The Online Monitoring Problem

property in a specification language

event stream

monitor

verdict stream

verdicts denote whether the property holds at EVERY position in the event stream
not considered: instrumentation (how to generate the event stream)

specification language

Metric Dynamic Logic (MDL)

\[ \psi, \varphi = p \mid \neg p \mid \psi \land \varphi \mid \psi \lor \varphi \mid r.s = \star_i \psi \mid r+s \mid rs \mid r^* \]

the usual syntactic sugar
(until, next, since, previous, always, eventually, once, historically, ...)
more expressive than MTL
incomparable to MFOTL (propositional but regex)
future intervals may be unbounded

Example

\[
\begin{align*}
\neg r & \quad \text{true} \\
\neg r & \quad \text{false} \\
\neg s & \quad \text{false} \\
\neg s & \quad \text{true} \\
\neg r.s & \quad \text{false} \\
\neg r.s & \quad \text{true} \\
\neg r^* & \quad \text{false} \\
\neg r^* & \quad \text{true} \\
\psi & \quad \text{false} \\
\psi & \quad \text{true} \\
\varphi & \quad \text{false} \\
\varphi & \quad \text{true} \\
\end{align*}
\]

informal policy

within the next 2 time-units both \text{"enter\"} and \text{"exit\"} must happen and \text{"enter\"} must happen before \text{"exit\"}.

The MonPoly Monitoring Tool

Implementation Language: OCaml

algorithmic ideas:
- state update via dynamic programming
- future dependencies treated symbolically as variables in Boolean expressions
- different representations of Boolean expressions (explicit and BDD)
- keep only distinct Boolean expressions in memory
- different representations of Boolean expressions
- future dependencies treated symbolically as variables in Boolean expressions
- keep only distinct Boolean expressions in memory
- almost event-rate independent memory consumption
  (almost = logarithmic in the event-rate; in practice: constant)

Evaluation

- formula size
- event rate in events/s
- avg of 10 random formulas

Features

algorithmic ideas:
- translation of temporal operators into incrementally updated auxiliary first-order predicates
- efficient sliding window algorithm
- waiting queue for future dependencies

- fast
- at least one order of magnitude slower
- negation can occur freely
- incomparable to MFOTL (propositional but regex)

- fast
- at least one order of magnitude slower
- negation can occur freely
- incomparable to MFOTL (propositional but regex)

Industrial Case Studies

example policy: The synchronization scripts must run for at least 1 second and for no longer than 6 hours.

example policy: Long-running SSH sessions must not last longer than 24 hours.

Google

Google monitor usage-control policies in a network of 35000 computers used both within Google's corporate network and externally.

example policy: Long-running SSH sessions must not last longer than 24 hours.

Nokia

Nokia monitor authentication policies in a network of 35000 computers used both within Google's corporate network and externally.

example policy: Long-running SSH sessions must not last longer than 24 hours.

Eugene Zalinescu

Technical Universität München

Felix Klaedtke

NEC

David Basin

Srdan Krstić

ETH Zürich

Dmitrii Traytel

75 Big Data
National Research Programme

Aerial: Almost Event-Rate Independent Algorithms for Monitoring Metric Regular Properties

https://bitbucket.org/traytel/aerial

https://sourceforge.net/projects/monpoly/