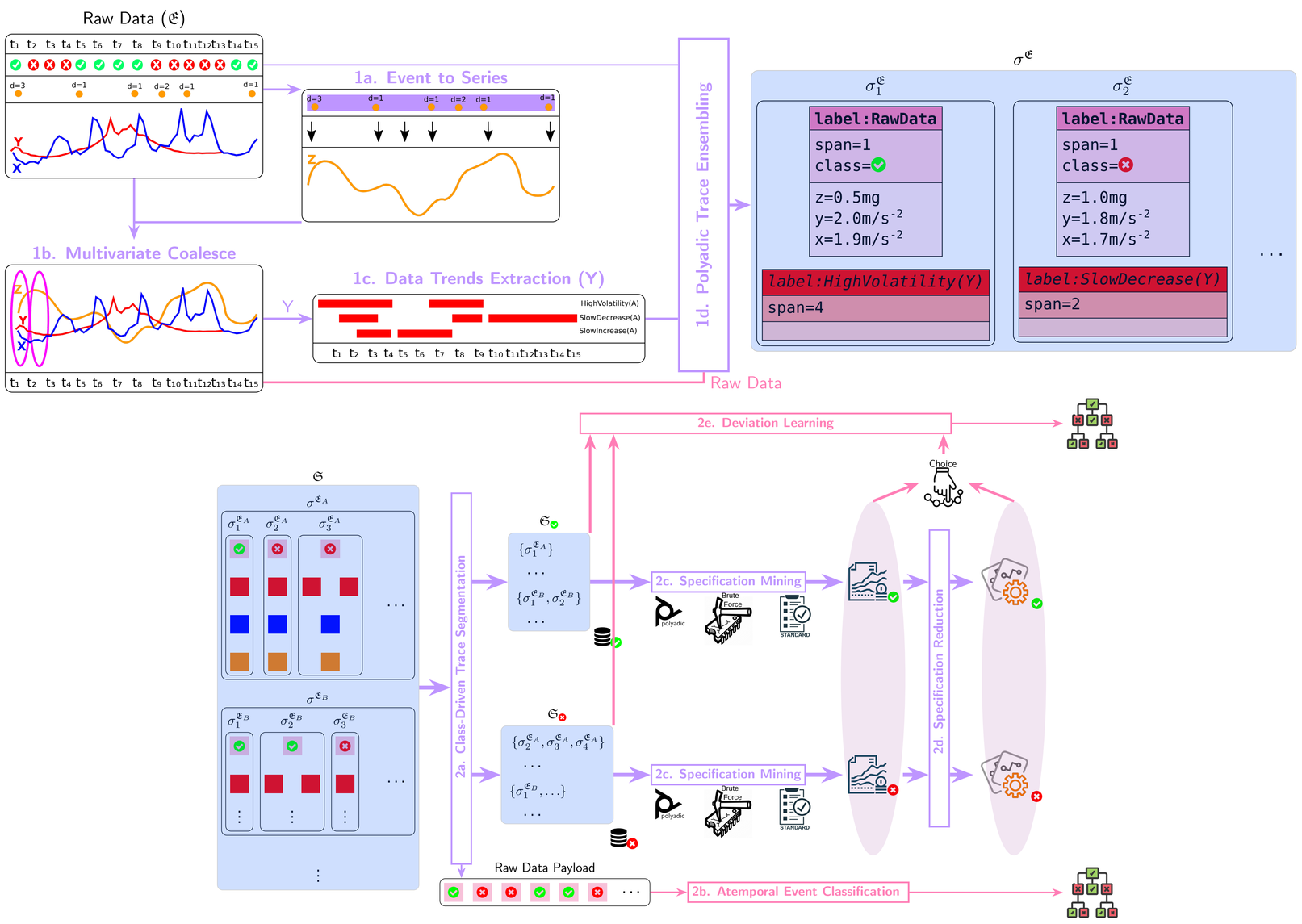
KnoBAB: Making Logic Fast

Giacomo.Bergami@newcastle.ac.uk
School of Computing, Newcastle University

Birds-Eye View



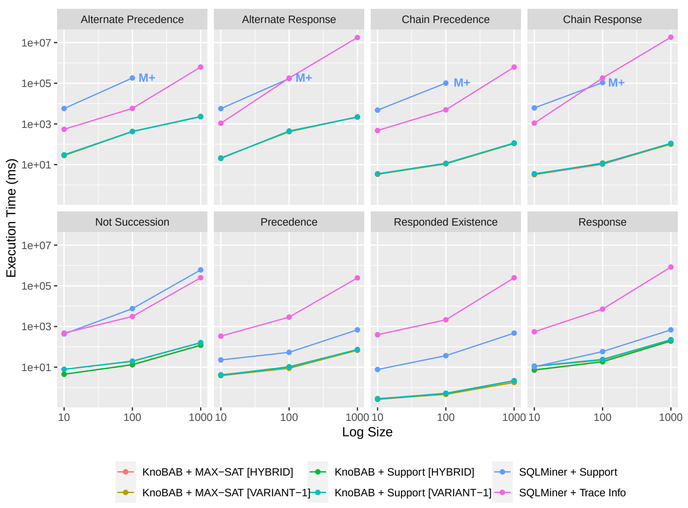
Relational Models

- Column DBMS for efficiently representing sparse relations. All indices exploit linear data structures.
- Intermediate Results exploit the Nested Relational Model. The relaxation of the 1NF allows the association of multiple activation, target, or correlation conditions to any event, while a sorted linear representation allows efficient data scanning.

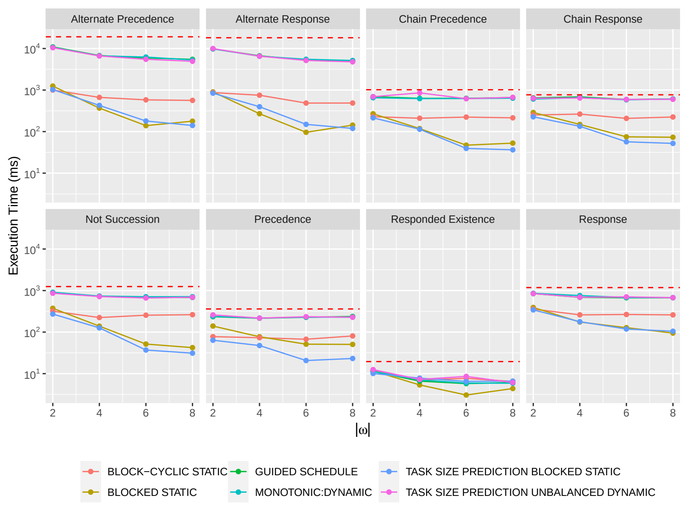
Temporal Data Analysis

- This task can be applied both to numerical time series, as well as over durative sequence of timestamped actions, both associated to a classification label (desired vs. undesired state).
- When possible, we support the transformation of single durative events into furhter time series.
- The proposed methodology allows to extract an explainable model through temporal correlations and numerical data trends.
- We extract both a human-readable characterization of the behaviour of each specific system state, as well as the temporal behaviour differentiating those.

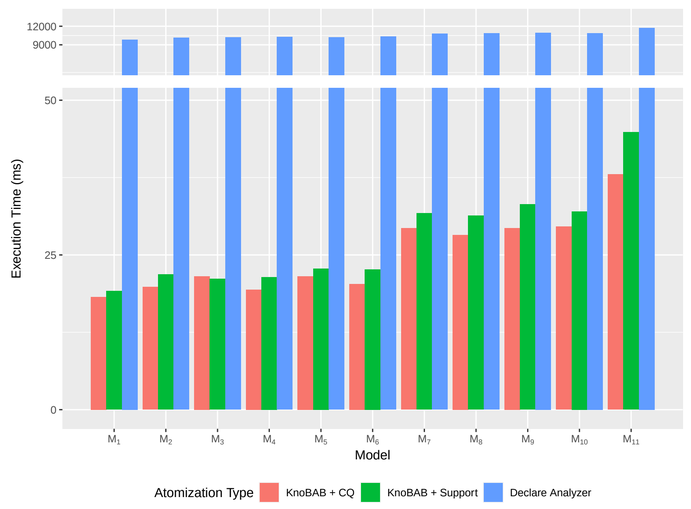
Model Mining (pt1)



Parallelisation

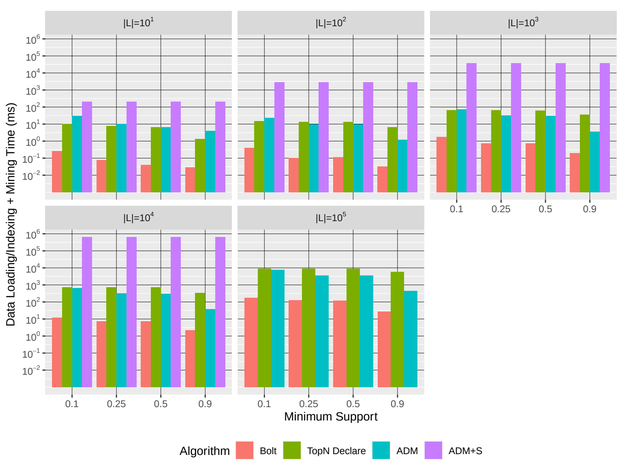


Data-Aware Model Checking



Model Mining (pt2) & Mimimal Model Description Length

Log Gen. Model, $\mathcal{M}^*$	Bolt	TopN	ADM
ChainPrecedence(b, c)	✓	✗	✓(+Choice,CoExist.)
ChainResponse(d, e)	✓	✗	✓(+Choice,CoExist.)
Choice(f, g)	RespExistence(g, f)	✓(+Choice(g, f))	✓(+Choice(g, f))
ExclChoice(h, a)	✓	✗	✗
Exists(b, 1)	✓	✗	✓
Init(e)	✓	✓	✓
Precedence(c, b)	✓	✗	✓(+Choice,RespEx.,CoEx.)
RespExistence(e, f)	Response(e, f)	✓	✓
RespExistence(h, i)	CoExistence(h, i)	✗	✓(+ CoExistence(h, i))
RespExistence(i, h)		✗	✓(+ CoExistence(i, h))
Response(e, f)	✓	✓(+ RespExistence(e, f))	✓(+ RespExistence(e, f))
$\lvert\mathcal{M}_\theta\rvert$	197	112	548
$\lvert\mathcal{M}_{\geq 0.999}\rvert$	46	35	119
Mining time (μ)	$1.73 \cdot 10^0$ms	$5.17 \cdot 10^1$ms	$1.40 \cdot 10^2$ms



This result shows that our algorithm can achieve minimal model description length if compared to current mining algorithms without requiring an explicit "learning algorithm" for model minimality and no further additional metric to the confidence and support metrics, that are also used as branch&bound heuristic for quickening the model mining algorithm.
At last, logic can stand besides learning based approaches while achieving algorithmical optimality!

Customisability & Interoperability

- ✓ Defining the declarative semantics at runtime.
- ✓ Declarative Model Mining.
- ✓ Data/Dataless log generation from models.
- ✓ Supporting multiple logs.
- ✓ Terminal client.
- ✓ HTTP Requests through JDBC-like library.

Transparency

- GoogleTests and Continuous Integration, for guaranteeing that the code, once compiled, works and that all of the tests are satisfactorly run!
- Tests provide empirical evidence for the correctenss of some theoretical proofs.
- Datasets publicly available through **OSF.org**
- `https://github.com/datagram-db/knobab`