Safety Analysis of Parameterised Networks with Non-Blocking Rendez-Vous

Lucie Guillou, Arnaud Sangnier, Nathalie Sznajder
IRIF, Université Paris Cité  IRIF, Université Paris Cité  LIP6, Sorbonne Université
16 oct. 2023, Verification Seminar
Parameterised Distributed Networks

- Unknown number of agents
- Each agent follows a protocol given as a finite-state machine
- Synchronous Communication
- Interleaving Semantics

... a protocol
Is there a number of agents such that there exists a run leading to a bad configuration?
The Model

- All agents execute the same finite-state machine called a Protocol.

### Protocol Diagram

- **Initial State**: 0
- **Internal Transition**: 1 to 2
- **Reception Transition**: 3 to 0
- **Sending Transition**: 4 to 5
- **Finite Set of Messages**: 1, 2, 3, 4, 5

**Diagram Notes**
- Transition labels: ![a], ![b], ![c], ?a, ?b, ?c
Communication by Rendez-Vous

Initial configuration with 4 processes

\[ \text{RDV}(a) \rightarrow \tau \rightarrow \text{RDV}(b) \rightarrow \ldots \]

\( \rightarrow \text{IMPOSSIBLE TO REACH} \) STATES 4 AND 5.
Communication by Non-Blocking Rendez-Vous

- Ex: Java Parallel Multithreads Programming
  - Wait / Notify
- Rendez-Vous is no longer symmetric
- More behaviors than in the rendez-vous semantics.
Communication by Non-Blocking Rendez-Vous
Verification Problems.

**Conf-Cover:** Given a protocol and a configuration, is there such that

**SynchrO:** Given a protocol and a state, is there such that
Results

* Rendez-Vous:
  * CONE-COVER: $\mathcal{E}$ Ptime [GS 92]
  * SYNCHRO: $\mathcal{E}$ Ptime [HS 2020] [BER 2021]

* Non-Blocking Rendez-Vous:
  * CONE-COVER: EXPSPACE - complete [CONCUR' 23]
  * SYNCHRO: Undecidable [CONCUR' 23]
Results

Non-Blocking Rendez-Vous:
- CONE-COVER: \text{ExpSPACE - complete}
- SYNCHRO: Undecidable

\text{ExpSPACE - membership}:
Rackoff, \text{ExpSPACE - membership of Coverability for Vector Addition Systems with States (VASS)}.

\text{ExpSPACE - hardness}:
Lipton, \text{ExpSPACE - hardness of Coverability for VASS}.

Undecidability:
Simulation of a 2-counters machine with tests to 0.
Why such a complexity gap?

In Rendez-vous semantics, we have a nice property:

Copycat Lemma:
If a state is coverable, then any configuration is coverable

$\Rightarrow$ Conf-COVER and SYNCHRO in Ptime
Why such a complexity gap?

Main ingredient: with non-blocking rendez-vous, we can isolate some processes
Why such a complexity gap?

Main ingredient: with non-blocking rendez-vous, we can isolate some processes.

at most one process on \{1, 2, 3, 5\}.

\[ \begin{array}{c}
?b, !c \\
\rightarrow \\
!a \\
\rightarrow \\
1 \\
\downarrow ?a \\
\rightarrow \\
2 \\
\downarrow ?a \\
\rightarrow \\
3 \\
\downarrow ?a \\
\end{array} \]

\[ \begin{array}{c}
\text{NB}(a) \\
\rightarrow \\
\text{RDV}(a) \\
\end{array} \]

\[ \begin{array}{c}
\Rightarrow \text{Conf. COVER} \\
\text{EXPSpace - hard} \\
\text{SYNCHRO Undecidable} \\
\end{array} \]
Results

Non-Blocking Rendez-Vous:

- Conf-Cover: EXPSPACE-complete
- SYNCHRO: Undecidable

EXPSPACE-membership:

Rackoff, EXPSPACE-membership of Coverability for Vector Addition Systems with States (VASS).

EXPSPACE-hardness:

Lipton, EXPSPACE-hardness of Coverability for VASS.

4 State-Cover (covering one state)

Undecidability:

Simulation of a 2-counters machine with tests to 0.
A \textbf{VASS} of dimension $d \in \mathbb{N}$ is a tuple $(S, \Delta)$, $\Delta \subseteq S \times \mathbb{Z}^d \times S$, with $S_0$ an initial state.

Configuration: $(s, \sigma) \in S$, $\sigma \in \mathbb{N}^d$.

Semantics: $(s, \sigma) \xrightarrow{\Delta} (s', \sigma')$ if $\delta = (s, t, s') \in \Delta$.

$\sigma' = \sigma + t$.
Preliminaries

A VASS of dimension $d \in \mathbb{N}$ is a tuple $(S, \Delta) \Delta \subseteq S \times \mathbb{Z}^d \times S$, with $S_0$ an initial state.

Configuration: $(s, \sigma) \ s \in S, \ \sigma \in \mathbb{N}^d$

Semantics: $(s, \sigma) \xrightarrow{\delta} (s', \sigma') \ \delta = (s, t, s') \in \Delta$

$\sigma' = \sigma + t$

$V = (S, \Delta)$ is **k-bounded** $(k \in \mathbb{N})$ if $\forall (s, \sigma)$ s.t $\ (S_0, 0_d) \xrightarrow{*} (s, \sigma)$, $\forall i \in \{1, d\}, \ \sigma(i) \leq k.$
Preliminaries

\[ 2\text{Exp}-\text{bounded VASS} = 2^{2^n} - \text{bounded VASS} \]

\( n \) = size of the VASS
Preliminaries

2Exp-bounded VASS = $2^{2^n}$-bounded VASS
($n =$ size of the VASS)

COVER: \( V = (S, A) \) and \( s_0 \in S, \exists v \text{ s.t. } (s_0, O_0) \rightarrow^k (s_f, v) \)
Preliminaries

2\text{Exp}-\text{bounded VASS} = 2^{2^n} - \text{bounded VASS} \\
(\text{n = size of the VASS})

\text{COVER: } V = (S, \Delta) \text{ and } s \in S, \exists u \text{ s.t. } (s_0, 0_d) \xrightarrow{n} (s_f, u) ?

\text{Thm: } [1, 2] \text{ COVER for } 2\text{Exp}-\text{bounded VASS is ExpSpace-hard.}

Expspace-hardness of COVER

Steps of the proof:

① How to simulate a k-bounded VASS with our model

② How to adapt the construction for 2Exp-bounded VASS.
Regaining synchronization between two processes...

1. ! hds(m)
2. ? ack(m)
3. 

4. ? hds(m)
5. ! ack(m)
6. 

\[ \text{HDS}(m) \]

ADV(hds(m))

ADV(ack(m))
Simulate a K-bounded VASS.

Final state

Controller

Processes simulating vectors' values.

The protocol.
Simulating vectors' values.

Value of $v(i)$: number of processes on $1_i$

Processes:
- Simulating
- Vectors' value:

Controller's part:

$\delta = (s, t, \delta')$  
$t(1) = +2$  
$t(2) = -1$

Controller:

$\xrightarrow{\text{HDS(inc}_1\text{)}} \xrightarrow{\text{HDS(inc}_1\text{)}} \xrightarrow{\text{HDS(dec}_2\text{)}} \delta'$
Let's be careful...

\[ \mathcal{O} \xrightarrow{!c} \text{Controller} \xrightarrow{?c} \mathcal{O} \]

- It starts with some non-empty components' place
- **NB**: at most \( k \) processes on each component's place.
- Non-blocking part
- We need to "reset" vectors' values

\[ \text{NBC}(c) \xrightarrow{} \mathcal{O} \ldots \xrightarrow{} \mathcal{O} \xrightarrow{} \text{RDV}(c) \]

\[ \ldots \]
Reset vectors' values.

Before reaching \( \Theta \), the controller resets components:

\[
0 \xrightarrow{!\text{AST 1}} 0 \xrightarrow{!\text{AST 1}} 0 \cdots \xrightarrow{!\text{AST 1}} 0 \xrightarrow{!\text{RST 2}} 0 \xrightarrow{!\text{RST 2}} 0 \cdots 0 \xrightarrow{!\text{RST d}} \Theta
\]

\( k \) times

\[
!\text{AST i} \xrightarrow{?\text{RST i}} 0
\]
Simulate a $2^{\text{Exp}}$-bounded VASS

Inspired by Lipton's proof for Expspace-hardness for VASS.

Find a way to reset places of $2^{2n}$. 
Results

Non-Blocking Rendez-Vous:

- Conf-Cover: EXPSPACE-complete
- SYNCHRO: Undecidable

EXPSPACE-membership:

Rackoff, EXPSPACE-membership of Coverability for Vector Addition Systems with States (VASS).

EXPSPACE-hardness:

Lipton, EXPSPACE-hardness of Coverability for VASS.

- 4 State-Cover (covering one state)

Undecidability:

Simulation of a 2-counters machine with tests to 0.
A Restriction: Wait-Only Protocol

Protocols where each state is either:

- an action state
- a waiting state

\[
\begin{align*}
0 & \xrightarrow{! b} 2 \\
2 & \xrightarrow{? a} 3 \\
3 & \xrightarrow{? b} 4 \\
4 & \xrightarrow{? b} 0
\end{align*}
\]
Wait - Only Protocols

- Non-Blocking Rendez-Vous: with Wait-Only Protocols
  - Conf-Cover: ExpSPACE complete in Ptime
  - Synchro: Undecidable

- Why? Isolation mechanism is less powerful

- How? Abstraction on Configurations. Inductive computation until saturation
Abstract Configurations.

Coverable states with any number of agents.

Reachable all together.

Gives us all the coverable configurations.
Computation

![Diagram of a computation process with nodes labeled 0, 1, 2, 3, and 4, connected by edges labeled with symbols !a, !b, ?a, and ?b.]

<table>
<thead>
<tr>
<th>$S$</th>
<th>Tokas</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Computation

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>Toks</td>
</tr>
<tr>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>
At most one agent on state 2, reachable through a non-blocking sending of $b$. 
At most one agent on state 2, reachable through a non-blocking sending of b.
Computation

Table:

<table>
<thead>
<tr>
<th>$S$</th>
<th>Toks</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0, 1, 3, 4</td>
</tr>
<tr>
<td>2, b</td>
<td></td>
</tr>
</tbody>
</table>
Computation

\[ \text{NB}(a)^K \land \text{NDV}(b)^K \]
Computation

<table>
<thead>
<tr>
<th>S</th>
<th>Tokas</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>
At most one process on states \{1, 2\}
Computation

At most one process on states \{1, 2\}
Computation

\[(NB(b) \ ADV(a))^k\]
Computation

\[(NB(b) \text{ RDV}(a))^k\]
Conclusion and Future Work

- New semantics leading to an important complexity gap compared to the rendez-vous semantics.

- Restriction allowing to regain a Ptime algorithm for the conf-COVER problem.

- New restriction allowing to regain decidability for the SYNCHRO problem?
Thank you everyone for your attention.