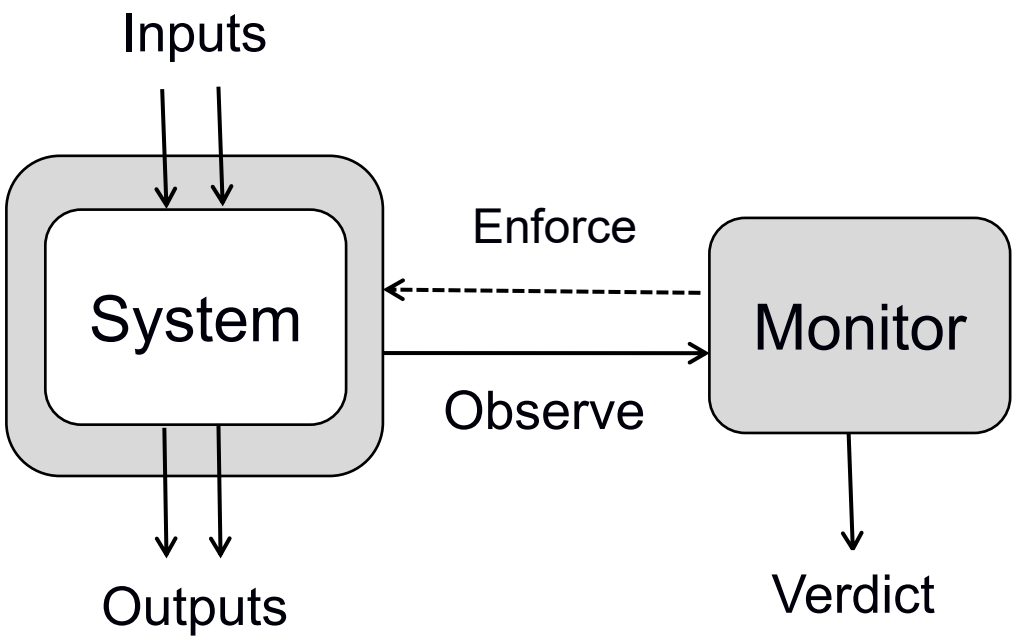
Monitorability under Assumptions

N. Ege Saraç
IST Austria

joint work with Tom Henzinger
RV 2020

Online Black-Box Monitoring



Bounded number of monitor operations
per system operation (“real time” [Rabin63]).

Summary

1. Assumptions in monitoring
2. Register monitors (with Thomas Ferrère and Tom Henzinger)

Two Trends that Favor Runtime Verification

1. Ever increasing hardware parallelism
   - □ many-core processors, computer clusters
   - □ not all hardware resources will go into performance
     (some will hopefully go into assurance)
2. Ever increasing software complexity
   - □ machine-learned components, cloud connectivity
   - □ the static “verification gap” will increase
     (software complexity grows faster than verification capability)

Two Trends that Favor Runtime Verification

1. Ever increasing hardware parallelism
   - □ many-core processors, computer clusters
   - □ not all hardware resources will go into performance
     (some will hopefully go into assurance)
2. Ever increasing software complexity
   - □ machine-learned components, cloud connectivity
   - □ the static “verification gap” will increase
     (software complexity grows faster than verification capability)

Two Observations about Runtime Verification

1. Static verification is about emptiness,
runtime verification is about membership

- formalisms and methods need to reflect this
- opening for quantitative and approximate methods
  (distance measures between behaviors rather than systems)

2. To increase acceptance and trust of monitors,
unintrusive online third-party approaches are essential

- unintrusive means black-box and low overhead
- online means real-time and best-effort
- third-party means that system and monitor are designed independently

Two Observations about Runtime Verification

1. Static verification is about emptiness,
runtime verification is about membership

- formalisms and methods need to reflect this
- opening for quantitative and approximate methods
  (distance measures between behaviors rather than systems)

2. To increase acceptance and trust of monitors,
unintrusive online third-party approaches are essential

- unintrusive means black-box and low overhead
- online means real-time and best-effort
- third-party means that system and monitor are designed independently

Two Observations about Runtime Verification

1. Static verification is about emptiness,
runtime verification is about membership

- formalisms and methods need to reflect this
- opening for quantitative and approximate methods
  (distance measures between behaviors rather than systems)

2. To increase acceptance and trust of monitors,
unintrusive online third-party approaches are essential

- unintrusive means black-box and low overhead
- online means real-time and best-effort
- third-party means that system and monitor are designed independently

Response Property

Observations	$\Sigma = \{a, b, t, o\}$
Behaviors	Σ^{ω}
Response	$\Box(a \rightarrow \Diamond b)$

o t t o a t o t t t b t o o t t o t a t o t t o a o t o o t t t o o b o t o t o t o t o t ...

Response is live (every finite behavior can be extended to satisfy response).
Response is co-live (every finite behavior can be extended to violate response).

Response is not monitorable in any sense of the word
(no finite behavior allows a positive or negative verdict).

Response Property

Observations	$\Sigma = \{a, b, t, o\}$
Behaviors	Σ^{ω}
Response	$\Box(a \rightarrow \Diamond b)$

o t t o a t o t t t b t o o t t o t a t o t t o a o t o o t t t o o b o t o t o t o t o t ...

Response is live (every finite behavior can be extended to satisfy response).
Response is co-live (every finite behavior can be extended to violate response).

Response is not monitorable in any sense of the word
(no finite behavior allows a positive or negative verdict).

Bounded Response Property

Observations	$\Sigma = \{a, b, t, o\}$
Behaviors	Σ^{ω}
Bounded response	$\Box(a \rightarrow \Diamond_{\leq 5}\ b)$

$o\ t\ t\ o\ a\ t\ o\ t\ t\ t\ b\ t\ o\ o\ t\ t\ o\ t\ a\ t\ o\ t\ t\ o\ a\ o\ t\ o\ o\ t\ t\ t\ o\ o\ b\ o\ t\ o\ t\ o\ t\ o\ t\ o\ t$ ...

Bounded response is not live (consider the prefix $a\ t\ t\ t\ t\ t\ t$).
Bounded response is not safe (consider the trace $a\ o\ o\ o\ o\ o\ o$ ...).
Bounded response is co-live (consider the extension $a\ t\ t\ t\ t\ t\ t\ \ldots$).

Bounded response is monitorable
(for every finite behavior the continuation $a\ t\ t\ t\ t\ t\ t$ allows a negative verdict).

Bounded Response Property

Observations	$\Sigma = \{a, b, t, o\}$
Assumption	$\Box \Diamond t$
Bounded response	$\Box(a \rightarrow \Diamond_{\leq 5} b)$

o t t o a t o t t t b t o o t t o t a t o t t o a o t o o t t t o o b o t o t o t o t o t ...

Bounded response is safe under the assumption [H92]
(every violating behavior has a finite prefix that allows a negative verdict).

Assumptions make properties “safer” and “more monitorable”.

Assumptions

An assumption $A \subseteq \Sigma^{\omega}$ restricts the universe of possible behaviors.

How do assumptions arise?

1. Knowledge about the system (predictive monitoring)
2. Knowledge about the environment (e.g. $\Box \Diamond t$)
3. Information from other monitors (decentralized monitoring)
   - □ decomposition of monitors
   - □ assume-guarantee monitoring

Monitorability under Assumptions

A finite behavior α allows a positive (negative) verdict for property *P* under assumption *A* if all extensions of α in *A* satisfy (resp. violate) *P*.

The property *P* is monitorable under assumption *A* if for every finite prefix α from *A*, there is a finite continuation β from *A* such that $\alpha\beta$ allows a positive or negative verdict for *P* under *A*.

Note: monitorability under assumption Σ^ω is Pnueli-Zaks monitorability.

An Assumption for Response

Observations	$\Sigma = \{a, b, o\}$
Assumption *A*	$\Box(a \wedge \ominus((\neg b) S a) \rightarrow \Box \neg b)$
Response *R*	$\Box(a \rightarrow \Diamond b)$

Response *R* is not monitorable.

Response *R* is monitorable under assumption *A*

(for every finite behavior the continuation aa allows a negative verdict).

What is Monitorable under Assumptions?

The property *P* is safe under assumption *A* if for every behavior in $A \setminus P$ has a finite prefix that allows a negative verdict for *P* under *A*.

The property *P* is co-safe under assumption *A* if for every behavior in $A \cap P$ has a finite prefix that allows a positive verdict for *P* under *A*.

Let *Obl(A)* be the positive boolean combinations of safety and co-safety properties under assumption *A*.
Let *Mon(A)* be the monitorable properties under *A.*

Theorem (cf. FalconeFernandezMounier12, BauerLeuckerSchallhart11):

1. $Obl(A) \subseteq Mon(A)$.
2. *Mon(A)* is closed under boolean operations.

Cantor Topology on Σ^ω

Cantor metric on behaviors in Σ^ω:

$d(w, w') = 1/2^n$ where n is the length of the longest common prefix of w and w'

Topological operations on properties (and assumptions), i.e., subsets of Σ^ω:

- closure = smallest closed superset
- interior = largest open subset
- boundary = set difference between closure and interior

AlpernSchneider85, DiekertLeucker14:

- safe = closed
- co-safe = open
- live = dense
- monitorable = boundary has empty interior

Cantor Topology on Σ^ω

Cantor metric on behaviors in Σ^ω:

$d(w, w') = 1/2^n$ where n is the length of the longest common prefix of w and w'

Topological operations on properties (and assumptions), i.e., subsets of Σ^ω:

- closure = smallest closed superset
- interior = largest open subset
- boundary = set difference between closure and interior

AlpernSchneider85, DiekertLeucker14:

- safe = closed
- co-safe = open
- live = dense
- monitorable = boundary has empty interior

Topological Construction of Assumptions

Theorem (cf. H92): For every property *P*, there are unique weakest liveness assumptions *A* and *B* such that *P* is safe under *A* and co-safe under *B*.

Proof:

$A = co(closure(P) \backslash P)$
$B = co(P \backslash interior(P))$

Theorem: For every property *P*, there is a co-safety assumption *C* such that $P \in Mon(C)$.

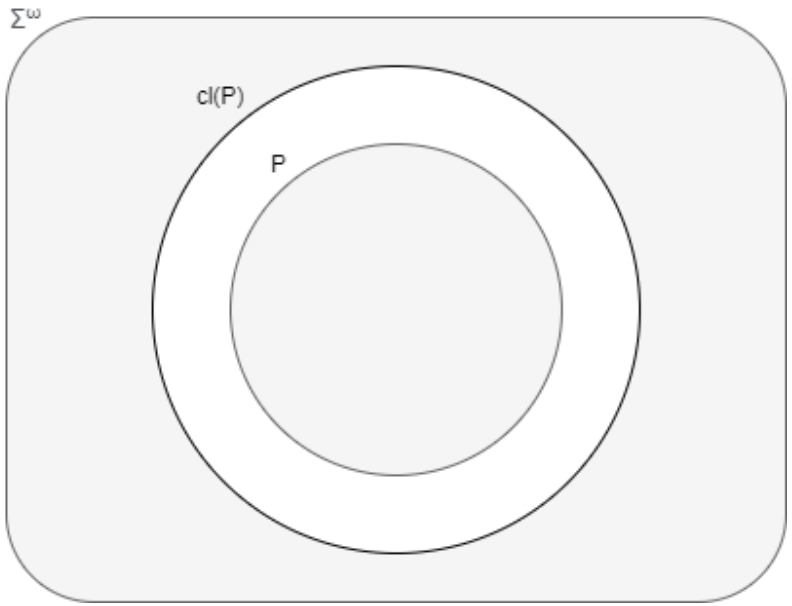
$boundary(P) = (closure(P) \backslash P) \cup (P \backslash interior(P))$
$C = co(closure(interior(boundary(P))))$

Topological Construction of Assumptions

Theorem (cf. H92): For every property *P*, there are unique weakest liveness assumptions *A* and *B* such that *P* is safe under *A* and co-safe under *B*.

Proof:

$A = co(closure(P) \setminus P)$



Topological Construction of Assumptions

Theorem (cf. H92): For every property *P*, there are unique weakest liveness assumptions *A* and *B* such that *P* is safe under *A* and co-safe under *B*.

Proof:

$A = co(closure(P) \backslash P)$
$B = co(P \backslash interior(P))$

Theorem: For every property *P*, there is a co-safety assumption *C* such that $P \in Mon(C)$.

$boundary(P) = (closure(P) \backslash P) \cup (P \backslash interior(P))$
$C = co(closure(interior(boundary(P))))$

Topological Construction of Assumptions

Observations	$\Sigma = \{a, b, c, o\}$
Assumption *A*	$(\neg b)Ua$
Property *P*	$(\Box(a \rightarrow \Diamond b) \vee (\neg b)Ua) \wedge \Diamond c$

P is live (consider the extension $c\ b\ b\ b\ b\ b\ b$ ...).
P is not monitorable (consider the finite behavior b).
P is monitorable under assumption *A* (consider the continuation c).

Topological Construction of Assumptions

Observations	$\Sigma = \{a, b, c, o\}$
Assumption *A*	$(\neg b)Ua$
Property *P*	$(\Box(a \rightarrow \Diamond b) \vee (\neg b)Ua) \wedge \Diamond c$

P is live (consider the extension $c\ b\ b\ b\ b\ b\ b$ ...).
P is not monitorable (consider the finite behavior b).
P is monitorable under assumption *A* (consider the continuation c).

Open Question

What is the structure of the assumptions that make a given property monitorable?

Note: all presented results hold equally for

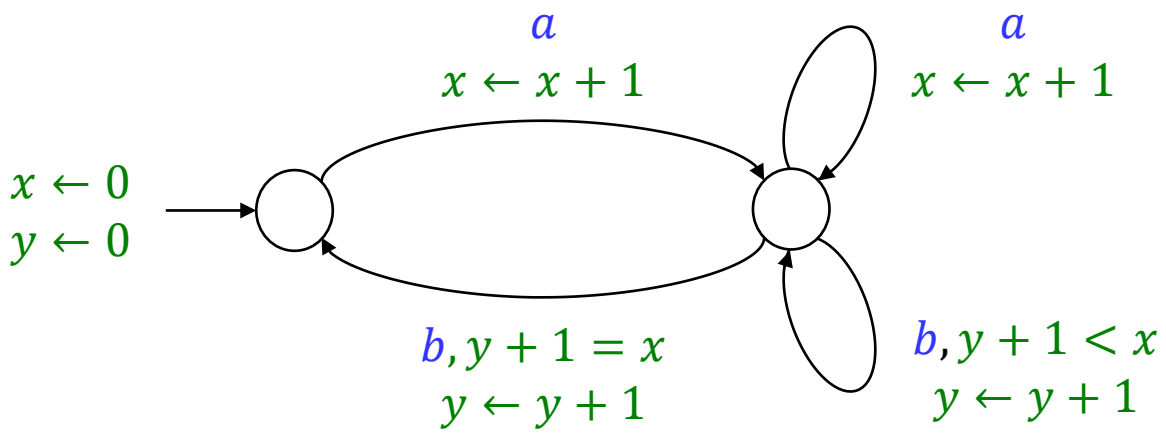
1. arbitrary properties and assumptions
2. ω–regular properties and assumptions
3. LTL properties and assumptions

Safety Monitoring with Registers
or
Assumptions can Save Resources

with Thomas Ferrère and Tom Henzinger
A Theory of Register Monitors
LICS 2018

Counter Safety Monitor

Counters: $(+1, =)$ (equivalently, $(+1, -1, = 0)$)



$$\{w \in \Sigma^{\omega} : \forall \pi \prec w : \#(a, \pi) \geq \#(b, \pi)\}$$

Expressiveness of Register Monitors

$\Sigma_k = \{0, \ldots, k\}$
$P_k = \{w \in \Sigma_k^\omega : \forall \pi \prec w : \forall i < k : \#(i+1, \pi) \geq \#(i, \pi)\}$

Theorem:

1. For all k, the property P_k can be real-time monitored with $k+1$ counters but not with k counters.
2. Any number of counters can be simulated in real time by four registers with $(+, -, \geq)$.

Open:

Does the hierarchy collapse for safety monitors with addition?
How about safety monitors with multiplication?

Counter Assumptions

$P_k = P_i \cap A_{i,k}$ for $i \leq k$

$A_{i,k} = \{w \in \Sigma_k^\omega : \forall \pi \prec w : \forall i \leq j < k : \#(j+1, \pi) \geq \#(j, \pi)\}$

Theorem:

For all k, the property P_k can be real-time monitored with i counters under assumption $A_{i,k}$ (which can be monitored with $k - i$ counters).

Summary

1. Assumptions can make properties monitorable and save monitor resources.

2. The expressiveness of online counter monitors increases with the number of registers.

Thank you!