Exploiting partial orders and symmetries in efficient analysis of message-passing concurrency

Subodh Sharma, Sept 6 2022, Presented in IARCS Verification Seminar Series

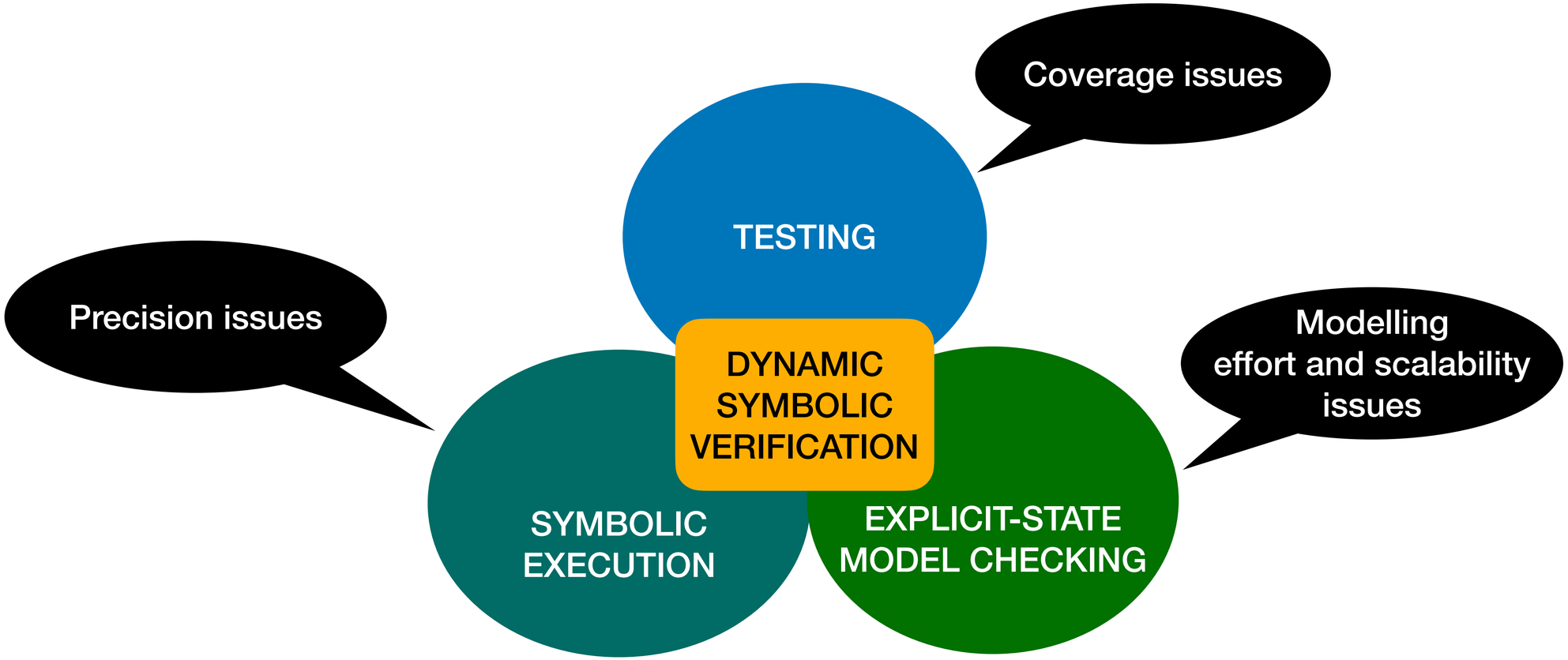
Message Passing Systems

- Heavy use in scientific computing, client-server, peer-to-peer applications, consensus protocols etc.
  - Many languages and APIs — Go, Erlang, **MPI**, etc.
- Large and complex computation code interleaved with communication
- Many issues: communication races, deadlocks, assertion violations etc.
  - Reachability is undecidable when each process is a FSM with FIFO buffers **[Brand and Zafiropulo, JACM'83]**
- Lack of use of partial orders in practical verification techniques

Soundness & Completeness

- Given a deductive system $\mathtt{D}$:
  - $\mathtt{D}$ is sound for a logic $\mathscr{L}$, if for every formula $\phi \in \mathscr{L}$, $\vdash_{\mathtt{D}} \phi \Rightarrow \vDash \phi$
    - All formulae proven by the deductive system are indeed valid!
  - $\mathtt{D}$ is complete for a logic $\mathscr{L}$, if for every formula $\phi \in \mathscr{L}$, $\vDash \phi \Rightarrow \vdash_{\mathtt{D}} \phi$
    - All valid formulae are also proven by the deductive system!

Dynamic Symbolic Verification



- **DSE: sound and complete w.r.t. a fixed input, relatively scalable**

Syntax and Matches-before Relation

$$
\begin{array}{rl}
C ::= & x := e \\
\mid & send(pid, x, h) \\
\mid & recv(pid, x, h) \\
\mid & wait(h) \\
\mid & Barrier
\end{array}
$$

$$
\begin{array}{c}
pid \in PID \\
pid^* \in PID \cup \{\star\} \\
x, h \in \mathcal{V}
\end{array}
$$

- Assume the standard expression grammar for e.
- h is the request handle.
- *wait* is a blocking call (behaves differently under buffering)
- **Matches-before** ordering relation:
  - $send(\ldots, h) \rightarrow wait(h)$; $recv(\ldots, h) \rightarrow wait(h)$
  - $send(pid, \ldots) \rightarrow send(pid, \ldots)$ (send to different destinations are unordered)
  - $wait(h) \rightarrow C$

Formal MPI Semantics

- State $s = \langle I, M \rangle$
  - I is the set of issued calls
  - M is the set of matched calls $\subseteq I$
- Issuable(x): Set of all x: $x \notin I$ and for all y with $y \prec_{po} x$, we have $y \in I$
- Ready(m, q): If for all $a \in m$ and every $s \prec_{mo} a$, we have $s \in M$
- Semantics of P is a FSM $S(P) = \langle Q, q_0, A, \delta \rangle$

Modeling Message Passing Programs

- Each process is a finite state machine (FSM)
  - Deterministic computation
  - FIFO system buffers (0 — synchronous, k/∞ — asynchronous)
  - Asynchronous non-blocking (send/recv) calls.
  - Programs have finite control and finite data domain

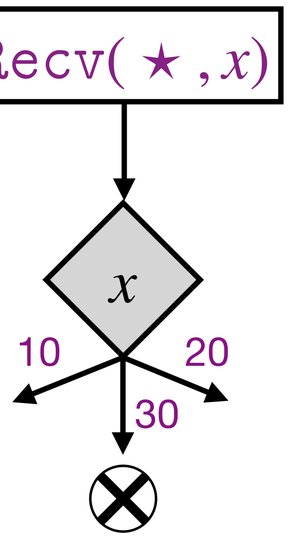
Nondeterminism: Reason for verification

```
void P1() {
  R1: Recv(*, x);
  if (x==10)
  R2: Recv(*, y);
  else if (x==20)
  R3: Recv(*, y);
  else { }
  R5: Recv(*, z);
}
```

```
void P2(){
  Send(P1, 10);
}
```

```
void P3(){
  Send(P1, 10);
}
```

```
void P4(){
  Send(P1, 30);
}
```



Multi-path program

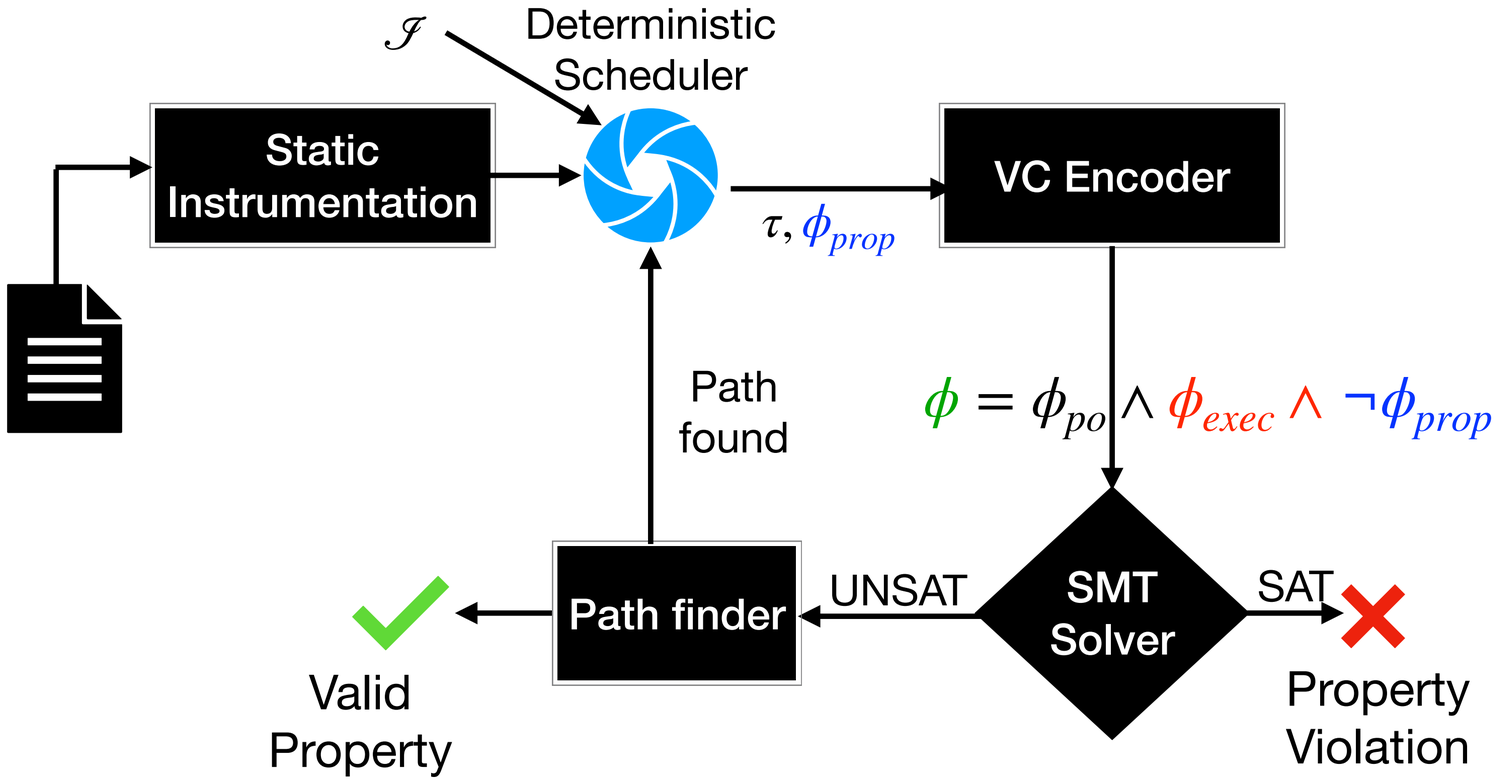
- Nondeterminism due to scheduling and control.
- Theoretical Results[TOPLAS'17, FM'18]

Complexity

The deadlock discovery problem is NP-complete for both finite and infinite buffering models.

Our technique using equivalence partitioning

- Propositional Encoding
  - Sound and complete
- Path finder:
  - nonchronological search for a new equivalence class
- Termination
  - Due to finite # of control decisions
- Validate results by concrete scheduling



Soundness & Completeness

There is a deadlocking trace *iff* there is a satisfying assignment to the variables in the encoding.

Propositional Encoding

PO Constraints

$$\bigwedge_{b \in C} \bigwedge_{a \prec b} t_{ab} \tag{1}$$

$$\bigwedge_{a,b \in C} \left(t_{ab} \rightarrow (clk_a < clk_b)\right) \tag{2}$$

Uniqueness & Clock Equality

$$\bigwedge_{\{a,b\} \in \mathbb{M}^+} \left(s_{ab} \rightarrow \bigwedge_{c \in \mathbb{M}^+(a), c \neq b} \neg s_{ac} \wedge \bigwedge_{c \in \mathbb{M}^+(b), c \neq a} \neg s_{cb}\right) \tag{3}$$

$$\bigwedge_{(a,b) \in \mathbb{M}^+} \left(s_{ab} \rightarrow (clk_a = clk_b)\right) \tag{4}$$

Match correct & Matched only

$$\bigwedge_{a \in C} \left(m_a \rightarrow \bigvee_{b \in \mathbb{M}^+(a)} s_{ba} \wedge \bigvee_{b \in \mathbb{M}^+(a)} s_{ab}\right) \tag{5}$$

$$\bigwedge_{\alpha \in \mathbb{M}^+} \left(s_\alpha \rightarrow \bigwedge_{a \in \alpha} m_a\right) \tag{6}$$

Match only issued

$$\bigwedge_{a \in C} \left(m_a \rightarrow r_a\right) \tag{7}$$

$$\bigwedge_{b \in C} \left(r_b \leftrightarrow \bigwedge_{a \prec b} m_a\right) \tag{8}$$

Not all matched

$$\bigwedge_{\alpha \in \mathbb{M}^+} \left(\bigvee_{a \in \alpha} (m_a \vee \neg r_a)\right) \tag{9}$$

$$\bigvee_{a \in C} \neg m_a \tag{10}$$

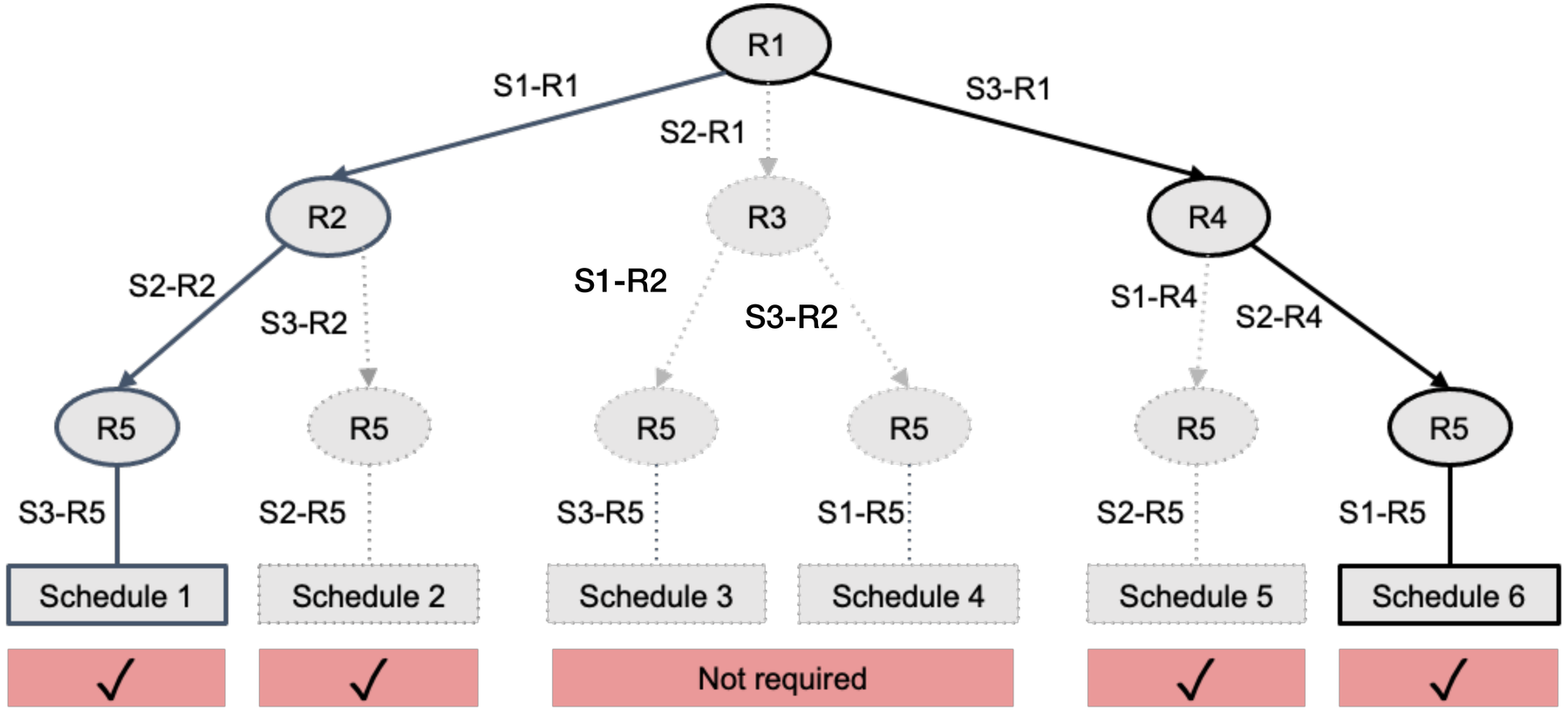
Savings through an example

```
void P1() {
   R1: Recv(*, x);
   if (x==10)
   R2: Recv(*, y);
   else if (x==20)
   R3: Recv(*, y);
   else { }
   R5: Recv(*, z);
}
```

```
void P2(){
   Send(P1, 10);
}
```

```
void P3(){
   Send(P1, 10);
}
```

```
void P4(){
   Send(P1, 30);
}
```



In general: exponential savings over runtime verification techniques

Results

B'mark	# Proc	ISP (CAV08, PPoPP08)		Aislinn Time (s) (FM17)	CIVL Time (s) (ASE15, TACAS16)	HERMES [TOPLAS17, FM18]	
		Runs	Time (s)			Runs	Time (s)
Matrix Multiply	6	720	837.64	1.33	48.07	**1**	**12.93**
	8	~1.5K	TO	2.21	258.86	**1**	**19.67**
Monte Carlo	6	120	151.60	5.03	*	**5**	**4.82**
	8	>1.2K	TO	10.17	*	**7**	**7.43**
Integrate	6	>3k	TO	27.84	157.63	**1**	**0.49**
	8	>3k	TO	TO	173.27	**1**	**0.85**

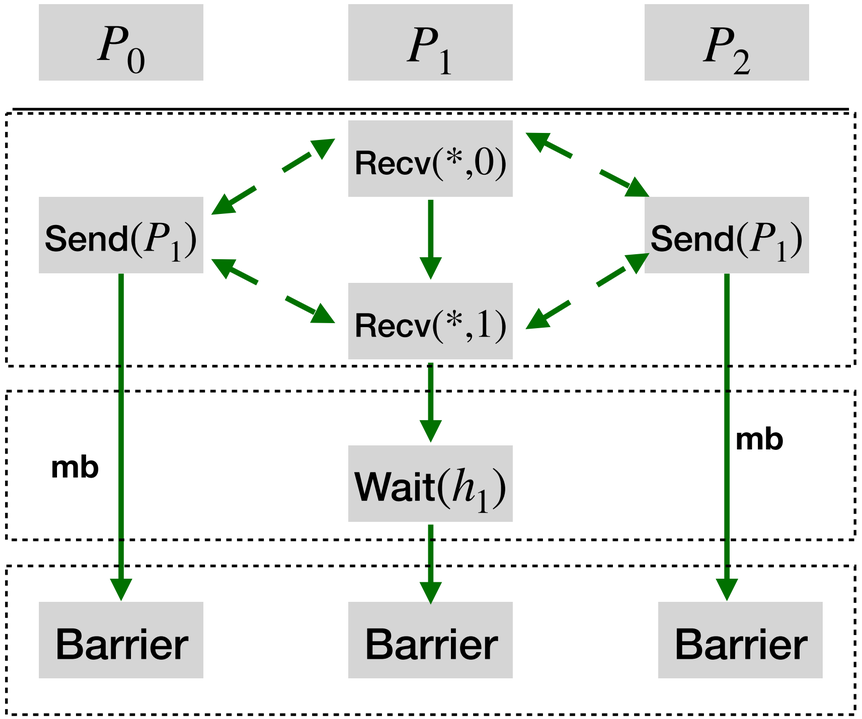
* : Benchmark not supported
TO : Time out after 1800 secs

Summary: Orders of magnitude faster than explicit-state or full symbolic verifiers

Generality: **J**ava 8 Multi-threaded Apache libraries for wait-notify deadlocks [ICSME20, ICST21]

Exploiting Symmetries: Epochs

- Problem: Solvers are stuck on exploring symmetric runs of a program
- Problem: Symmetries may not be global



Program Graph: $G = (C, E)$ s.t.

$\forall c_1, c_2 \in C, (c_1, c_2) \in E \iff c_1 \prec c_2$ and

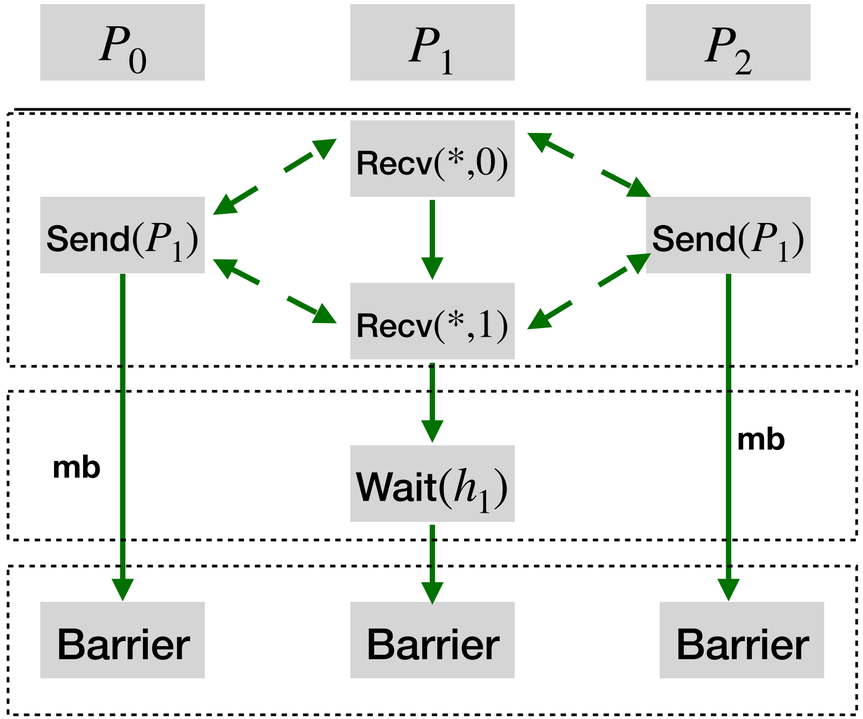
$(c_1, c_2), (c_2, c_1) \in E \iff \{c_1, c_2\} \in M$

Communication Epochs: SCCs of program graph

Detecting & Specifying Symmetries

- Automorphism $\pi : C \rightarrow C$ if $(c_1, c_2) \in E \iff (\pi(c_1), \pi(c_2)) \in E$

- Lemma: π preserves matches and deadlocks



$$\phi = \phi_{po} \wedge \phi_{exec} \wedge \neg\phi_{prop} \wedge \phi_{sbp}$$

- SBP (symmetry breaking predicate): Lex-Leader constraints

$$0 \leq i \leq n, P_i(V, W) \stackrel{\text{def}}{=} (V_{i-1} = W_{i-1} \rightarrow s_{\alpha_i} \leq s_{\beta_i})$$

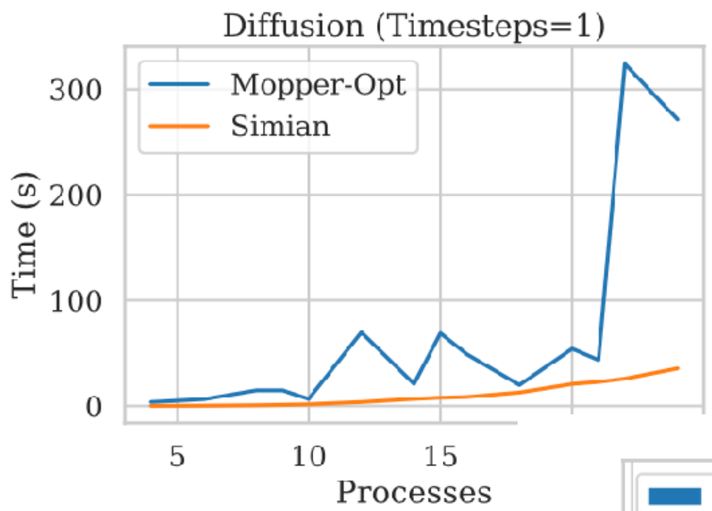
$$[\tau_1] \leq [\tau_2] \stackrel{def}{=} \bigwedge P_i([\tau_1], [\tau_2])$$

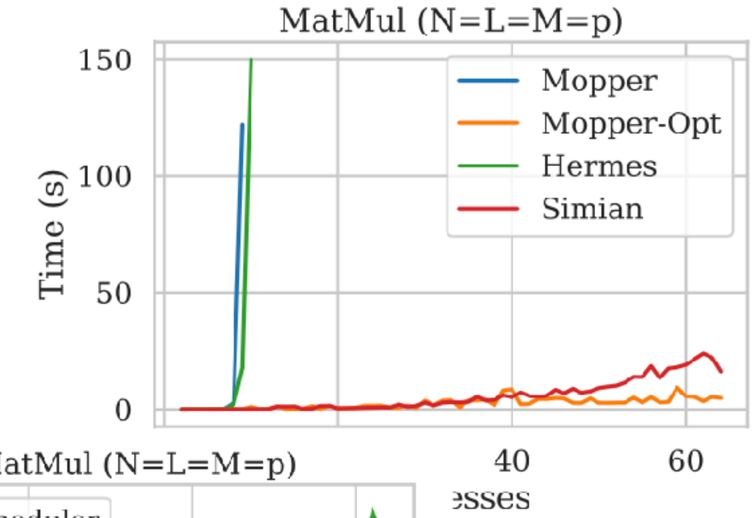
$$\phi_{sbp} = \bigwedge_{\pi \in B} [\tau] \leq [\pi(\tau)]$$

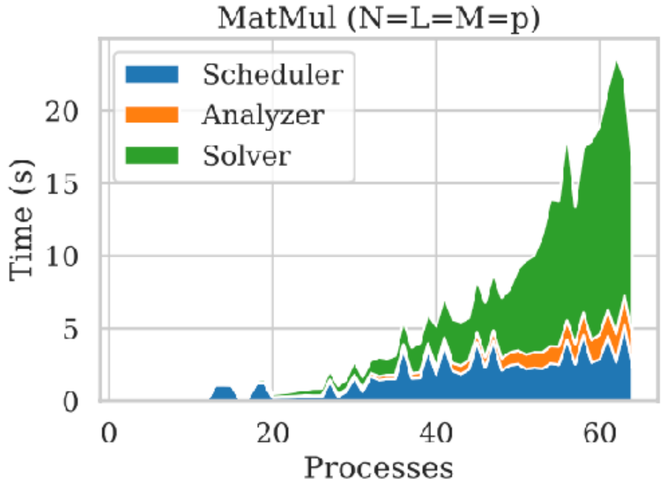
Rishabh Ranjan, Ishita Agrawal, **Subodh Sharma**:Exploiting Epochs and Symmetries in Analysing Distributed Systems. ASE 2022

Exploiting Symmetries: Results

Name vs X	X	Deadlock	Hermes	Simian
Adder	8	No	0.212	0.06
vs	16	No	TO	1.157
Processes	32	No	TO	1.912
	64	No	TO	7.257
GaussElim	8	No	0.187	**0.186**
vs	16	No	0.258	**0.283**
Processes	32	No	1.993	**2.334**
	64	No	3.912	**3.226**
Heat	8	No	0.406	**0.325**
vs	16	No	0.636	1.506
Processes	32	No	2.005	4.255
	64	No	4.464	**3.232**
HeatErrors	8	Yes	**0.309**	0.353
vs	16	Yes	0.662	**0.565**
Processes	32	Yes	2.71	**2.19**
	64	Yes	6.435	5.051
Integrate	8	No	0.209	0.06
vs	16	No	TO	**0.131**
Processes	32	No	TO	**1.854**
	64	No	TO	7.583
Diffusion	4	No	TO	**0.122**
(Timesteps=1)	8	No	TO	**0.716**
vs	16	No	TO	**8.566**
Processes	24	No	TO	**36.308**
Diffusion	2	No	TO	**0.225**
(Grid=2x2)	4	No	TO	**1.023**
vs	8	No	TO	7.197
Timesteps	16	No	TO	**67.854**
MatMul	8	No	1.815	**0.076**
(N=L=M=8)	16	No	3.032	**0.114**
vs	32	No	4.014	1.477
Processes	64	No	5.48	**2.766**
MatMul	8	No	1.918	**0.076**
(N=L=M=p)	16	No	TO	**0.224**
vs	32	No	TO	**2.822**
Processes	64	No	TO	16.175
MatMul	4	No	0.071	0.065
(p=8)	6	No	0.112	0.07
vs	8	No	1.79	**0.076**
Size (N=L=M)	12	No	TO	**4.938**







- Summary:
  - Exponential savings over the multi-path contribution
  - Negligible analysis overheads (<10%)

Thank You!

svs@cse.iitd.ac.in

https://subodhvsharma.github.io

Collaborators


Rahul P.


Dhriti


Cesar R


Daniel K


Vojtech F


Saurabh J

Ishita Agarwal
Rishabh Ranjan
Ganesh Narayanaswamy