On Parameterized Verification of Asynchronous Shared-Memory Pushdown Systems

Marie Fortin\textsuperscript{1} \hspace{1cm} Anca Muscholl\textsuperscript{2} \hspace{1cm} Igor Walukiewicz\textsuperscript{2}

\textsuperscript{1}LSV, ENS Cachan
\textsuperscript{2}LaBRI, University of Bordeaux

Highlights 2016, Brussels
(C, D)-systems
(\mathcal{C}, \mathcal{D})\text{-systems}

One leader process \(\mathcal{D}\)
(pushdown system)
The model

Repeated reachability

Universal reachability

General result

\[(C, D)\text{-systems}\]

One leader process \(D\)

(pushdown system)

Arbitrarily many identical and anonymous contributors \(C\)

(pushdown systems)
(C, D)-systems

One leader process D
(pushdown system)

Bounded shared register
read/write
No lock

Arbitrarily many identical and
anonymous contributors C
(pushdown systems)
### Previous Work

<table>
<thead>
<tr>
<th>The model</th>
<th>Repeated reachability</th>
<th>Universal reachability</th>
<th>General result</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Is there a run of the $(C, D)$-system where the leader performs $\top$, for some number of contributors?</td>
<td>$\text{Expspace}$</td>
<td>$\text{Pspace}$-complete</td>
</tr>
<tr>
<td></td>
<td>$[\text{Hague, 2011}]$</td>
<td>$[\text{Esparza, Ganty, Majumdar, 2013}]$</td>
<td>$\text{Pspace}$-hard and in $\text{Nexptime}$</td>
</tr>
<tr>
<td></td>
<td>Is there a run of the $(C, D)$-system where the leader performs $\top$ infinitely often, for some number of contributors?</td>
<td>$[\text{Durand-Gasselin, Esparza, Ganty, Majumdar, 2015}]$</td>
<td>$[\text{La Torre, Muscholl, Walukiewicz, 2015}]$</td>
</tr>
</tbody>
</table>
Previous Work

**Reachability**

Is there a run of the \((C, D)\)-system where the leader performs a special action \(\top\), for some number of contributors?

[Hague, 2011] \textbf{EXPSPACE}

[Esparza, Ganty, Majumdar, 2013] \textbf{PSPACE-complete}

[La Torre, Muscholl, Walukiewicz, 2015] Generalization
Previous Work

### Reachability
Is there a run of the \((C, D)\)-system where the leader performs a special action \(\top\), for some number of contributors?

[Hague, 2011] \textbf{Expspace}

[Esparza, Ganty, Majumdar, 2013] \textbf{PSPACE-complete}

[La Torre, Muscholl, Walukiewicz, 2015] Generalization

### Repeated reachability
Is there a run of the \((C, D)\)-system where the leader performs \(\top\) infinitely often, for some number of contributors?

[Durand-Gasselin, Esparza, Ganty, Majumdar, 2015] \textbf{PSPACE-hard and in \textsc{Nexptime}}
<table>
<thead>
<tr>
<th>The model</th>
<th>Universal reachability</th>
<th>General result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Repeated reachability</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The repeated reachability problem is $\text{Pspace}$-complete.

\[ \text{Durand-Gasselin, Esparza, Ganty, Majumdar, 2015} \]:

- reduction to the case of finite-state contributors, by bounding the stacks of the contributors
- $\text{NP}$ in the case of finite-state contributors $\rightarrow \text{Nexptime}$ for pushdown contributors

We re-use the reduction to finite-state contributors, but change the decision procedure
The repeated reachability problem is $\mathsf{PSPACE}$-complete.

Durand-Gasselin, Esparza, Ganty, Majumdar, 2015: Reduction to the case of finite-state contributors, by bounding the stacks of the contributors $\mathsf{NP}$ in the case of finite-state contributors $\rightarrow \mathsf{Nexptime}$ for pushdown contributors.

We re-use the reduction to finite-state contributors, but change the decision procedure.
Repeated reachability

Theorem

The repeated reachability problem is $P_{SPACE}$-complete.

[Durand-Gasselin, Esparza, Ganty, Majumdar, 2015]:

- Reduction to the case of finite-state contributors, by bounding the stacks of the contributors
- NP in the case of finite-state contributors $\rightarrow$ $NEXPTIME$ for pushdown contributors
Theorem
The repeated reachability problem is \( \text{PSPACE}\)-complete.

[Durand-Gasselin, Esparza, Ganty, Majumdar, 2015]:
- Reduction to the case of finite-state contributors, by bounding the stacks of the contributors
- \( \text{NP} \) in the case of finite-state contributors \( \rightarrow \text{NEXPTIME} \) for pushdown contributors

We re-use the reduction to finite-state contributors, but change the decision procedure
Key steps for the $\text{PSPACE}$ upper bound

- Look for an ultimately periodic run.
Key steps for the \textbf{PSPACE} upper bound

- Look for an ultimately periodic run.
- Adapt from finite runs to infinite periodic runs the techniques of [La Torre, Muscholl, Walukiewicz, 2015]:

\[
\text{Intersection emptiness of } \ell + 1 \text{ finite automata computable in } \text{PSPACE}.
\]
Key steps for the $\text{PSPACE}$ upper bound

- Look for an ultimately periodic run.
- Adapt from finite runs to infinite periodic runs the techniques of [La Torre, Muscholl, Walukiewicz, 2015]:
  - Define a transition system $D^\kappa = D + \text{capacity}$: set of values written by the contributors in the register.

**Idea:** if a contributor can produce a value $g$ once, by adding copies of this contributor we can produce as many $g$’s as needed.
Key steps for the \textsc{Pspace} upper bound

- Look for an ultimately periodic run.
- Adapt from finite runs to infinite periodic runs the techniques of [La Torre, Muscholl, Walukiewicz, 2015]:
  - Define a transition system $D^\kappa = D + \text{capacity}$: set of values written by the contributors in the register.

  \textbf{Idea:} if a contributor can produce a value $g$ once, by adding copies of this contributor we can produce as many $g$’s as needed.

  - Define similarly $C^\kappa$. A loop in $D^\kappa$ corresponds to a loop in the $(C, D)$-system if each addition to the capacity is \textbf{supported} by a loop in $C^\kappa$ producing the necessary write.
Key steps for the **PSPACE** upper bound

- Look for an ultimately periodic run.
- Adapt from finite runs to infinite periodic runs the techniques of [La Torre, Muscholl, Walukiewicz, 2015]:
  - Define a transition system $\mathcal{D}^\kappa = \mathcal{D} + \text{capacity}$: set of values written by the contributors in the register.

**Idea:** if a contributor can produce a value $g$ once, by adding copies of this contributor we can produce as many $g$’s as needed.

- Define similarly $\mathcal{C}^\kappa$. A loop in $\mathcal{D}^\kappa$ corresponds to a loop in the $(\mathcal{C}, \mathcal{D})$-system if each addition to the capacity is supported by a loop in $\mathcal{C}^\kappa$ producing the necessary write.
- Replace $\mathcal{D}^\kappa$ by its downward closure, and look for a supported loop in $\mathcal{D}^\kappa \downarrow$: one run of $\mathcal{D}^\kappa \downarrow + \ell$ runs of $\mathcal{C}^\kappa$. 

Intersection emptiness of $\ell + 1$ finite automata computable in **Pspace**.
Key steps for the \textsf{PSPACE} upper bound

- Look for an ultimately periodic run.
- Adapt from finite runs to infinite periodic runs the techniques of [La Torre, Muscholl, Walukiewicz, 2015]:
  - Define a transition system $\mathcal{D}^\kappa = \mathcal{D} + \text{capacity}$: set of values written by the contributors in the register.

\textbf{Idea:} if a contributor can produce a value $g$ once, by adding copies of this contributor we can produce as many $g$'s as needed.

- Define similarly $\mathcal{C}^\kappa$. A loop in $\mathcal{D}^\kappa$ corresponds to a loop in the $(\mathcal{C}, \mathcal{D})$-system if each addition to the capacity is \textit{supported} by a loop in $\mathcal{C}^\kappa$ producing the necessary write.
- Replace $\mathcal{D}^\kappa$ by its downward closure, and look for a supported loop in $\mathcal{D}^\kappa \downarrow$: one run of $\mathcal{D}^\kappa \downarrow + \ell$ runs of $\mathcal{C}^\kappa$.
  $\rightarrow$ Intersection emptiness of $\ell + 1$ finite automata computable in \textsf{PSPACE}. 
### Universal reachability

<table>
<thead>
<tr>
<th>Reachability</th>
<th>Universal reachability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Is there a run of the ((C, D))-system where the leader performs a special action (⊤), for some number of contributors?</td>
<td>Does the leader perform (⊤) in all ((\text{finite or infinite})) maximal runs of the ((C, D))-system, for any number of contributors?</td>
</tr>
<tr>
<td>Repeated reachability</td>
<td></td>
</tr>
</tbody>
</table>
Universal reachability

Motivation
Correctness problems for distributed algorithms: is the outcome correct (does the leader execute $\top$) for an arbitrary number of participants and for every run of the algorithm?
Universal reachability

Motivation
Correctness problems for distributed algorithms: is the outcome correct (does the leader execute $\top$) for an arbitrary number of participants and for every run of the algorithm?

Specificity
We consider finite maximal runs as well as infinite ones
$\rightarrow$ deadlock detection
Universal reachability

Theorem

The universal reachability problem is co\text{NP}-complete.
Universal reachability

Theorem
The universal reachability problem is co\textsc{Nexptime}-complete.

Is there a maximal run without any occurrence of $\top$?
Universal reachability

**Theorem**

The universal reachability problem is co\textup{NEXPTIME}-complete.

Is there a maximal run without any occurrence of \( \top \)?

- For infinite runs: reduction to repeated reachability
The universal reachability problem is co\text{NEXPTIME}-complete.

Is there a maximal run without any occurrence of $\top$?

- For infinite runs: reduction to repeated reachability
- For finite runs: use the reduction to the case of finite-state contributors, show that it is \text{NP}-complete
The universal reachability problem is \text{coNExptime}-complete.

Is there a maximal run without any occurrence of \top?

- For infinite runs: reduction to repeated reachability
- For finite runs: use the reduction to the case of finite-state contributors, show that it is \text{NP}-complete
- \text{NExptime}-hardness: tiling of the $2^n \times 2^n$ square.
Until now: verification of properties on leader actions only.

We consider the verification of regular properties \( P \subseteq (\Sigma_C \cup \Sigma_D)^* \cup (\Sigma_C \cup \Sigma_D)^\omega \) on both

- finite and infinite traces
- leader and contributor actions
Generalization

Until now: verification of properties on leader actions only.

We consider the verification of regular properties
\[ \mathcal{P} \subseteq (\Sigma_C \cup \Sigma_D)^* \cup (\Sigma_C \cup \Sigma_D)^\omega \]
on both
- finite and infinite traces
- leader and contributor actions

**Restriction**

*\(C\)-expanding properties:* if \(u \in \mathcal{P}\) and \(u'\) is obtained by repeating some contributor actions in \(u\), then \(u' \in \mathcal{P}\).
Main result

Theorem

The following problem is \textsc{Nexptime}-complete:

\textbf{Input:} a $(C, D)$-system, and a regular $C$-expanding property $P \subseteq (\Sigma_C \cup \Sigma_D)^* \cup (\Sigma_C \cup \Sigma_D)\omega$.

\textbf{Question:} Is there a maximal trace of the $(C, D)$-system that belongs to $P$?
Main result

Theorem

The following problem is \textsc{NEXPTIME}-complete:

\textbf{Input:} a \( (C, D) \)-system, and a regular \( C \)-expanding property \( \mathcal{P} \subseteq (\Sigma_C \cup \Sigma_D)^* \cup (\Sigma_C \cup \Sigma_D)^\omega \).

\textbf{Question:} Is there a maximal trace of the \( (C, D) \)-system that belongs to \( \mathcal{P} \)?

Steps of the proof:

- Reduction to a property on leader actions, by transforming the \( (C, D) \)-system into one where all contributor actions are reflected in leader writes.
Main result

Theorem

The following problem is \( \text{NEXPTIME} \)-complete:

**Input:** a \(( \mathcal{C}, \mathcal{D} )\)-system, and a regular \( \mathcal{C} \)-expanding property \( \mathcal{P} \subseteq ( \Sigma_{\mathcal{C}} \cup \Sigma_{\mathcal{D}} )^* \cup ( \Sigma_{\mathcal{C}} \cup \Sigma_{\mathcal{D}} )^\omega \).

**Question:** Is there a maximal trace of the \(( \mathcal{C}, \mathcal{D} )\)-system that belongs to \( \mathcal{P} \) ?

Steps of the proof:

- Reduction to a property on leader actions, by transforming the \(( \mathcal{C}, \mathcal{D} )\)-system into one where all contributor actions are reflected in leader writes
- Reduction to our previous results
  - Infinite traces: reduction to repeated reachability
  - Finite traces: results about universal reachability
Summary

We improved the complexity upper bound for repeated reachability from \textit{Nexptime} to \textit{Pspace}. We introduced universal reachability, and showed that it is $\text{co-Nexptime}$-complete. Verification of regular $C$-expanding properties is also $\text{Nexptime}$-complete.

Thank you!
Summary

- We improved the complexity upper bound for repeated reachability from \( \text{Nexptime} \) to \( \text{Pspace} \)
Summary

- We improved the complexity upper bound for repeated reachability from $\text{NEXPTIME}$ to $\text{PSPACE}$
- We introduced universal reachability, and showed that it is $\text{coNEXPTIME}$-complete
We improved the complexity upper bound for repeated reachability from $\text{NEXPTIME}$ to $\text{PSPACE}$.

We introduced universal reachability, and showed that it is $\text{coNEXPTIME}$-complete.

Verification of regular $C$-expanding properties is also $\text{NEXPTIME}$-complete.
Summary

- We improved the complexity upper bound for repeated reachability from \textbf{Nexptime} to \textbf{Pspace}
- We introduced universal reachability, and showed that it is \textbf{coNexptime}-complete
- Verification of regular $C$-expanding properties is also \textbf{Nexptime}-complete

Thank you!