

Model Checking Almost All Paths Can Be Less Expensive Than Checking All Paths

M. Schmalz¹ H. Völzer² D. Varacca³

¹ETH Zürich, Switzerland

²IBM Zurich Research Laboratory, Switzerland

³PPS - CNRS & Univ. Paris 7, France

FSTTCS 2007

Outline

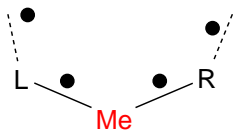
- 1 Introduction
 - Motivation
 - Fair Correctness
- 2 Comparing Complexities
 - Coincidence
 - Separation
 - Practical Advantage

Outline

- 1 Introduction
 - Motivation
 - Fair Correctness
- 2 Comparing Complexities
 - Coincidence
 - Separation
 - Practical Advantage

Motivation

Dijkstra's Dining Philosophers:



Specification:

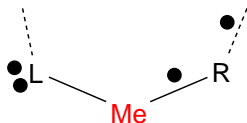
- mutual exclusion
- starvation freedom

The system is not correct!

- L and R may conspire against me.
- I have no chance to pick up the two forks at once.

Motivation

Dijkstra's Dining Philosophers:



Specification:

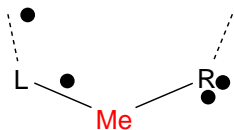
- mutual exclusion
- starvation freedom

The system is not correct!

- L and R may conspire against me.
- I have no chance to pick up the two forks at once.

Motivation

Dijkstra's Dining Philosophers:



Specification:

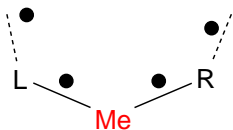
- mutual exclusion
- starvation freedom

The system is not correct!

- L and R may conspire against me.
- I have no chance to pick up the two forks at once.

Motivation

Dijkstra's Dining Philosophers:



Specification:

- mutual exclusion
- starvation freedom

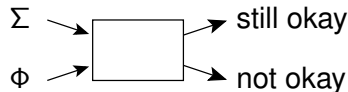
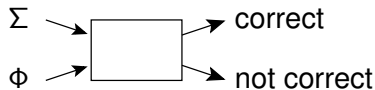
A better system may be...

- impossible (e.g. fault-tolerant consensus),
- more complicated,
- less efficient,
- unnecessary. (Bad paths do not occur in practice.)

Motivation

Live with the system at hand!

- The system is not correct w. r. t. the original specification.
- **But most** runs satisfy the specification.
- Occurs in many system models (e. g. Lamport's Paxos algorithm).
- We need a weaker notion of correctness and a related model checker \Rightarrow fair correctness, fair model checking.



Outline

- 1 Introduction
 - Motivation
 - Fair Correctness
- 2 Comparing Complexities
 - Coincidence
 - Separation
 - Practical Advantage

Fair Correctness

Definition (Varacca, Völzer, 2006)

A system Σ is **fairly correct** w. r. t. a specification Φ iff there **exists** a fairness assumption F such that each path of Σ satisfies $F \Rightarrow \Phi$.

What is **fairness**?

We use the definition of Völzer, Varacca, Kindler (2005).

Fair Model Checking

Theorem (Varacca, Völzer, 2006)

If Σ is finite and Φ is ω -regular, then Σ is fairly correct w. r. t. Φ iff $\mathbb{P}(x \models \Phi) = 1$.

\mathbb{P} is an **arbitrary** Markov measure
(compatible with the transitions of Σ).

- Checking whether Σ is fairly correct w. r. t. Φ can be done by (qualitative) probabilistic model checking!
- Checking fair correctness can be motivated even when it is not natural to apply (qual.) prob. model checking.

Overview

What we already have:

- a notion of ‘almost correct’ \Rightarrow fair correctness
- a related model checking algorithm \Rightarrow qualitative probabilistic model checking

The next questions:

- Can we use the new view on qual. prob. model checking to improve existing algorithms?
- Is fair (= qual. prob.) model checking sometimes easier than classical model checking?

Outline

- 1 Introduction
 - Motivation
 - Fair Correctness
- 2 Comparing Complexities
 - Coincidence
 - Separation
 - Practical Advantage

Notation

Specifications

$$\begin{aligned} LTL & : \phi := \psi \mid \phi \vee \phi \mid \neg \phi \mid X \phi \mid \phi U \phi \\ L(F) & : \phi := \psi \mid \phi \vee \phi \mid \neg \phi \mid F \phi \\ L(F^\infty) & : \phi := \psi \mid \phi \vee \phi \mid \neg \phi \mid F^\infty \phi \end{aligned}$$

$F^\infty := G F =$ ‘infinitely often’

Notation

Problems

L : language of specifications (e.g. LTL, $L(F)$, ...)

- UMC(L): classical (universal) model checking
Is a system correct
w. r. t. a specification drawn from L ?
- FMC(L): fair model checking
Is a system fairly correct
w. r. t. a specification drawn from L ?

Coinciding Complexities

For many interesting L
 $UMC(L)$ and $FMC(L)$ have the same complexities:

| L | $UMC(L)$ | $FMC(L)$ |
|----------------|----------|----------|
| Büchi automata | PSPACE | PSPACE |
| LTL | PSPACE | PSPACE |
| LTL+past | PSPACE | PSPACE |
| $L(F)$ | co-NP | co-NP |
| CTL | linear | linear |
| CTL* | PSPACE | PSPACE |

Coinciding Complexities

For many interesting L
 $UMC(L)$ and $FMC(L)$ have the same complexities:

| L | $UMC(L)$ | $FMC(L)$ |
|----------------|----------|----------|
| Büchi automata | PSPACE | PSPACE |
| LTL | PSPACE | PSPACE |
| LTL+past | PSPACE | PSPACE |
| $L(F)$ | co-NP | co-NP |
| CTL | linear | linear |
| CTL* | PSPACE | PSPACE |

[Vardi, 1985]

[Courcoubetis, Yannakakis, 1995]

[S. master07]

Coinciding Complexities

For many interesting L
 $UMC(L)$ and $FMC(L)$ have the same complexities:

| L | $UMC(L)$ | $FMC(L)$ |
|----------------|----------|----------|
| Büchi automata | PSPACE | PSPACE |
| LTL | PSPACE | PSPACE |
| LTL+past | PSPACE | PSPACE |
| $L(F)$ | co-NP | co-NP |
| CTL | linear | linear |
| CTL* | PSPACE | PSPACE |

[Varacca, Völzer, 2006]

Coinciding Complexities

For many interesting L
 $UMC(L)$ and $FMC(L)$ have the same complexities:

| L | $UMC(L)$ | $FMC(L)$ |
|----------------|----------|----------|
| Büchi automata | PSPACE | PSPACE |
| LTL | PSPACE | PSPACE |
| LTL+past | PSPACE | PSPACE |
| $L(F)$ | co-NP | co-NP |
| CTL | linear | linear |
| CTL* | PSPACE | PSPACE |

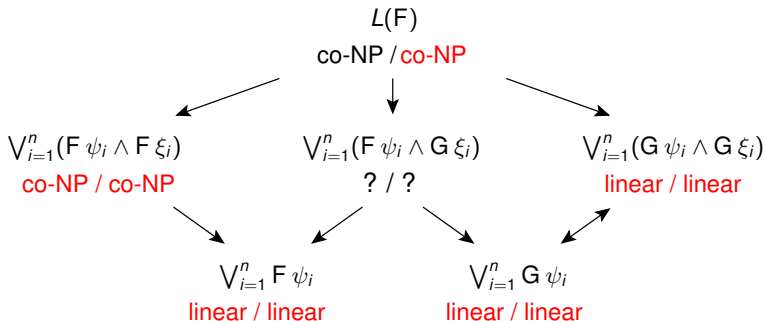
(new)

Main Idea of the Proof

- With similar arguments as in the classical case:
FMC($L(F)$) is co-NP-hard.
- With a reduction from FMC($L(F)$) to UMC($L(F)$):
FMC($L(F)$) belongs to co-NP.
- For this, we proved: $\Sigma \models_{\text{fairly}} \Phi$ iff $\Sigma \models (SF_{\Sigma} \Rightarrow \Phi)$.

More Coincidence Results

We examined the complexities of $UMC(L)$ and $FMC(L)$ for several $L \subset L(F)$:



Outline

- 1 Introduction
 - Motivation
 - Fair Correctness
- 2 Comparing Complexities
 - Coincidence
 - **Separation**
 - Practical Advantage

Differing Complexities

Theorem

- 1 $UMC(L(F^\infty))$ is co-NP complete.
- 2 $FMC(L(F^\infty))$ can be solved in linear time.

Proof Structure

Classical Model Checking is co-NP Complete

$$\bigvee_{i=1}^n (F^\infty \psi_i \wedge F^\infty \xi_i) \in L(F^\infty) \subset L(F)$$

- Sistla, Clarke, 1985: $UMC(L(F))$ belongs to co-NP.
- Emerson, Lei, 1987:
Classical model checking of a system and a specification $\bigvee_{i=1}^n (F^\infty \psi_i \wedge F^\infty \xi_i)$ is co-NP hard.
(The ψ_i, ξ_i are state formulas.)
- Together: $UMC(L(F^\infty))$ is co-NP complete.

Differing Complexities

Theorem

- 1 $UMC(L(F^\infty))$ is co-NP complete.
- 2 $FMC(L(F^\infty))$ can be solved in linear time.

Structure of Proof

Muller Formulas

Definition

A formula Φ is a Muller formula iff

- $\Phi \in L(F^\infty)$,
- each variable is 'below' a temporal operator (in the syntax tree).

Examples:

- $F^\infty \zeta \Rightarrow F^\infty \eta$ ✓
- $F^\infty (G^\infty \zeta \vee \eta)$ ✓
- $G \zeta, \zeta U \eta$ ✗
- $\zeta \wedge F^\infty \eta$ ✗

Proof Structure

Fair Model Checking of Muller Formulas Can Be Solved in Linear Time

Lemma

FMC('Muller') can be solved in linear time.

Idea of the algorithm:

- Let a system Σ and $F^\infty \zeta \vee F^\infty \eta$ be given.
- Simplification: Σ is strongly connected.
- Divide:
 - Determine states q with $\Sigma, q \models_{\text{fairly}} F^\infty \zeta$.
 - Determine states q with $\Sigma, q \models_{\text{fairly}} F^\infty \eta$.
- Conquer:
Determine states q with $\Sigma, q \models_{\text{fairly}} F^\infty \zeta \vee F^\infty \eta$.



Proof Structure

Fair Model Checking of $L(F^\infty)$ Can Be Solved in Linear Time

Lemma

$FMC(L(F^\infty))$ can be solved in linear time.

Idea of proof:

- Consider a system Σ and $\zeta \wedge F^\infty \eta$.
- Satisfaction of ζ only depends on the initial state of Σ .
- It suffices to check either $true \wedge F^\infty \eta$ or $false \wedge F^\infty \eta$.
- Both are (basically) Muller formulas.



Outline

- 1 Introduction
 - Motivation
 - Fair Correctness
- 2 Comparing Complexities
 - Coincidence
 - Separation
 - Practical Advantage

Practical Advantage

Integration of the algorithm for Muller formulas in the fair model checker of Courcoubetis and Yannakakis yields:

- elimination of Muller **sub**formulas from the input formula in linear time
- Before, this elimination required exponential time (in the worst case).
- Details can be found in [S. master07].

Summary

- Fair model checking is a **more permissive** version of **classical** model checking.
- If the specification belongs to $L(F^\infty)$, then **fair** model checking can be solved in **linear time** whereas **classical** model checking requires **exponential** time.
- Future work
 - Our results hold for **qualitative** probabilistic model checking.
Todo:
generalization to (**general**) probabilistic model checking
 - implementation and case studies