Towards A ~~Hybrid~~ (Combined?) Approach to Software Verification

Dario Della Monica[1]   Adrian Francalanza[2]

[1]**ICE-TCS, School of Computer Science, Reykjavik University, Iceland**
`dariodm@ru.is`

[2]**University of Malta, Malta**
`adrian.francalanza@um.edu.mt`

NWPT 2015
Reykjavik, 23 October 2015

Outline

Introduction: model checking vs. runtime verification (motivations)

Runtime verification for μHML (= μ-calculus)

Extending runtime verification applicability: ~~hybrid~~ (combined?) approach

Outline

Introduction: model checking vs. runtime verification (motivations)

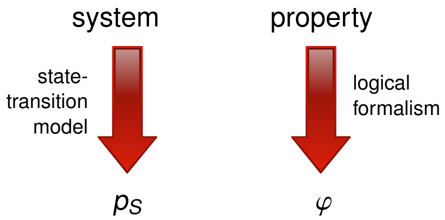
Runtime verification for μHML (= μ-calculus)

Extending runtime verification applicability: ~~hybrid~~ (combined?) approach

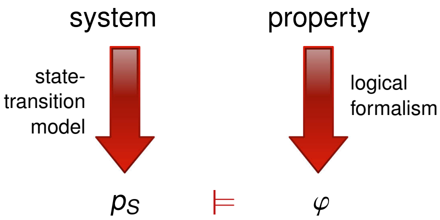
Model checking

system property

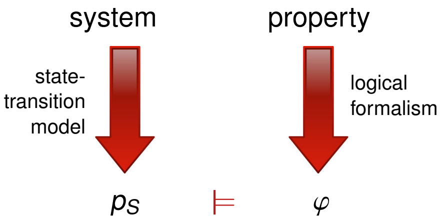
Model checking



Model checking



Model checking



unfeasible for most real-world applications
(state explosion problem)

Runtime verification

monitoring a single partial execution and try to give a verdict

eventually p

Runtime verification

monitoring a single partial execution and try to give a verdict

eventually p

Runtime verification

monitoring a single partial execution and try to give a verdict

eventually p

$\neg p$

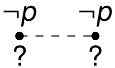
•

?

Runtime verification

monitoring a single partial execution and try to give a verdict

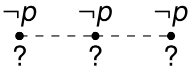
eventually p



Runtime verification

monitoring a single partial execution and try to give a verdict

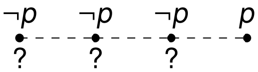
eventually p



Runtime verification

monitoring a single partial execution and try to give a verdict

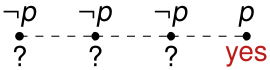
eventually p



Runtime verification

monitoring a single partial execution and try to give a verdict

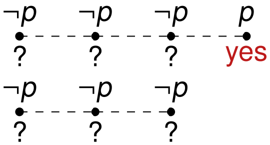
eventually p



Runtime verification

monitoring a single partial execution and try to give a verdict

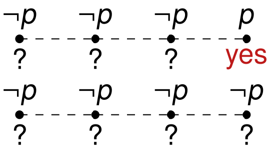
eventually p



Runtime verification

monitoring a single partial execution and try to give a verdict

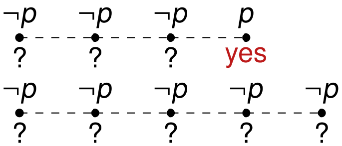
eventually p



Runtime verification

monitoring a single partial execution and try to give a verdict

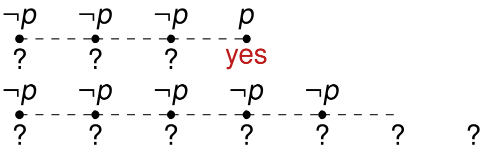
eventually p



Runtime verification

monitoring a single partial execution and try to give a verdict

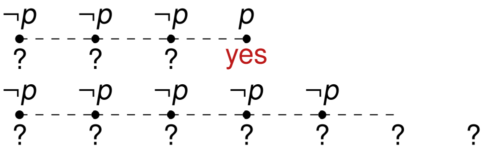
eventually p



Runtime verification

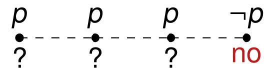
monitoring a single partial execution and try to give a verdict

eventually p



---

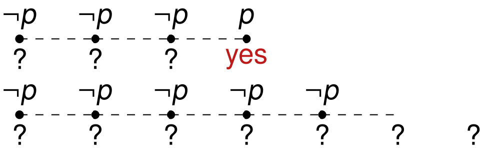
always p



Runtime verification

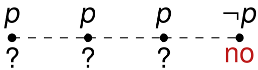
monitoring a single partial execution and try to give a verdict

eventually p



---

always p

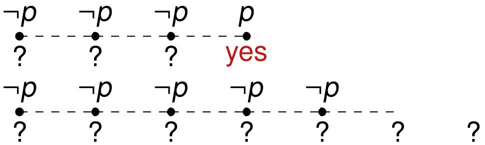


[(p or q) until r]
or (always p)

Runtime verification

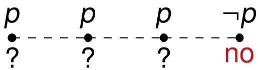
monitoring a single partial execution and try to give a verdict

eventually p

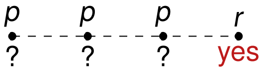


---

always p

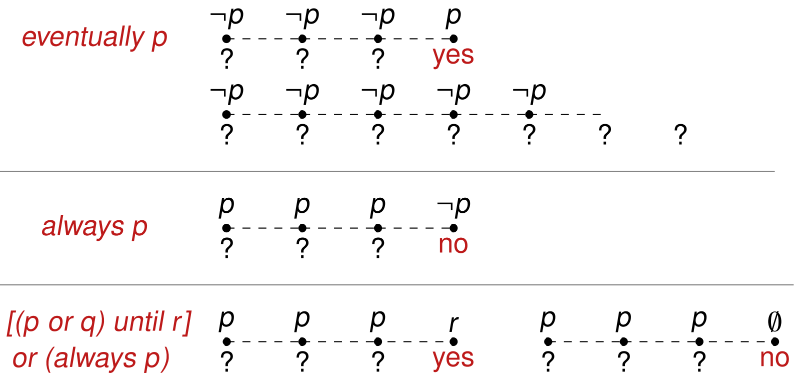


[(p or q) until r] or (always p)



Runtime verification

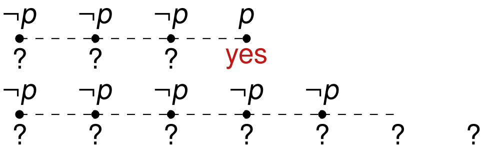
monitoring a single partial execution and try to give a verdict



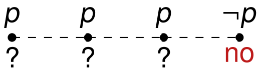
Runtime verification

monitoring a single partial execution and try to give a verdict

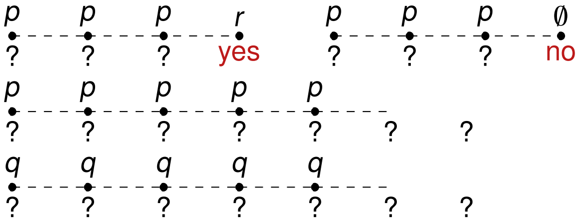
eventually p



always p



[(p or q) until r] or (always p)



Outline

Introduction: model checking vs. runtime verification (motivations)

Runtime verification for μHML (= μ-calculus)

Extending runtime verification applicability: ~~hybrid~~ (combined?) approach

Monitorability

Definition (monitorability)

φ is monitorable := φ is suitable to be runtime verified

:= either

there exists

a finite witness for satisfaction

whenever φ is true

or

there exists

a finite witness for violation

whenever φ is false

The branching time logic μHML

$\varphi, \phi \in \mu\text{HML} ::=$

tt	(truth)	$\mid$ ff	(falsehood)
$\mid \varphi \vee \phi$	(disjunction)	$\mid \varphi \wedge \phi$	(conjunction)
$\mid \langle\alpha\rangle\varphi$	(possibility)	$\mid [\alpha]\varphi$	(necessity)
$\mid \min X.\varphi$	(min. fixpoint)	$\mid \max X.\varphi$	(max. fixpoin)
$\mid X$	(rec. variable)		

The maximal monitorable subset

$$\pi, \varpi \in \text{cHML} ::= \text{tt} \quad \mid \text{ff} \quad \mid \pi \vee \varpi \quad \mid \langle\alpha\rangle\pi \quad \mid \min X.\pi \quad \mid X$$

$$\theta, \vartheta \in \text{sHML} ::= \text{tt} \quad \mid \text{ff} \quad \mid \theta \wedge \vartheta \quad \mid [\alpha]\theta \quad \mid \max X.\theta \quad \mid X$$

Francalanza, Aceto, Ingólfsdóttir, **On verifying Hennessy-Milner logic with recursion at runtime**. In Runtime Verification, 2015

The branching time logic μHML

$\varphi, \phi \in \mu\text{HML} ::=$

tt	(truth)	$\mid \text{ff}$	(falsehood)
$\mid \varphi \vee \phi$	(disjunction)	$\mid \varphi \wedge \phi$	(conjunction)
$\mid \langle \alpha \rangle \varphi$	(possibility)	$\mid [\alpha]\varphi$	(necessity)
$\mid \min X.\varphi$	(min. fixpoint)	$\mid \max X.\varphi$	(max. fixpoin)
$\mid X$	(rec. variable)		

The maximal monitorable subset

$$\pi, \varpi \in \text{cHML} ::= \text{tt} \quad \mid \text{ff} \quad \mid \pi \vee \varpi \quad \mid \langle \alpha \rangle \pi \quad \mid \min X.\pi \quad \mid X$$

$$\theta, \vartheta \in \text{sHML} ::= \text{tt} \quad \mid \text{ff} \quad \mid \theta \wedge \vartheta \quad \mid [\alpha]\theta \quad \mid \max X.\theta \quad \mid X$$

Francalanza, Aceto, Ingólfsdóttir, **On verifying Hennessy-Milner logic with recursion at runtime**. In Runtime Verification, 2015

Monitorability: examples

The maximal monitorable subset

$$\pi, \varpi \in \text{cHML} ::= \text{tt} \quad | \ \text{ff} \quad | \ \pi \vee \varpi \quad | \ \langle \alpha \rangle \pi \quad | \ \min X.\pi \quad | \ X$$
$$\theta, \vartheta \in \text{sHML} ::= \text{tt} \quad | \ \text{ff} \quad | \ \theta \wedge \vartheta \quad | \ [\alpha]\theta \quad | \ \max X.\theta \quad | \ X$$

Examples

$\langle a \rangle$tt

Monitorability: examples

The maximal monitorable subset

$$\pi, \varpi \in \text{cHML} ::= \text{tt} \quad | \ \text{ff} \quad | \ \pi \vee \varpi \quad | \ \langle \alpha \rangle \pi \quad | \ \min X.\pi \quad | \ X$$

$$\theta, \vartheta \in \text{sHML} ::= \text{tt} \quad | \ \text{ff} \quad | \ \theta \wedge \vartheta \quad | \ [\alpha]\theta \quad | \ \max X.\theta \quad | \ X$$

Examples

$\langle a \rangle \text{tt}$ $\qquad$ $[a]\text{ff}$

Monitorability: examples

The maximal monitorable subset

$$\pi, \varpi \in \text{cHML} ::= \text{tt} \quad | \text{ ff} \quad | \; \pi \vee \varpi \quad | \; \langle \alpha \rangle \pi \quad | \text{ min } X.\pi \quad | \; X$$

$$\theta, \vartheta \in \text{sHML} ::= \text{tt} \quad | \text{ ff} \quad | \; \theta \wedge \vartheta \quad | \; [\alpha]\theta \quad | \text{ max } X.\theta \quad | \; X$$

Examples

$\langle a \rangle \text{tt}$ $\qquad$ $[a]\text{ff}$ $\qquad$ $\langle a \rangle \text{tt} \vee \langle b \rangle \text{tt}$

Monitorability: examples

The maximal monitorable subset

$$\pi, \varpi \in \text{cHML} ::= \text{tt} \quad | \ \text{ff} \quad | \ \pi \vee \varpi \quad | \ \langle \alpha \rangle \pi \quad | \ \min X.\pi \quad | \ X$$

$$\theta, \vartheta \in \text{sHML} ::= \text{tt} \quad | \ \text{ff} \quad | \ \theta \wedge \vartheta \quad | \ [\alpha]\theta \quad | \ \max X.\theta \quad | \ X$$

Examples

$\langle a \rangle \text{tt}$ $\qquad$ $[a]\text{ff}$ $\qquad$ $\langle a \rangle \text{tt} \vee \langle b \rangle \text{tt}$ $\qquad$ $\langle a \rangle (\langle b \rangle \text{tt} \vee [c]\text{ff})$

Monitorability: examples

The maximal monitorable subset

$$\pi, \varpi \in \mathsf{cHML} ::= \mathsf{tt} \quad | \ \mathsf{ff} \quad | \ \pi \vee \varpi \quad | \ \langle\alpha\rangle\pi \quad | \ \min X.\pi \quad | \ X$$

$$\theta, \vartheta \in \mathsf{sHML} ::= \mathsf{tt} \quad | \ \mathsf{ff} \quad | \ \theta \wedge \vartheta \quad | \ [\alpha]\theta \quad | \ \max X.\theta \quad | \ X$$

Examples

$\langle a\rangle\mathsf{tt}$ $\quad$ $[a]\mathsf{ff}$ $\quad$ $\langle a\rangle\mathsf{tt} \vee \langle b\rangle\mathsf{tt}$ $\quad$ ~~$\langle a\rangle(\langle b\rangle\mathsf{tt} \vee [c]\mathsf{ff})$~~

Outline

Introduction: model checking vs. runtime verification (motivations)

Runtime verification for μHML (= μ-calculus)

Extending runtime verification applicability: ~~hybrid~~ (combined?) approach

Where runtime verification can reach so far

$\langle a \rangle(\langle b \rangle \mathrm{tt} \vee [c]\mathrm{ff})$: not monitorable, do model checking

Where runtime verification can reach so far

$\langle a \rangle(\langle b \rangle \mathrm{tt} \vee [c]\mathrm{ff})$: not monitorable, do model checking

$$\langle a \rangle(\langle b \rangle \mathrm{tt} \vee [c]\mathrm{ff}) \quad \equiv \quad \langle a \rangle \langle b \rangle \mathrm{tt} \ \vee \ \langle a \rangle [c]\mathrm{ff}$$

Where runtime verification can reach so far

$\langle a \rangle(\langle b \rangle \mathrm{tt} \vee [c]\mathrm{ff})$: not monitorable, do model checking

$$\langle a \rangle(\langle b \rangle \mathrm{tt} \vee [c]\mathrm{ff}) \equiv \underline{\langle a \rangle\langle b \rangle \mathrm{tt}} \vee \underline{\langle a \rangle[c]\mathrm{ff}}$$

monitorable:
do runtime verification

not monitorable:
do model checking

An (RV/MC)decomposition

$$\varphi \equiv \varphi_{RV} \overset{\wedge}{\underset{\vee}{}} \varphi_{MC}$$

Universal: $\wedge$, $[\alpha]$, max X

Existential: $\vee$, $\langle\alpha\rangle$, min X

Skip

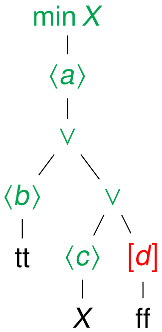
An (RV/MC)decomposition

$$\varphi \equiv \varphi_{RV} \overset{\wedge}{\vee} \varphi_{MC}$$

Universal: $\wedge$, $[\alpha]$, max X

Existential: $\vee$, $\langle\alpha\rangle$, min X

$\min X.\langle a\rangle(\langle b\rangle tt \vee \langle c\rangle X \vee [d]ff)$



An (RV/MC)decomposition

$$\varphi \equiv \varphi_{RV} \overset{\wedge}{\vee} \varphi_{MC}$$

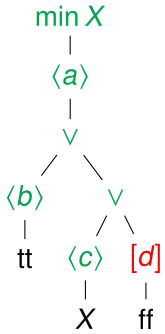
Universal: $\wedge$, $[\alpha]$, max X

Existential: $\vee$, $\langle\alpha\rangle$, min X

$$\min X.\langle a\rangle(\langle b\rangle \mathrm{tt} \vee \langle c\rangle X \vee [d]\mathrm{ff})$$

$$\equiv$$

$$\big(\min X.\langle a\rangle(\langle b\rangle \mathrm{tt} \vee \langle c\rangle X)\big) \vee \boxed{\;?\;}$$



Skip

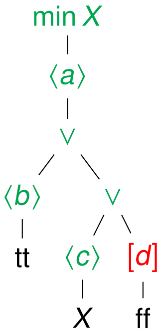
An (RV/MC)decomposition

$$\varphi \equiv \varphi_{RV} \overset{\wedge}{\vee} \varphi_{MC}$$

Universal: $\wedge$, $[\alpha]$, max X

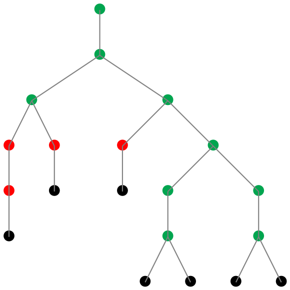
Existential: $\vee$, $\langle\alpha\rangle$, min X

$$\min X.\langle a\rangle(\langle b\rangle tt \vee \langle c\rangle X \vee [d]ff)$$
$$\equiv$$
$$\big(\min X.\langle a\rangle(\langle b\rangle tt \vee \langle c\rangle X)\big) \vee \big(\min X.\langle a\rangle([d]ff \vee \langle c\rangle X)\big)$$



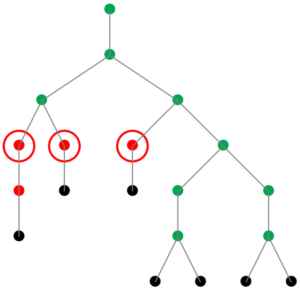
Skip

The decomposition procedure



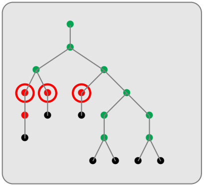
Skip

The decomposition procedure

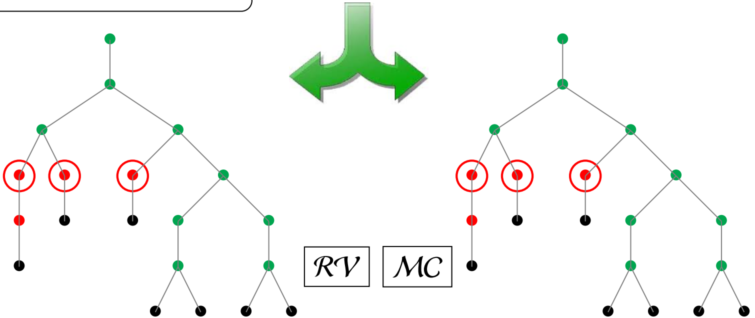


Skip

The decomposition procedure

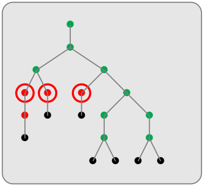


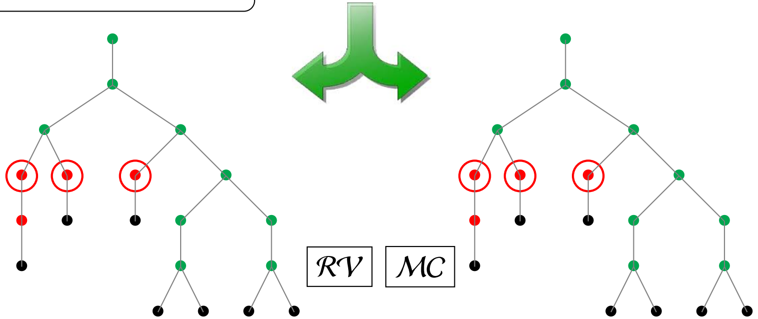
1. Remove subtrees rooted in high univ. nodes
2. Remove new leaves



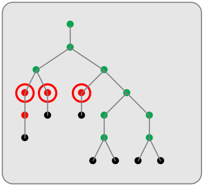
The decomposition procedure

1. Remove subtrees rooted in high univ. nodes
2. Remove new leaves

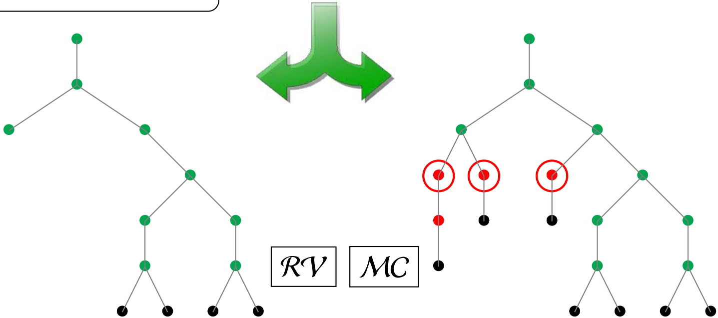




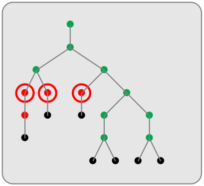
The decomposition procedure



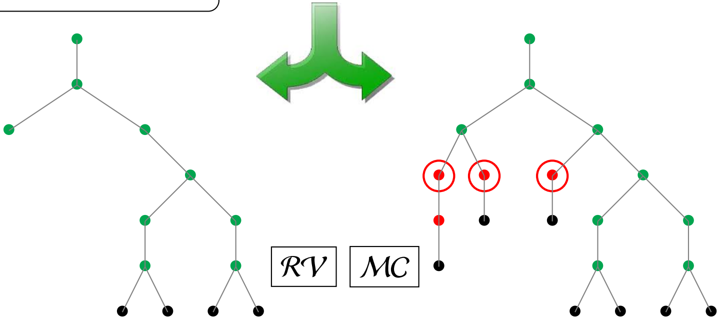
1. Remove subtrees rooted in high univ. nodes
2. Remove new leaves



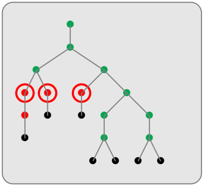
The decomposition procedure



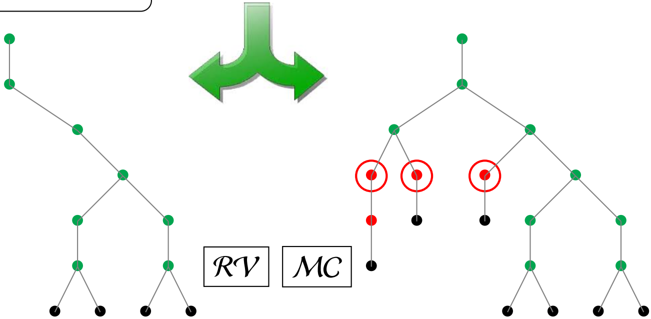
1. Remove subtrees rooted in high univ. nodes
2. Remove new leaves



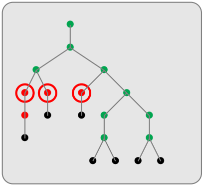
The decomposition procedure



1. Remove subtrees rooted in high univ. nodes
2. Remove new leaves

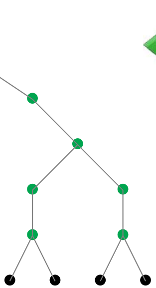


The decomposition procedure

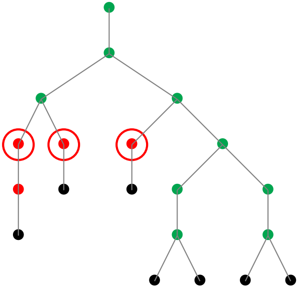


1. Remove subtrees rooted in high univ. nodes
2. Remove new leaves

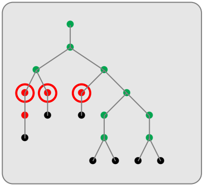
1. Collect subtrees rooted in high univ. nodes
2. Closure under bindings



$\mathcal{RV}$ $\mathcal{MC}$

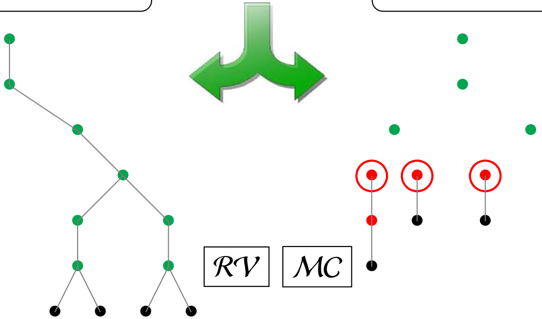


The decomposition procedure

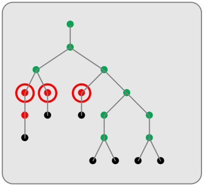


1. Remove subtrees rooted in high univ. nodes
2. Remove new leaves

1. Collect subtrees rooted in high univ. nodes
2. Closure under bindings

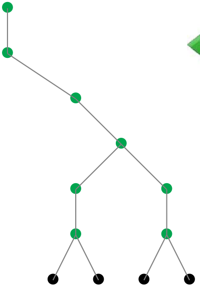


The decomposition procedure

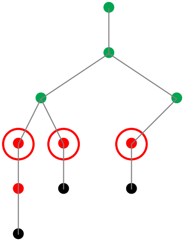


1. Remove subtrees rooted in high univ. nodes
2. Remove new leaves

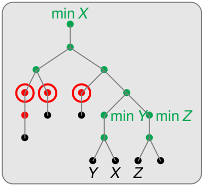
1. Collect subtrees rooted in high univ. nodes
2. Closure under bindings



$\mathcal{RV}$ $\mathcal{MC}$

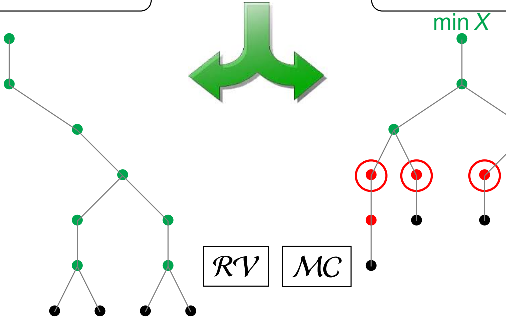


The decomposition procedure



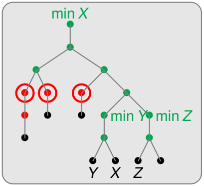
1. Remove subtrees rooted in high univ. nodes
2. Remove new leaves

1. Collect subtrees rooted in high univ. nodes
2. Closure under bindings



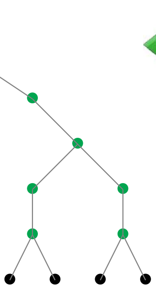
Skip

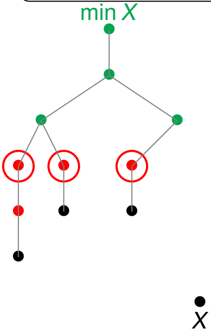
The decomposition procedure



1. Remove subtrees rooted in high univ. nodes
2. Remove new leaves

1. Collect subtrees rooted in high univ. nodes
2. Closure under bindings

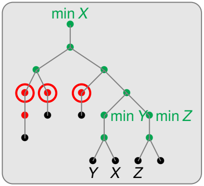




$\mathcal{RV}$ $\mathcal{MC}$

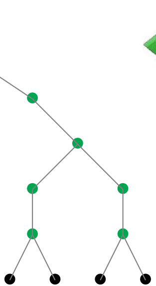
Skip

The decomposition procedure

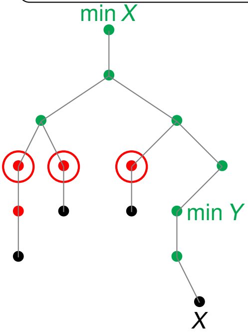


1. Remove subtrees rooted in high univ. nodes
2. Remove new leaves

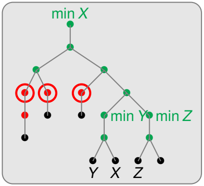
1. Collect subtrees rooted in high univ. nodes
2. Closure under bindings



$\mathcal{RV}$ $\mathcal{MC}$

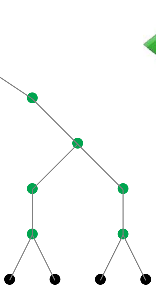


The decomposition procedure

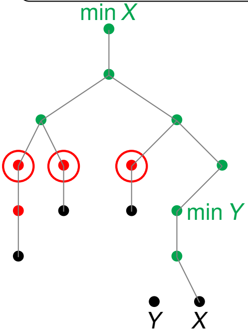


1. Remove subtrees rooted in high univ. nodes
2. Remove new leaves

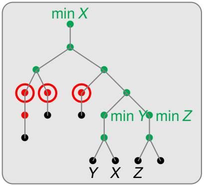
1. Collect subtrees rooted in high univ. nodes
2. Closure under bindings



$\mathcal{RV}$ $\mathcal{MC}$

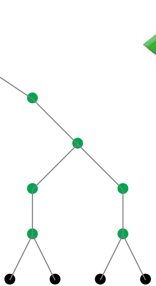


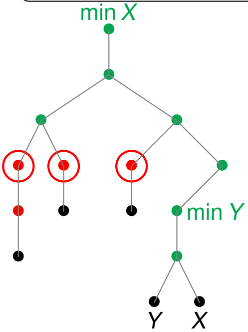
The decomposition procedure



1. Remove subtrees rooted in high univ. nodes
2. Remove new leaves

1. Collect subtrees rooted in high univ. nodes
2. Closure under bindings





$\mathcal{RV}$ $\mathcal{MC}$

Skip

(RV/MC)-decomposition at work

$$\min X.\langle a\rangle(\langle b\rangle \mathrm{tt} \vee \langle c\rangle X \vee [d]\mathrm{ff})$$
$$\equiv$$
$$\Big(\min X.\langle a\rangle(\langle b\rangle \mathrm{tt} \vee \langle c\rangle X)\Big) \vee \Big(\min X.\langle a\rangle([d]\mathrm{ff} \vee \langle c\rangle X)\Big)$$

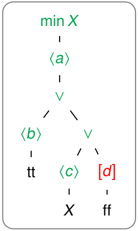
Skip

(RV/MC)-decomposition at work

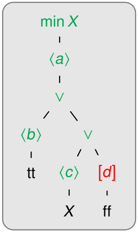
$$\min X.\langle a\rangle(\langle b\rangle tt \vee \langle c\rangle X \vee [d]ff)$$

$$\equiv$$

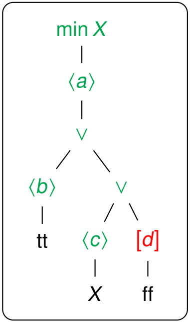
$$\big(\min X.\langle a\rangle(\langle b\rangle tt \vee \langle c\rangle X)\big) \vee \big(\min X.\langle a\rangle([d]ff \vee \langle c\rangle X)\big)$$

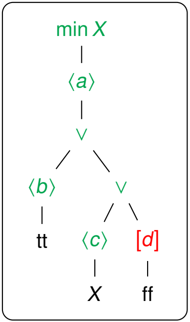


(RV/MC)-decomposition at work

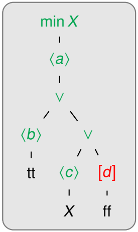


$$\min X.\langle a\rangle(\langle b\rangle tt \vee \langle c\rangle X \vee [d]ff)$$
$$\equiv$$
$$\big(\min X.\langle a\rangle(\langle b\rangle tt \vee \langle c\rangle X)\big) \vee \big(\min X.\langle a\rangle([d]ff \vee \langle c\rangle X)\big)$$

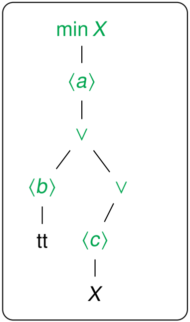


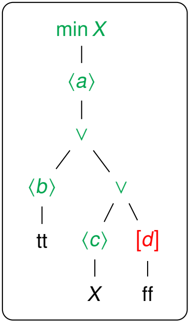


(RV/MC)-decomposition at work



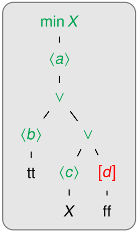
$$\min X.\langle a\rangle(\langle b\rangle\mathrm{tt} \vee \langle c\rangle X \vee [d]\mathrm{ff})$$
$$\equiv$$
$$\Big(\min X.\langle a\rangle(\langle b\rangle\mathrm{tt} \vee \langle c\rangle X)\Big) \vee \Big(\min X.\langle a\rangle([d]\mathrm{ff} \vee \langle c\rangle X)\Big)$$



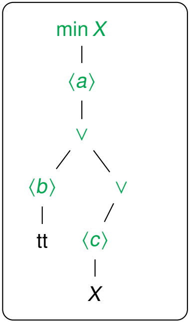


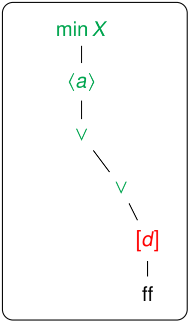
Skip

(RV/MC)-decomposition at work

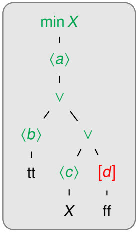


$$\min X.\langle a\rangle(\langle b\rangle tt \vee \langle c\rangle X \vee [d]ff)$$
$$\equiv$$
$$\Big(\min X.\langle a\rangle(\langle b\rangle tt \vee \langle c\rangle X)\Big) \vee \Big(\min X.\langle a\rangle([d]ff \vee \langle c\rangle X)\Big)$$

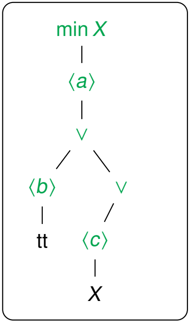


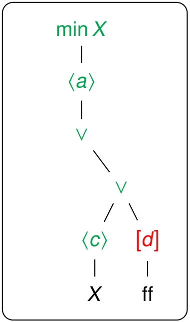


(RV/MC)-decomposition at work

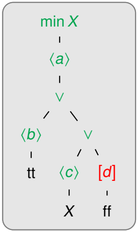


$$\min X.\langle a\rangle(\langle b\rangle tt \vee \langle c\rangle X \vee [d]ff)$$
$$\equiv$$
$$\Big(\min X.\langle a\rangle(\langle b\rangle tt \vee \langle c\rangle X)\Big) \vee \Big(\min X.\langle a\rangle([d]ff \vee \langle c\rangle X)\Big)$$





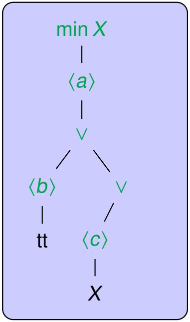
(RV/MC)-decomposition at work

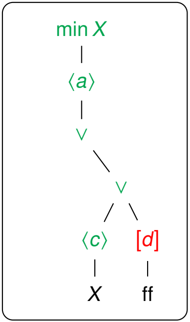


$$\min X.\langle a\rangle(\langle b\rangle\mathrm{tt}\vee\langle c\rangle X\vee[d]\mathrm{ff})$$

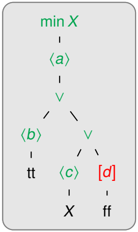
$$\equiv$$

$$\big(\min X.\langle a\rangle(\langle b\rangle\mathrm{tt}\vee\langle c\rangle X)\big)\vee\big(\min X.\langle a\rangle([d]\mathrm{ff}\vee\langle c\rangle X)\big)$$





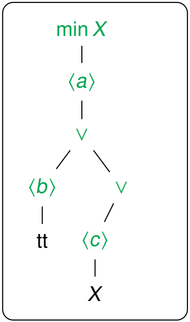
(RV/MC)-decomposition at work

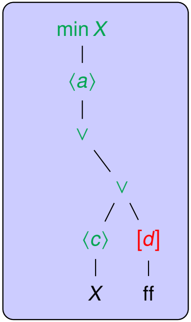


$$\min X.\langle a \rangle(\langle b \rangle tt \vee \langle c \rangle X \vee [d] ff)$$

$$\equiv$$

$$\big(\min X.\langle a \rangle(\langle b \rangle tt \vee \langle c \rangle X)\big) \vee \big(\min X.\langle a \rangle([d] ff \vee \langle c \rangle X)\big)$$





Correctness

Claim (correctness)

- $\varphi \equiv \varphi_{\mathcal{RV}} \vee \varphi_{\mathcal{MC}}$ (existential)
- $\varphi \equiv \varphi_{\mathcal{RV}} \wedge \varphi_{\mathcal{MC}}$ (universal)

Correctness

Claim (correctness)

- $\varphi \equiv \varphi_{\mathcal{RV}} \vee \varphi_{\mathcal{MC}}$ (existential)
- $\varphi \equiv \varphi_{\mathcal{RV}} \wedge \varphi_{\mathcal{MC}}$ (universal)

Conclusions and future direction

Contribution

- A decomposition of μHML formulae into:
  - a runtime verification formula
  - a model checking formula
- Runtime verification is applicable to a larger set of formulae

Future work

- Formal correctness proof
- Empirical tests for efficiency
- Extending the approach to other formalisms
- Expressiveness study of induced μHML fragments/hierarchy

The end

Thank you