Symbolic Monitoring against Specification Parametric in Time and Data

Masaki Waga[1,2], Étienne André[3,4], Ichiro Hasuo[1,2]

Background
Runtime Verification (RV)
Runtime verification (or monitoring) is not exhaustive but
- it does not require system model
- i.e., Blackbox systems are OK
- e.g., System w/
  - Machine learning components
  - 3rd-party components
  - Unknown environment
- It tends to be scalable for complex systems

Issues in RV
- It is difficult to determine the best threshold in the spec.
  - e.g., What is the best threshold of “too large acceleration”?
- We want quantitative results rather than Boolean results
- Clearly satisfied vs. Satisfied but near the borderline

Our Solution
Symbolic Monitoring w/ Parametric Specifications

Symbolic Monitoring
Input
- Time-series data
  - System log (event + data + timestamp)
  - e.g., withdrawal(Alice, 100 USD), withdrawal(Bob, 120 USD)
- Parameterized real-time spec. with data
  - Spec. to be monitored
    - e.g., the total amount of p-days withdrawal by user N should be < T USD

Output
- All of the param. val. such that the log satisfies the spec.
  - e.g., (N, V, T, p) = (Alice, 140, 3, 0), (Alice, 135, 4, 0), (Bob, 20, 1, 0), ...
- Infinitely many → Symbolic representation

Contribution
- Introduced parametric timed data automata (PTDA)
  - PTDA: NFA + clock/data variables + time/data parameters
- Gave symbolic monitoring algorithm over a PTDA spec.
  - Idea: follow trans. using symbolic representation
  - (Potentially) infinitely many param. val. → symbolic representation/operations
- Experiments → Scalable!!
  - Demo is available on Google Colab

Official Version arXiv Version Tool Demo

Symbolic Monitoring Algorithm

Detail on PTDA

Example of a PTDA
withdraw(u, a)
withdraw(u, a)
withdraw(u, a), u = N
start
f0
H := 0, sum := a
l1
l2
l3
f1
f2
f3
f4

Q: What is “Data”?
A: Any triple (D, DE, DU)

D: infinite domain
DE: Boolean expression (for guards)
DU: updates (for variable updates/assignments)

Tool Demo

Scalability
Detects dominant withdrawals by a user
Scalable: Periodic

Detects dominant withdrawals by a user
Scalable: Periodic

Contribution
- 20,000 entries in 1 - 2 min.
- Execution time is linear in these two benchmarks
- Much more efficient than the worst case!!
- Important lesson: these useful spec. can be monitored efficiently

Idea of the Algorithm
follow the transitions of PTDA
+ abstraction of clock/data/param. val.
(e.g., by convex polyhedra or lists of forbidden strings)
+ (Non-deterministic branching by breadth first search)

Termination
Our symbolic monitoring algorithm terminates for any data types (D, DE, DU) such that we can compute restriction, update, emptiness checking, and projection.

Examples
- Strings (S) with lists (of forbidden strings)
- Rationals (Q) with convex polyhedra

See also an illustration!!

This work is partially supported by JST ERATO HASUO Metamathematics for Systems Design Project (No. JPMJER1903), by JSPS Grants-in-Aid No. 15KT0012 & 18J22498, and by the ANR national research program PACS (ANR-14-CE28-0002).