Selective Monitoring

Radu Grigore  Stefan Kiefer

Concur 2018
Beijing, 4 September 2018
We are interested in safety specs only.

Some pairs (system, spec) are diagnosable, some are not.
Diagnosability is PSPACE-complete

Theorem (cf. Bertrand, Haddad, Lefaucheux, 2014)

*Diagnosability is PSPACE-complete.*

Proof sketch. Reduce from universality of NFA where all states are initial and accepting.

### Diagram

- **Initial State:** $1\#$
- **Final States:** $\frac{1}{2}a$
- **Transitions:**
  - $1\# \xrightarrow{\frac{1}{2}\#} Q$
  - $Q \xrightarrow{\frac{1}{2}|Q|} a$
  - $a \xrightarrow{\frac{1}{2}a} \frac{1}{2}a$
  - $\frac{1}{2}a \xrightarrow{\frac{1}{2}b} \frac{1}{2}b$
  - $\frac{1}{2}b \xrightarrow{\frac{1}{2}a, \frac{1}{2}b, \#} Q$
  - $Q \xrightarrow{\#} a, b$
  - $a, b \xrightarrow{\#} a, b, \#$
Selective monitoring

We don’t insist on diagnosability.

A (selective) monitor is feasible if the probability of giving a verdict is as high as for the monitor that observes everything.
We don’t insist on diagnosability.

A (selective) monitor is feasible if the probability of giving a verdict is as high as for the monitor that observes everything.

Consider observation prefix $a \perp a$
We don’t insist on diagnosability.

A (selective) monitor is **feasible** if the probability of giving a verdict is as high as for the monitor that observes everything.

Consider observation prefix $a \perp a$
Selective monitoring

We don’t insist on diagnosability.

A (selective) monitor is feasible if the probability of giving a verdict is as high as for the monitor that observes everything.

Consider observation prefix $a \perp a$
Cost of a monitor

\[ C_\rho := \text{number of observations that } \rho \text{ makes (random var.)} \]

\[ c_{\inf} := \inf_{\text{feasible } \rho} \mathbb{E}[C_\rho] \]

**Proposition**

*If* (system, spec) *is diagnosable then* \( c_{\inf} < \infty \).

Proof sketch. Eagerly observe everything until a verdict can be given. Then stop observing.

Converse doesn’t hold.

**Theorem**

*It is PSPACE-complete to check whether* \( c_{\inf} < \infty \).

Proof similar to PSPACE-completeness of diagnosability.
\[ C_\rho := \text{number of observations that } \rho \text{ makes} \] (random var.)

\[ c_{\inf} := \inf_{\text{feasible } \rho} \mathbb{E}[C_\rho] \]

**Theorem**

*It is undecidable to check whether \( c_{\inf} < 3 \).*

Proof sketch. Reduce from the problem whether a given probabilistic automaton accepts some word with prob \( > \frac{1}{2} \).

Hard to get right.
Cost of a monitor

\[ C_\rho := \text{number of observations that } \rho \text{ makes (random var.)} \]

\[ c_{\inf} := \inf_{\text{feasible } \rho} E[C_\rho] \]

**Theorem**

*It is undecidable to check whether \( c_{\inf} < 3 \).*

Proof sketch. Reduce from the problem whether a given probabilistic automaton accepts some word with prob \( \frac{1}{2} \).

Hard to get right.

“Computing an optimal monitor” is also hard.
Proposition

In the non-hidden case we always have diagnosability.

Proof sketch. Observe everything and follow along in the DFA until a bottom SCC of the product has been reached.

Key Observation

In the non-hidden case, maximum procrastination is optimal.
Proposition

In the non-hidden case we always have diagnosability.

Proof sketch. Observe everything and follow along in the DFA until a bottom SCC of the product has been reached.

Key Observation

In the non-hidden case, maximum procrastination is optimal.
Proposition
In the non-hidden case we always have diagnosability.

Proof sketch. Observe everything and follow along in the DFA until a bottom SCC of the product has been reached.

Key Observation
In the non-hidden case, maximum procrastination is optimal.

Radu Grigore, Stefan Kiefer
Selective Monitoring
Proposition

In the non-hidden case we always have diagnosability.

Proof sketch. Observe everything and follow along in the DFA until a bottom SCC of the product has been reached.
Proposition

*In the non-hidden case we always have diagnosability.*

Proof sketch. Observe everything and follow along in the DFA until a bottom SCC of the product has been reached.

Key Observation

*In the non-hidden case, maximum procrastination is optimal.*
The optimal monitor acts as follows:

1. Compute $k$, the minimum number of observations such that skipping $k$ observations leads to confusion.
2. Skip $k−1$ observations, and then make 1 observation.
The optimal monitor acts as follows:

1. Compute $k$, the minimum number of observations such that skipping $k$ observations leads to confusion.
2. Skip $k-1$ observations, and then make 1 observation.
Non-Hidden Case

The optimal monitor acts as follows:

1. Compute $k$, the minimum number of observations such that skipping $k$ observations leads to confusion.
2. Skip $k-1$ observations, and then make 1 observation.

Here $k = \infty$. So, choose $k$ very large.
At every stage the monitor has a belief \( \{(s_1, q_1), \ldots, (s_m, q_m)\} \) about where the product MC \( \times \) DFA is.

We might have \( m > 1 \) but all \( (s_i, q_i) \) in the belief must be language equivalent in a certain DFA.

To compute

\[
c_{\text{inf}} := \inf_{\text{feasible } \rho} \mathbb{E}[C_\rho]
\]

one can set up and solve a small linear equation system. (A belief with \( k = \infty \) has an expected cost of 1.)

**Theorem**

*In the non-hidden case one can compute \( c_{\text{inf}} \) in polynomial time.*
We have shown: maximal procrastination is optimal. How much better is maximal procrastination than the baseline?

We took 11 open-source Java projects among those most forked on GitHub, totaling 80,000 Java methods.

- On each, we ran the Facebook Infer static analyzer to compute a symbolic flowgraph (SFG) skeleton for MC
- For each MC skeleton we sampled transition probabilities from Dirichlet distributions. (The optimal monitor is independent of those transition probabilities.)
- We considered a fixed safety property about iterators.
- In >90% of cases the optimal monitor is trivial and $\mathbb{E}[C_\rho] = 0$, because Infer decides the property statically.
- On the remaining methods we computed $c_{inf}$ using Gurobi.
- Our implementation is in a fork of Infer, on GitHub.
# Experiments

<table>
<thead>
<tr>
<th>Name</th>
<th>Methods</th>
<th>SFGs</th>
<th>LOC</th>
<th>Count</th>
<th>Avg-Size</th>
<th>Max-Size</th>
<th>Med</th>
<th>GAvg</th>
</tr>
</thead>
<tbody>
<tr>
<td>tomcat</td>
<td>26K</td>
<td>52K</td>
<td>946K</td>
<td>343</td>
<td>69</td>
<td>304</td>
<td>0.53</td>
<td>0.50</td>
</tr>
<tr>
<td>okhttp</td>
<td>3K</td>
<td>6K</td>
<td>49K</td>
<td>110</td>
<td>263</td>
<td>842</td>
<td>0.46</td>
<td>0.42</td>
</tr>
<tr>
<td>dubbo</td>
<td>8K</td>
<td>16K</td>
<td>176K</td>
<td>91</td>
<td>111</td>
<td>385</td>
<td>0.53</td>
<td>0.51</td>
</tr>
<tr>
<td>jadex</td>
<td>4K</td>
<td>9K</td>
<td>48K</td>
<td>204</td>
<td>96</td>
<td>615</td>
<td>0.58</td>
<td>0.50</td>
</tr>
<tr>
<td>RxJava</td>
<td>12K</td>
<td>45K</td>
<td>192K</td>
<td>83</td>
<td>41</td>
<td>285</td>
<td>0.52</td>
<td>0.53</td>
</tr>
<tr>
<td>guava</td>
<td>22K</td>
<td>43K</td>
<td>1218K</td>
<td>1126</td>
<td>134</td>
<td>926</td>
<td>0.41</td>
<td>0.41</td>
</tr>
<tr>
<td>clojure</td>
<td>5K</td>
<td>19K</td>
<td>66K</td>
<td>219</td>
<td>120</td>
<td>767</td>
<td>0.44</td>
<td>0.44</td>
</tr>
<tr>
<td>AndroidUtilCode</td>
<td>3K</td>
<td>7K</td>
<td>436K</td>
<td>39</td>
<td>89</td>
<td>288</td>
<td>0.66</td>
<td>0.58</td>
</tr>
<tr>
<td>leakcanary</td>
<td>1K</td>
<td>1K</td>
<td>11K</td>
<td>12</td>
<td>79</td>
<td>268</td>
<td>0.66</td>
<td>0.59</td>
</tr>
<tr>
<td>deeplearning4j</td>
<td>21K</td>
<td>40K</td>
<td>408K</td>
<td>262</td>
<td>51</td>
<td>341</td>
<td>0.58</td>
<td>0.58</td>
</tr>
<tr>
<td>fastjson</td>
<td>2K</td>
<td>7K</td>
<td>47K</td>
<td>204</td>
<td>63</td>
<td>597</td>
<td>0.59</td>
<td>0.53</td>
</tr>
</tbody>
</table>
Empirical distribution of $\frac{c_{inf}}{E[C_{base}]}$, across all projects.
Can faults in a given system be diagnosed?
- diagnosability; originally for finite non-stochastic systems [SSLST, 1995]
- polynomial-time, but exponentially-sized monitors

Diagnosability in stochastic systems (labelled MCs)
- since [Thorsley, Teneketzis, 2005]
- many different notions of diagnosability
- most of them PSPACE-complete [Bertrand, Haddad, Lefaucheux, 2014]

Selective monitoring
- best-effort monitoring with a specified overhead budget, e.g., [Arnold, Vechev, Yahav, 2008]
- RVSE [SBSGHSZ, 2011] also computes a probability that the program run is faulty
- our approach is opposite: no compromises on precision