

Conformance Checking Modulo Theories for Multi-Perspective Processes

Paolo Felli,¹ Alessandro Gianola,² Marco Montali,² Andrey Rivkin,³ Sarah Winkler²

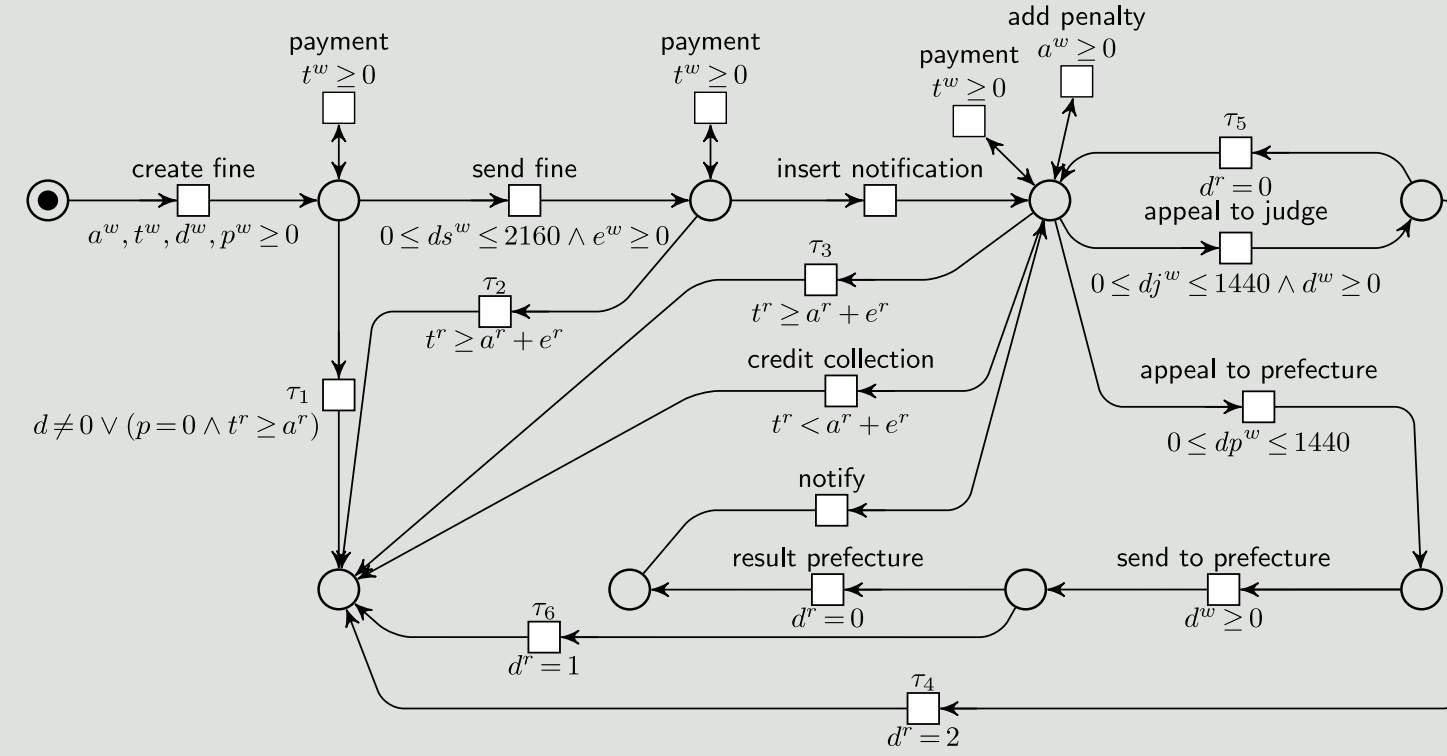
¹ Università di Bologna, Bologna, Italy

² Free University of Bozen-Bolzano, Bolzano, Italy

³ Technical University of Denmark, Copenhagen, Denmark

multi-perspective conformance checking

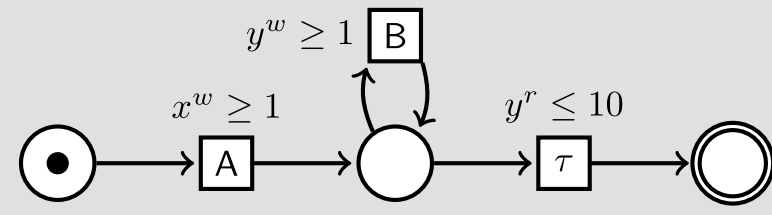
- check process model against behavior recorded in log
- activities come with payload: data of different types
- real-life logs: activities, data events, and timestamps can be uncertain



Data Petri Nets

- can be mined automatically from logs
- good tradeoff between simplicity and expressiveness
- maintain global variables
- arithmetic constraint guards

optimal alignments



| | | | |
|--------|--------|------|--|
| A, x=3 | B, y=0 | B, ∅ | |
| A, x=3 | B, y=1 | τ, ∅ | |

✓

| | | | |
|--------|--------|------|------|
| A, x=3 | B, y=0 | B, ∅ | » |
| A, x=3 | B, y=1 | » | τ, ∅ |

| | | | | | |
|--------|--------|--------|--------|------|------|
| » | A, x=3 | B, y=0 | » | B, ∅ | » |
| A, x=3 | » | » | B, y=1 | » | τ, ∅ |

data-aware cost function

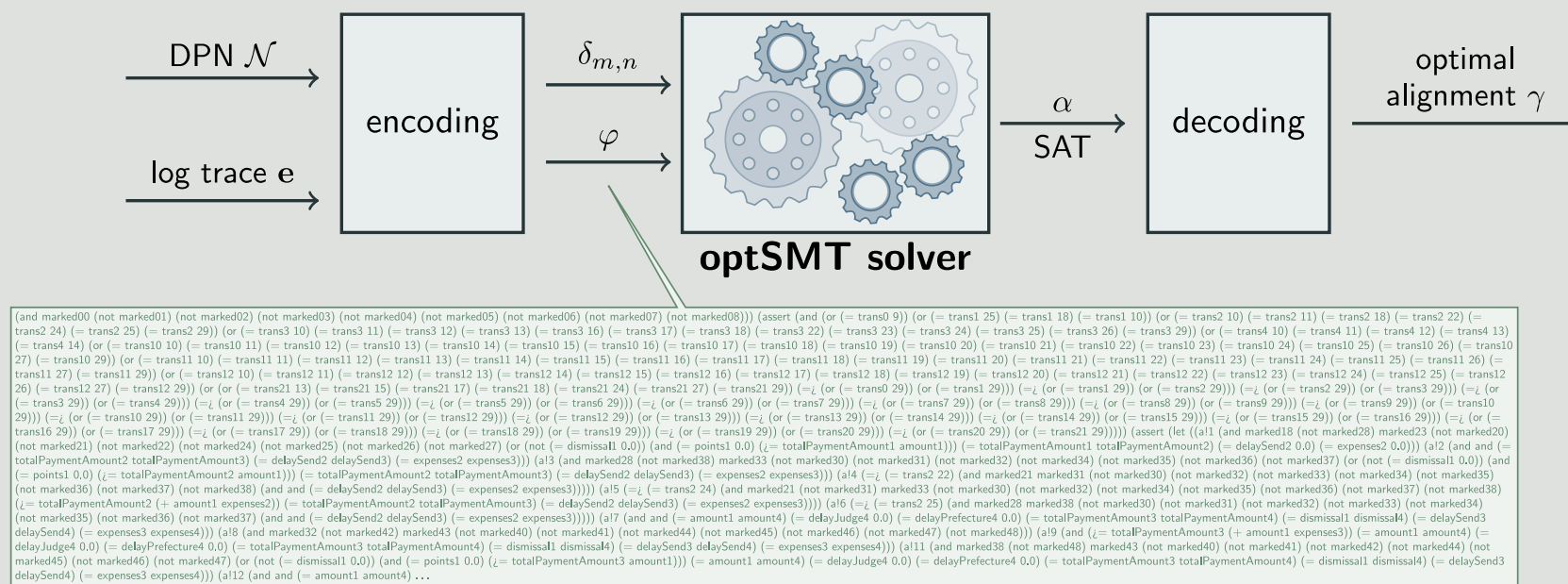
- edit distance with data-aware penalty functions

$$\begin{aligned} \delta_{0,0} &= 0 \\ \delta_{i+1,0} &= \delta_{i,0} + P_L(e_{i+1}) \\ \delta_{0,j+1} &= \delta_{0,j} + P_M(f_{j+1}) \\ \delta_{i+1,j+1} &= \min \begin{cases} \delta_{i,j} + P_e(e_{i+1}, f_{j+1}) \\ \delta_{i,j+1} + P_L(e_{i+1}) \\ \delta_{i+1,j} + P_M(f_{j+1}) \end{cases} \end{aligned}$$

- search for optimal alignment cost

| | | | | | |
|---|---|---|-----|---|-----|
| | | e | | | |
| | 0 | 1 | 2 | 3 | 4 |
| f | 1 | 0 | → 1 | 2 | 3 |
| | 2 | 1 | 2 | 2 | 3 |
| | 3 | 2 | 1 | 2 | → 3 |

SMT encoding



flexibility of SMT approach

- approach extends seamlessly to multi- and anti-alignments
- other cost functions can easily be encoded
- future work: more expressive guard language and models

conformance checking of traces with uncertainty

- logging systems induce uncertainty: imprecise sensors, lack of reliability
- consider four types of uncertainty
 - uncertain events
 - uncertain activities
 - uncertain timestamps
 - uncertain data
- aim: find optimal alignment for realization of trace with uncertainty
- cost function can take uncertainty information into account

implementation: cocomot

- implemented in Python using different SMT solver backends
- supports also multi- and anti-alignments
- trace clustering as preprocessing
- supports conformance checking of logs with uncertainty

