[0])

from pm4py.algo.discovery.alpha import algorithm as alpha_miner

net, initial_marking, final_marking = alpha_miner.apply(log)

from pm4py.visualization.petri_net import visualizer as pn_visualizer

gviz = pn_visualizer.apply(net, initial_marking, final_marking)

pn_visualizer.view(gviz)
```

