The Complexity of Bounded Context Switching with Dynamic Thread Creation

Pascal Baumann, Rupak Majumdar, Ramanathan S. Thinniyam, Georg Zetzsche

Max Planck Institute for Software Systems (MPI-SWS)

HIGHLIGHTS 2020 (originally ICALP 2020)

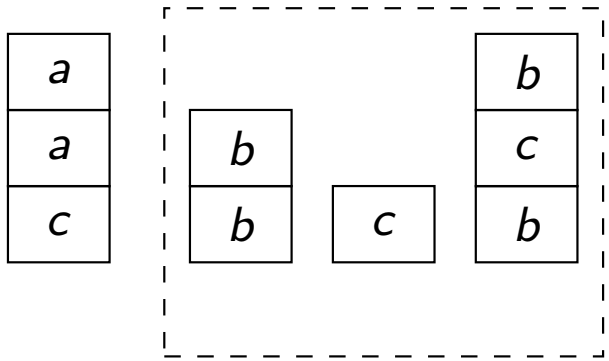
Dynamic Networks of Concurrent Pushdown Systems (DCPS)

Model Features:

Dynamic Networks of Concurrent Pushdown Systems (DCPS)

Model Features:

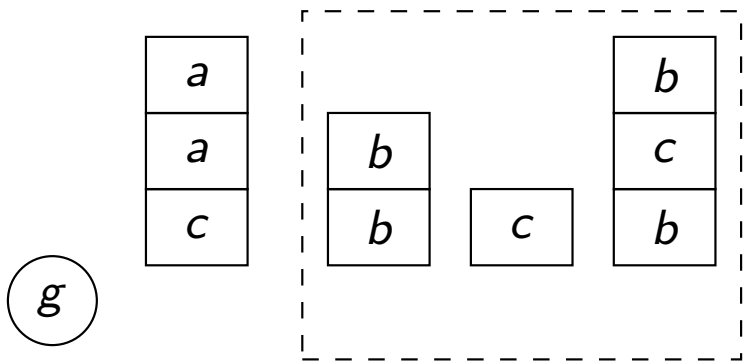
- Concurrent threads with local recursion.



Dynamic Networks of Concurrent Pushdown Systems (DCPS)

Model Features:

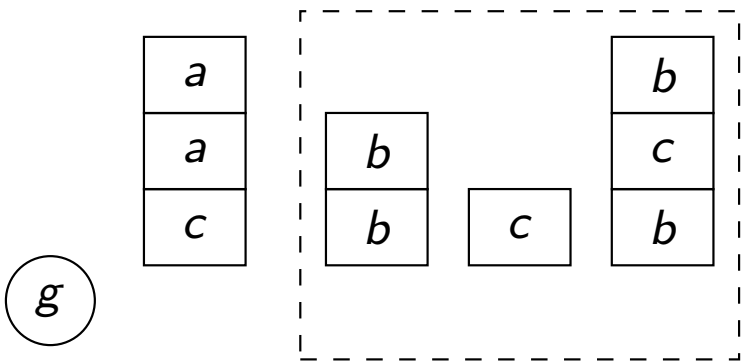
- Concurrent threads with local recursion.
- A finite global memory, accessible by all threads.



Dynamic Networks of Concurrent Pushdown Systems (DCPS)

Model Features:

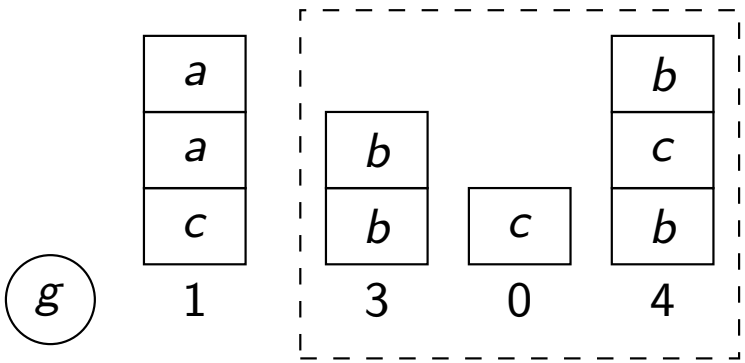
- Concurrent threads with local recursion.
- A finite global memory, accessible by all threads.
- New threads being spawned dynamically during execution.



Dynamic Networks of Concurrent Pushdown Systems (DCPS)

Model Features:

- Concurrent threads with local recursion.
- A finite global memory, accessible by all threads.
- New threads being spawned dynamically during execution.
- Bound K on context switches per thread (avoids undecidability).



Safety Verification

K-bounded state reachability problem for DCPS (SRP[K])

Input A DCPS $\mathcal{A}$ and a global state g

Question Is g K-bounded reachable in $\mathcal{A}$?

Safety Verification

K-bounded state reachability problem for DCPS (SRP[K])

Input A DCPS $\mathcal{A}$ and a global state g

Question Is g K-bounded reachable in $\mathcal{A}$?

SRP[0] is EXPSPACE-complete.

- Shown by Ganty and Majumdar (2012).

SRP[K] is EXPSPACE-hard and in 2EXPSPACE for every $K \geq 1$.

- Shown by Atig, Bouajjani, and Qadeer (2009).

Safety Verification

K-bounded state reachability problem for DCPS (SRP[K])

Input: A DCPS $\mathcal{A}$ and a global state g

Question: Is g K-bounded reachable in $\mathcal{A}$?

SRP[0] is EXPSPACE-complete.

- Shown by Ganty and Majumdar (2012).

SRP[K] is EXPSPACE-hard and in 2EXPSPACE for every $K \geq 1$.

- Shown by Atig, Bouajjani, and Qadeer (2009).

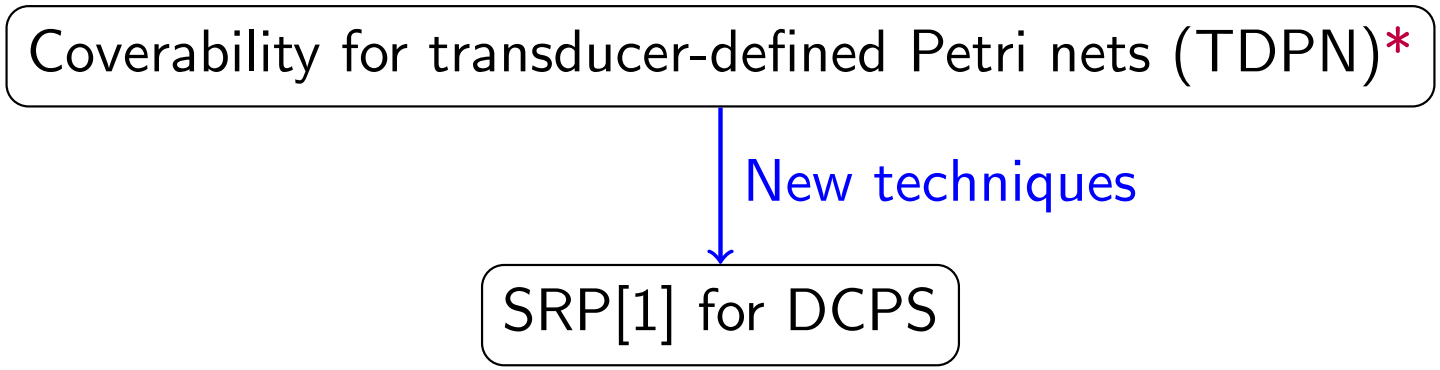
Our main result

SRP[K] is 2EXPSPACE-hard for every $K \geq 1$.

Proof Outline

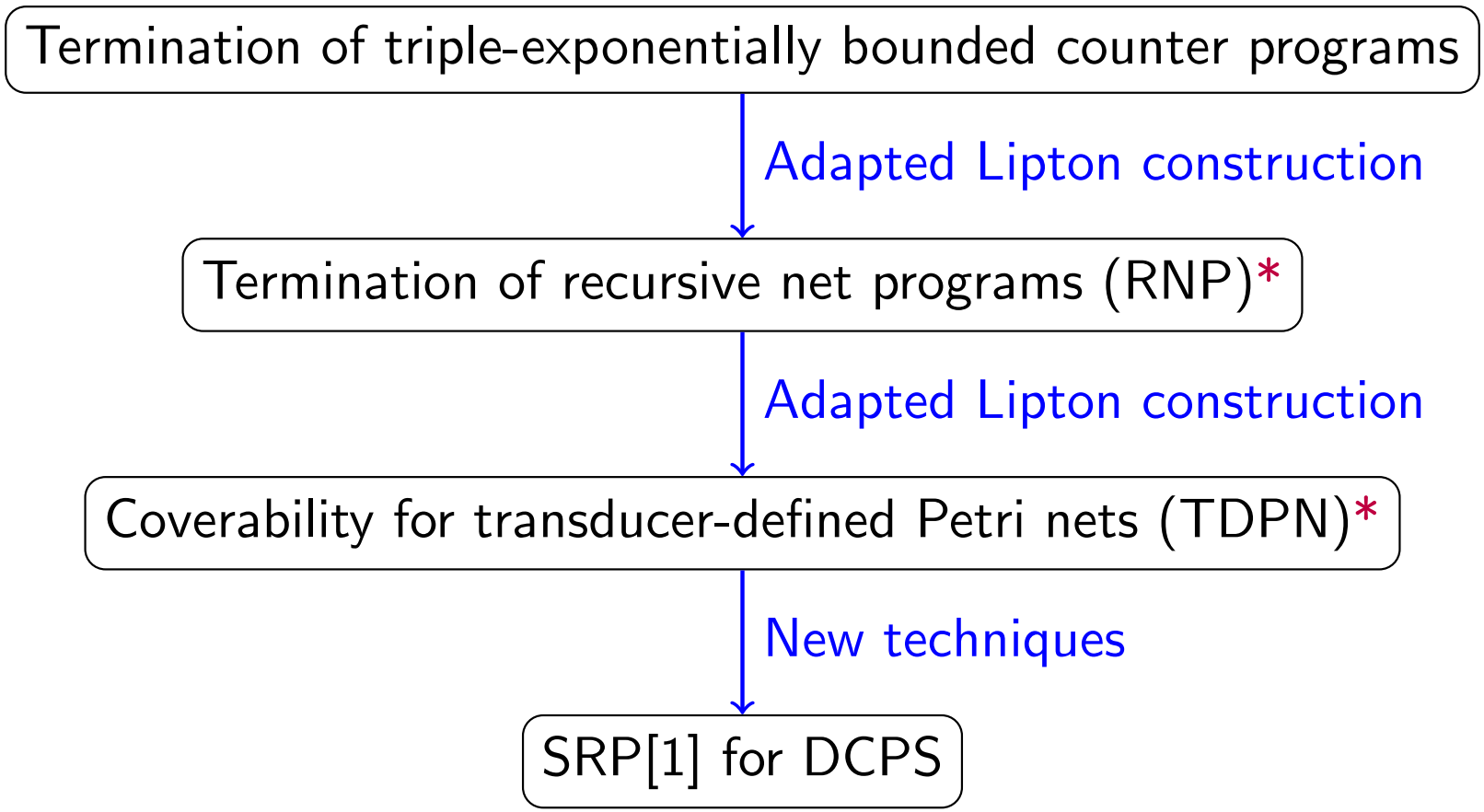
SRP[1] for DCPS

Proof Outline



*new model

Proof Outline



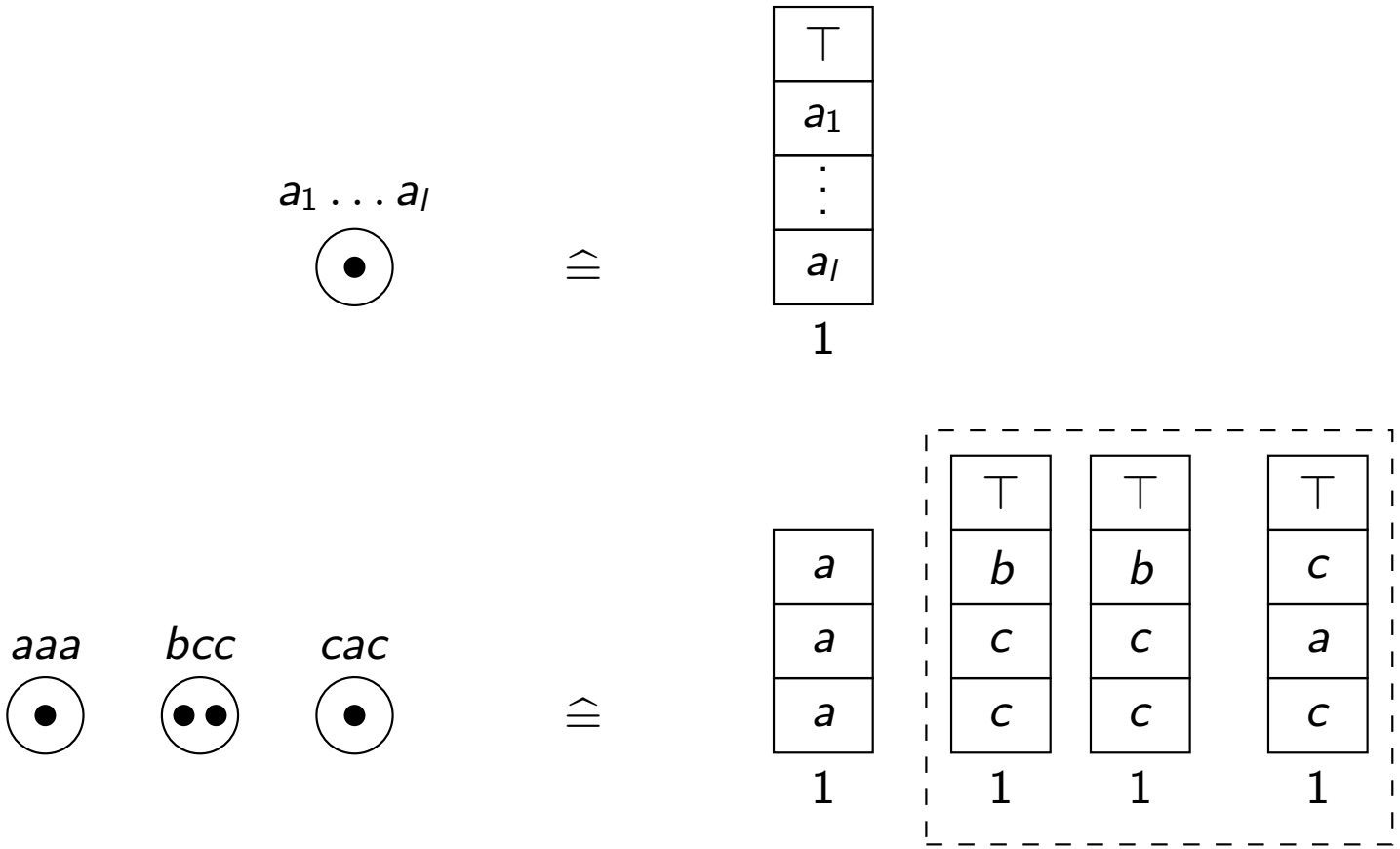
*new model

Thank you for your attention!

Any questions?

Appendix

Locking Inactive Threads



Lifting to 2EXPSPACE

We used $2^{2^d} = 2^{2^{d-1} \cdot 2} = \left(2^{2^{d-1}}\right)^2 = 2^{2^{d-1}} \cdot 2^{2^{d-1}}$.

- This means from one level to the next the bound gets squared.

$$(\cdots(2^{\overbrace{2}^{n\text{-times}}})^2 \cdots)^2 = 2^{2^n}$$

$$(\cdots(2^{\overbrace{2}^{2^n\text{-times}}})^2 \cdots)^2 = 2^{2^{2^n}}$$

Details of Known Results

Ganty and Majumdar (2012) consider threads running to completion.

- We can ensure that threads empty their stack in our model.
- This allows us to use their EXPSPACE-completeness result for $K = 0$.

Atig, Bouajjani, and Qadeer (2009) consider a slightly different DCPS:

- Each thread spawns with its parents cs-number plus 1.
- We can simulate our model in theirs using 2 more contex switches.
- Reduces our SRP[K] to their SRP[$K + 2$].
- This allows us to use their 2EXPSPACE-membership result.

Succinct Representation via Transducers

Use binary addresses $w = u.v$ for places:

- u: Role, i.e. which line, counter, or auxiliary place it is.
- v: Binary representation of recursion depth d.

Let the size of the RNP be h, the number of lines of code.

- Each counter appears in at least one line.
- Each line only needs at most one auxiliary place.
- Thus, the number of possibilities for u is linear in h.

Make the transducers distinguish each possible triple (pair) of prefixes u:

- Considering triples adds an exponent of 3, still poly in h.

Succinct Representation via Transducers

The recursion depth d changes by at most 1 at a time.

- Transducers have to check for equality or off-by-one on postfixes v.
- These checks require space linear in the number of bits.
- Since the maximum for d is 2^n, v has $\log(2^n) = n$ bits.

The triple (pair) of prefixes u tells us how the depths are related.

- Connect the paths for u with the appropriate checks at the end.

Sources I


Mohamed Faouzi Atig, Ahmed Bouajjani, and Shaz Qadeer.
Context-bounded analysis for concurrent programs with dynamic creation of threads.
In *Tools and Algorithms for the Construction and Analysis of Systems, 15th International Conference, TACAS 2009, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2009, York, UK, March 22-29, 2009. Proceedings*, pages 107–123, 2009.


Stéphane Demri, Diego Figueira, and M. Praveen.
Reasoning about data repetitions with counter systems.
In *28th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2013, New Orleans, LA, USA, June 25-28, 2013*, pages 33–42, 2013.

Sources II


Javier Esparza.
Decidability and complexity of Petri net problems – an introduction.
In G. Rozenberg and W. Reisig, editors, *Lectures on Petri Nets I: Basic Models. Advances in Petri Nets*, number 1491 in Lecture Notes in Computer Science, pages 374–428, 1998.


Pierre Ganty and Rupak Majumdar.
Algorithmic verification of asynchronous programs.
ACM Transactions on Programming Languages and Systems (TOPLAS), 34(1):6, 2012.


Alexander Kaiser, Daniel Kroening, and Thomas Wahl.
Dynamic cutoff detection in parameterized concurrent programs.
In *22nd International Conference on Computer Aided Verification, CAV 2010, Edinburgh, UK, July 15-19, 2010, Proceedings*, pages 645–659. Springer, 2010.

Sources III

Richard Lipton.
The reachability problem is exponential-space hard.
Yale University, Department of Computer Science, Report, 62, 1976.