Discovering Deterministic Finite State Automata from Event Logs for Business Process Analysis

Simone Agostinelli[1], Francesco Chiariello[1], Fabrizio Maria Maggi[2], Andrea Marrella[1], Fabio Patrizi[1]

[1] DIAG - Sapienza University of Rome, Italy
[2] KRDB - Free University of Bozen-Bolzano, Italy
chiariello@diag.uniroma1.it

Objective

- Leverage Model Learning (ML) for the automated discovery of Deterministic Finite-state Automata (DFAs) from event logs.
- No additional information used, such as DECLARE templates.

Process Mining

- Process Mining (PM):
  - Research area from Business Process Management (BPM).
  - Analysis of process data recorded in event logs to gain insight into business processes.

DFAs for PM

- DFAs have been employed for different PM tasks, e.g.,:
  - Process Discovery: select the constraints most frequently satisfied in an event log.
  - Conformance Checking: check whether a trace is a process instance veryfing whether the trace is accepted by the DFA.
  - Compliance Monitoring: verify, at run-time, the compliance of ongoing process executions with respect to some expected de-jure behavior.

Discovering DFAs

- To support the above BPM use cases, techniques for discovering DFAs from event logs are very valuable.
- Model Learning: algorithms for constructing black-box finite-state diagram models of systems relying on observed input–output data

Model Learning

- Main categories of ML algorithms:
  - Active Learning algorithms work by posing queries to the System Under Learning (SUL).
  - Passive Learning algorithms learn the behavioral model of the SUL from a pre-defined set of training data.

Contribution

- We investigate the effectiveness of the best-known passive learning algorithms.
- Cases considered:
  - only positive traces,
  - both positive and negative traces.
- The use of negative traces is greatly beneficial in improving the quality of the learnt process.

Passive Learning Algorithms

- Algorithms considered
  - Minimum Description Length (MDL)
  - Regular Positive and Negative Inference (RPNI)
  - Evidence Driven State Merging (EDSM)
  - L* [1]
- Comparison with the SoA tool Declare Miner.

[1] Adapted for passive learning.

Quality Metrics

- Metrics for evaluating the quality of the discovered DFAs:
  - Precision: degree to which the behaviors allowed by the process model are observed in the event log,
  - Fitness (or Recall): degree to which the behaviors observed in the event log are allowed by the process model;
  - Generalization: estimation of how well a model inferred will reproduce future behaviors not seen;
  - Simplicity: size of the model.

Real-life Logs

Table: Descriptive statistics of real-life logs.

Log Name	Total traces	Total$^+$ traces	Total$^-$ traces	Distinct traces (%)	Total events	Activity types	Trace length		
							min	avg	max
LOAN	13,087	8164	4923	33.4	262,200	36	3	20	175
ROAD	150,370	82737	67633	0.2	561,470	11	2	4	20
SEPSIS	1,050	838	212	80.6	15,214	16	3	14	185
REIMB	6,449	4248	2201	11.7	72,151	34	3	11	27
TRAVEL	7,065	4249	2816	20.9	86,581	51	3	12	90

Results

- Active learning algorithms are not suitable to generate DFAs from real-life event logs.
- Declare Miner and passive learning algorithms construct DFAs with similar values of generalization and precision.
- Passive learning algorithms generate simpler DFAs than Declare Miner.

Future Work

- Learn LTL_f formulae:
  - directly from logs, or,
  - going through Alternating Finite Automa.
- both approaches possible with SAT or ASP techniques (remember yesterday talk ☺)

Thank you!!