CONTROLLERS FOR VERIFICATION OF DISTRIBUTED SYSTEMS

Aiswarya Cyriac
Uppsala University, Sweden

Joint work with

Paul Gastin
LSV, ENS Cachan, France

K. Narayan Kumar
Chennai Mathematical Institute, India

Journées nationales du GDR-IM
02/02/2015 - Bordeaux
VERIFICATION

Model Checking

System $S$

Specification $\varphi$

$S \models \varphi$?

 ✓

✗

Refine $S$

(Fix bugs)
Model Checking

Undecidable in many cases

System $S$
Specification $\phi$

$S \models \phi$?

Refine $S$
(Fix bugs)

✓
✗
UNDER-APPROXIMATE VERIFICATION

Parametrised
UNDER-APPROXIMATE VERIFICATION

Parametrised
UNDER-APPROXIMATE VERIFICATION

> Parametrised
UNDER-APPROXIMATE VERIFICATION

Parametrised
UNDER-APPROXIMATE VERIFICATION

Parametrised
UNDER-APPROXIMATE VERIFICATION

- Parametrised
- Exhaustive
UNDER-APPROXIMATE VERIFICATION

- Parametrised
- Exhaustive
UNDER-APPROXIMATE VERIFICATION

Model Checking

\( S \models \varphi ? \)

System \( S \)
Specification \( \varphi \)

Refine \( S \) (Fix bugs)

\( \checkmark \)

Decidable

\( \gg \)
UNDER-APPROXIMATE VERIFICATION

Model Checking

\[ S \models^k \varphi \]

Decidable

System \( S \)

Specification \( \varphi \)

Refine \( S \) (Fix bugs)
UNDER-APPROXIMATE VERIFICATION

Model Checking

\[ S \models^k \phi \]

System \( S \)

Specification \( \phi \)

Refine \( S \) (Fix bugs)

Decidable

\[ \checkmark \]

\[ \times \]
UNDER-APPROXIMATE VERIFICATION

Model Checking

\[ S \models^k \varphi? \]

System \( S \)

Specification \( \varphi \)

Decidable

Refine \( S \) (Fix bugs)
UNDER-APPROXIMATE VERIFICATION

System $S$  
Specification $\varphi$

Model Checking $\models \varphi$?

Refine $S$ (Fix bugs)

$\succ$ Decidable

✓ $\times$
UNDER-APPROXIMATE VERIFICATION

Model Checking

\[ S \models^k \varphi? \]

System \( S \)
Specification \( \varphi \)

Refine \( S \)
(Fix bugs)

Decidable

\[ \checkmark \]

\[ \times \]
UNDER-APPROXIMATE VERIFICATION

Model Checking

$S \models^k \varphi$?

System $S$

Specification $\varphi$

Decidable

Refine $S$

(Fix bugs)
UNDER-APPROXIMATE VERIFICATION

Model Checking

$\models \phi$?

System $S$

Specification $\phi$

Refine $S$

(Fix bugs)

$\triangleright$ Decidable

✓

✗
CONTROLLERS FOR VERIFICATION OF DISTRIBUTED SYSTEMS
COMMUNICATING DISTRIBUTED SYSTEMS

Process 1

Network

Process 2

Process 4

Process 3
CONTROLLERS FOR VERIFICATION OF DISTRIBUTED SYSTEMS
COMMUNICATING DISTRIBUTED SYSTEMS

Process 1

Process 2

Process 3

Network
CONTROLLERS FOR DISTRIBUTED SYSTEMS

From

Process 1
Controller 1

To

Process 2
Controller 2

Process 3
Controller 3

Network
CONTROLLERS FOR DISTRIBUTED SYSTEMS

Network

Process 1 → Controller 1

Process 2 → Controller 2

Process 3 → Controller 3

Controller 1 → Controller 2

Controller 2 → Controller 3

From → To

Heavy

Fragile
CONTROLLERS FOR DISTRIBUTED SYSTEMS

- Collection of local controllers
- Communication via piggy-backing
- Privacy: Do NOT read states/messages
LET’S DESIGN A CONTROLLER

UNDER-APPROXIMATION: BOUNDED (K) PHASE

- Bounded number (k) of phases
- Phase: Receive from one process, send to all processes
- No cycles
- Bounded number \((k)\) of phases
- Phase: Receive from one proc, send to all procs
- No cycles
Bounded number (k) of phases
Phase: Receive from one proc, send to all procs
No cycles
Bounded number (k) of phases
Phase: Receive from one proc, send to all procs
No cycles
- Bounded number (k) of phases
- Phase: Receive from one proc, send to all procs
- No cycles
> Bounded number (k) of phases
> Phase: Receive from one proc, send to all procs
> No cycles
- Bounded number \((k)\) of phases
- Phase: Receive from one proc, send to all procs
- No cycles
- Bounded number ($k$) of phases
- Phase: Receive from one proc, send to all procs
- No cycles
Bounded number (k) of phases

Phase: Receive from one proc, send to all procs

No cycles
Bounded number (k) of phases
Phase: Receive from one proc, send to all procs
No cycles
- Bounded number \( (k) \) of phases
- Phase: Receive from one proc, send to all procs
- No cycles
DISTRIBUTED CONTROLLER FOR K-BOUNDED PHASE U-A

- Bounded number (k) of phases
- Phase: Receive from one proc, send to all processes
- No cycles
DISTRIBUTED CONTROLLER FOR K-BOUNDED PHASE U-A

- Bounded number (k) of phases
- Phase: Receive from one proc, send to all processes
- No cycles

A local controller for each process
DISTRIBUTED CONTROLLER FOR K-BOUNDED PHASE U-A

A local controller for each process

- Bounded number \((k)\) of phases
- Phase: Receive from one proc, send to all processes
- No cycles

- Has a Phase Counter
- Remembers current sender
DISTRIBUTED CONTROLLER FOR K-BOUNDED PHASE U-A

- A local controller for each process
  - Has a Phase Counter
  - Remembers current sender
  - Different sender?
  - Detect Cycle?

- Bounded number \((k)\) of phases
- Phase: Receive from one proc, send to all processes
- No cycles
DISTRIBUTED CONTROLLER FOR K-BOUNDED PHASE U-A

- A local controller for each process
- Has a Phase Counter
- Remembers current sender
- Different sender?
- Detect Cycle?
- Increment counter; Update channel
- Bounded number (k) of phases
- Phase: Receive from one proc, send to all processes
- No cycles
DISTRIBUTED CONTROLLER FOR K-BOUNDED PHASE U-A

A local controller for each process

- Bounded number (k) of phases
- Phase: Receive from one proc, send to all processes
- No cycles

Transitions

Different sender?

Detect Cycle?

State

Has a Phase Counter

Remembers current sender

Increment counter, Update channel
DISTRIBUTED CONTROLLER FOR K-BOUNDED PHASE U-A

Detect Cycle?

Phase Vectors
DISTRIBUTED CONTROLLER FOR K-BOUNDED PHASE U-A

Detect Cycle?

Phase Vectors

best info about phase number of other processes
DISTRIBUTED CONTROLLER FOR K-BOUNDED PHASE U-A

Detect Cycle?

Phase Vectors

Sends: tag with phase vector

best info about phase number of other processes
DISTRIBUTED CONTROLLER FOR K-BOUNDED PHASE U-A

Detect Cycle?

Phase Vectors

best info about phase number of other processes

Sends: tag with phase vector

Receives: update phase vector by taking MAX
CONTROLLERS FOR BOUNDED PHASE DISTRIBUTED SYSTEMS

- Collection of local controllers
- Communication via piggy-backing
- Privacy: Do NOT read states/messages
CONTROLLERS FOR
BOUND PHASE
DISTRIBUTED SYSTEMS

- Collection of local controllers
- Communication via piggy-backing
- Privacy: Do NOT read states/messages

- System independent
- Generic
- Deterministic
- Finite state
CONTROLLERS FOR VERIFICATION OF DISTRIBUTED SYSTEMS


Merci !