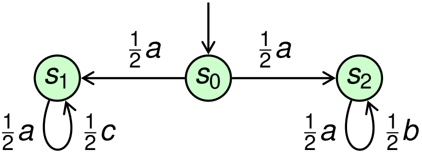
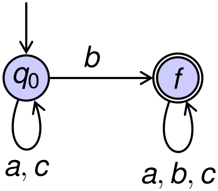
# Selective Monitoring

Radu Grigore     Stefan Kiefer

Concur 2018
Beijing, 4 September 2018

# Labelled Markov Chains and DFAs




We are interested in safety specs only.

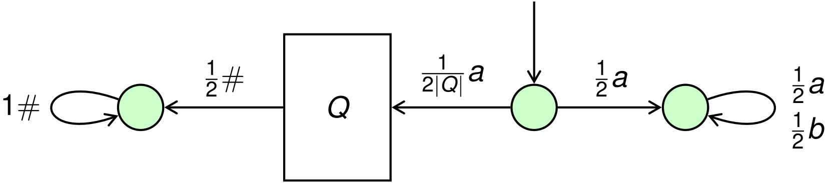
Some pairs (system, spec) are diagnosable, some are not.

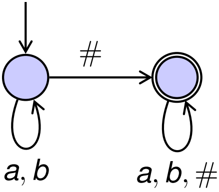
# Diagnosability is PSPACE-complete

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

*Diagnosability is PSPACE-complete.*

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

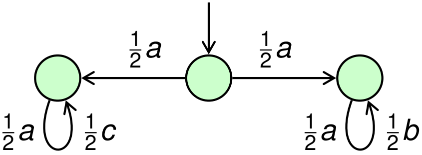


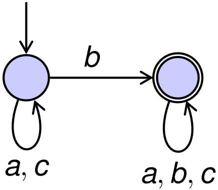



# Selective monitoring

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.

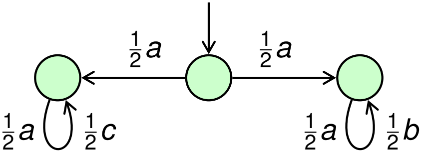


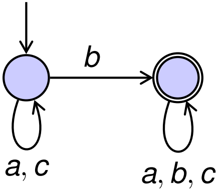



# Selective monitoring

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.



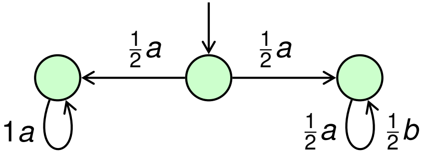



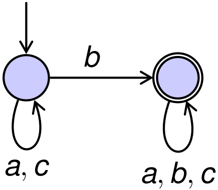

Consider observation prefix $a \bot a$

# 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.



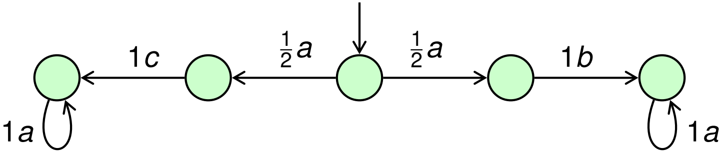



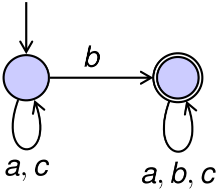

Consider observation prefix $a \perp a$

# 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 $a \bot a$

# 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.

# Cost of a monitor

$$C_\rho := \text{number of observations that } \rho \text{ makes (random var.)}$$
$$c_{\mathit{inf}} := \inf_{\text{feasible } \rho} \mathbb{E}[C_\rho]$$

**Theorem**

*It is undecidable to check whether $c_{\mathit{inf}} < 3$.*

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

Hard to get right. $\square$

# Cost of a monitor

$$C_\rho := \text{number of observations that } \rho \text{ makes (random var.)}$$
$$c_{inf} := \inf_{\text{feasible } \rho} \mathbb{E}[C_\rho]$$

**Theorem**

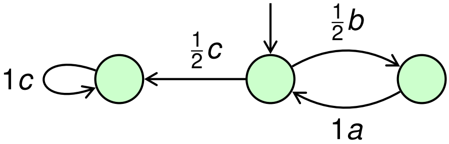
*It is undecidable to check whether $c_{inf} < 3$.*

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

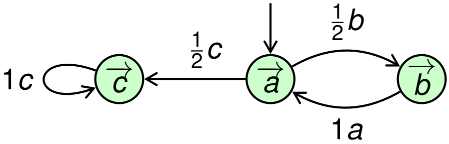
Hard to get right. $\square$

"Computing an optimal monitor" is also hard.

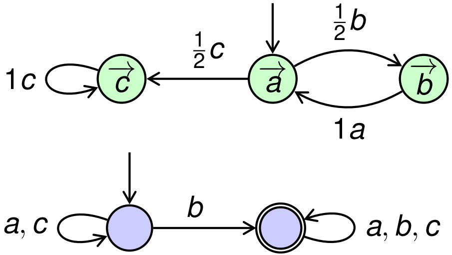
# Non-Hidden Markov Chains



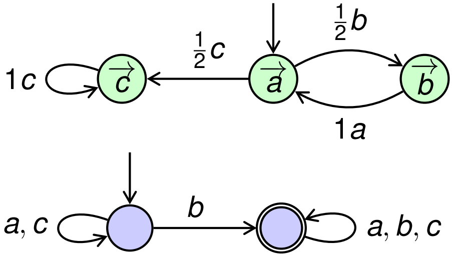
# Non-Hidden Markov Chains



# Non-Hidden Markov Chains



# Non-Hidden Markov Chains

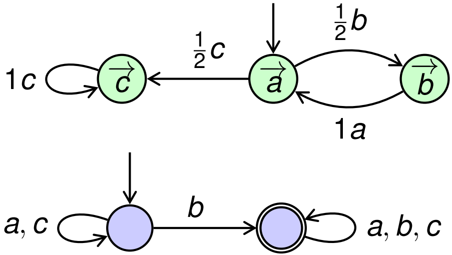



**Proposition**

*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. □

# Non-Hidden Markov Chains




**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.*

# Non-Hidden Case

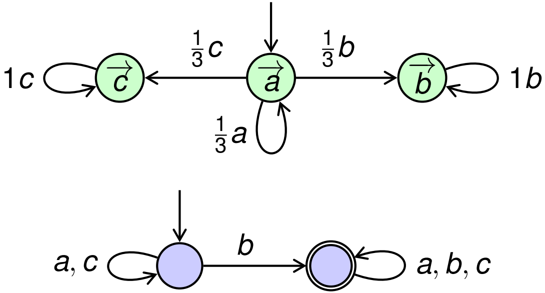
The optimal monitor acts as follows:

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.
3. Goto 1.

# Non-Hidden Case

The optimal monitor acts as follows:

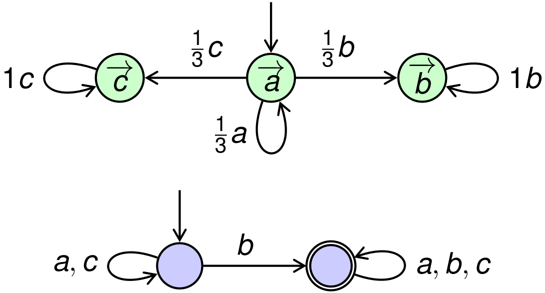
1. Compute $k$, the minimum number of observations such that skipping $k$ observations leads to confusion.
2. Skip $k-1$ observations, and then make 1 observation.
3. Goto 1.



# 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.
3. Goto 1.




Here $k = \infty$. So, choose $k$ very large.

# Non-Hidden Case

At every stage the monitor has a belief $\{(s_1, q_1), \dots, (s_m, q_m)\}$ 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

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

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 $c_{inf}$ in polynomial time.*

# Experiments

We have shown: maximal procrastination is optimal.
How much better is maximal procrastination than the baseline?

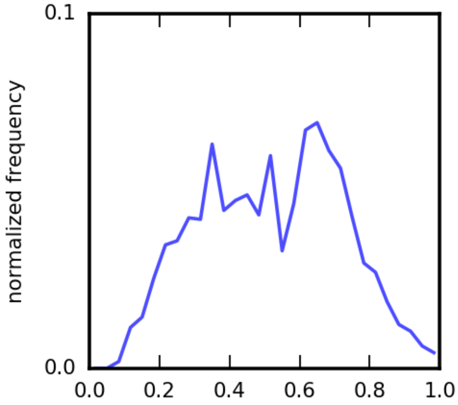
We took 11 open-source Java projects among those most forked on GitHub, totaling 80,000 Java methods.

- On each, we ran the Facebook Infer static analyzer to compute a symbolic flowgraph (SFG) $\longrightarrow$ 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

| Name | Project Size | | | Monitors | | | $\frac{c_{inf}}{\mathbb{E}[C_{base}]}$ | |
|---|---|---|---|---|---|---|---|---|
| | Methods | SFGs | LOC | Count | Avg-Size | Max-Size | Med | GAvg |
| tomcat | 26K | 52K | 946K | 343 | 69 | 304 | 0.53 | 0.50 |
| okhttp | 3K | 6K | 49K | 110 | 263 | 842 | 0.46 | 0.42 |
| dubbo | 8K | 16K | 176K | 91 | 111 | 385 | 0.53 | 0.51 |
| jadx | 4K | 9K | 48K | 204 | 96 | 615 | 0.58 | 0.50 |
| RxJava | 12K | 45K | 192K | 83 | 41 | 285 | 0.52 | 0.53 |
| guava | 22K | 43K | 1218K | 1126 | 134 | 926 | 0.41 | 0.41 |
| clojure | 5K | 19K | 66K | 219 | 120 | 767 | 0.44 | 0.44 |
| AndroidUtilCode | 3K | 7K | 436K | 39 | 89 | 288 | 0.66 | 0.58 |
| leakcanary | 1K | 1K | 11K | 12 | 79 | 268 | 0.66 | 0.59 |
| deeplearning4j | 21K | 40K | 408K | 262 | 51 | 341 | 0.58 | 0.58 |
| fastjson | 2K | 7K | 47K | 204 | 63 | 597 | 0.59 | 0.53 |

# Experiments




Empirical distribution of $\frac{c_{inf}}{\mathbb{E}[C_{base}]}$, across all projects.

# Related Work

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