Simplifying Verification by Layered Reasoning

Ernst-Rüdiger Olderog

&

Mani Swaminathan

CARL VON OSSIETZKY universität OLDENBURG

Observation

Parallel systems are difficult to understand and verify

a) conceptually: reasoning about interference
b) automatically: large state spaces (interleaving)

Observation

Parallel systems are difficult to understand and verify

a) conceptually: reasoning about interference
b) automatically: large state spaces (interleaving)

E.g., in the AVACS project "Beyond Timed Automata"
we use an automata-theoretic approach to verification:

$$\begin{array}{lcccc} & & COD & \text{satisfies} & DC\ ? \\ & & \downarrow & & \downarrow \\ \text{PEA:} & \mathcal{A}(CSP) \parallel \mathcal{A}(OZ) \parallel \mathcal{A}(DC) & \parallel & & \mathcal{A}_{test}(\neg DC) \end{array}$$

Is bad state of $\mathcal{A}_{test}(\neg DC)$ not reachable ?

Then model checking using various tools is applied.

Motivation

If parallel composition can be replace by
sequential composition, verification can be simplified

a) conceptually, e.g. nointerference freedom test
b) automatically by smaller state states

Explore this idea in various contexts.

Cases from program verification

Disjoint programs: Hoare (1975)

Let $S_1, \ldots, S_n$ be pairwise disjoint **while** programs.

Then parallel and sequential compostion are input/output equivalent:

$$[S_1 \| \ldots \| S_n] \equiv_{i/o} S_1 ; \ldots ; S_n.$$

Cases from program verification

Programs with restricted shared variables:

Consider **while** programs $S_1, \ldots, S_n$ and $T_1, \ldots, T_n$, where S_i is disjoint from T_j whenever $i \neq j$.

Then the programs

$$P \equiv [S_1 \,;\, T_1 \,\|\ldots\|\, S_n \,;\, T_n]$$

and

$$L \equiv [S_1 \|\ldots\| S_n] \,;\, [T_1 \|\ldots\| T_n]$$

are input/output equivalent: $P \equiv_{i/o} L$.

Elrad and Francez (1982) called the subprograms $[S_1 \|\ldots\| S_n]$ and $[T_1 \|\ldots\| T_n]$ of L layers of P.

Layered Composition

Zwiers and Janssen et al. (1991–1994)

- Layered composition $A \bullet B$:

  - independent actions are executed in parallel,
  - dependent actions are executed sequentailly

- The operator $\bullet$ is intermediate between ; and $\|$:

  $$Tr(A;B) \subseteq Tr(A \bullet B) \subseteq Tr(A \| B)$$

  Thus $\bullet$ enables:

  - parallel exceution (physical system structure)
  - sequential reasoning (logical system structure)

CCL Transformation

If there are no cross dependencies, i.e.,

$S_{i,j} \leftrightsquigarrow S_{k,l}$ for $i \neq k$ and $j \neq l$,

then

$$\begin{array}{ccccc} S_{0,0} & \| \cdots \| & S_{0,m} & & (S_{0,0} \| \cdots \| S_{0,m}) \\ \bullet & \| \cdots \| & \bullet & & \bullet \\ \vdots & \| \cdots \| & \vdots & & \vdots \\ \bullet & \| \cdots \| & \bullet & & \bullet \\ S_{n,0} & \| \cdots \| & S_{n,m} & & (S_{n,0} \| \cdots \| S_{n,m}) \end{array} \quad \equiv$$

where $\equiv$ is a strong equivalence. Janssen (1994)

Round-Based Algorithms ...

... for fault-tolerant distributed computing

Chaouch-Saad, Charron-Bost & Merz (2009)

Parallel processes with asynchronous message passing.
Each process p executes an infinite sequence of rounds.

Rounds are communication-closed:
messages are valid only in the round they were sent in.

Heard-Of model of Charron-Bost & Schiper
specifies sets of processes $HO(p, r)$ from which p receives messages in round r.

When in round r process p has received messages from all processes in $HO(p, r)$, it proceeds to round $r + 1$.

Reduction Theorem

Reduction Theorem. In the HO model
fine-grained executions F (mixing rounds in parallel)
are locally equivalent ($\approx$) to
coarse-grained executions C (proceeding in rounds).

$$
\begin{array}{lcccccccl}
F: & S_{1,0} & \| & \cdots & \| & S_{n,0} & & (S_{1,0} & \| \cdots \| & S_{n,0}) & : C \\
 & ; & & \cdots & & ; & & & ; & & \\
 & S_{1,1} & & \cdots & & S_{n,1} & \approx & (S_{1,1} & \| \cdots \| & S_{n,1}) & \\
 & ; & & \cdots & & ; & & & ; & & \\
 & . & & \cdots & & . & & . & . & . & \\
 & . & & \cdots & & . & & . & . & . &
\end{array}
$$

Application

Local properties do not compare states or round numbers of different processes.

Corollary.
If a local property holds for all C then also for all F.

Application to Consensus Algorithms:

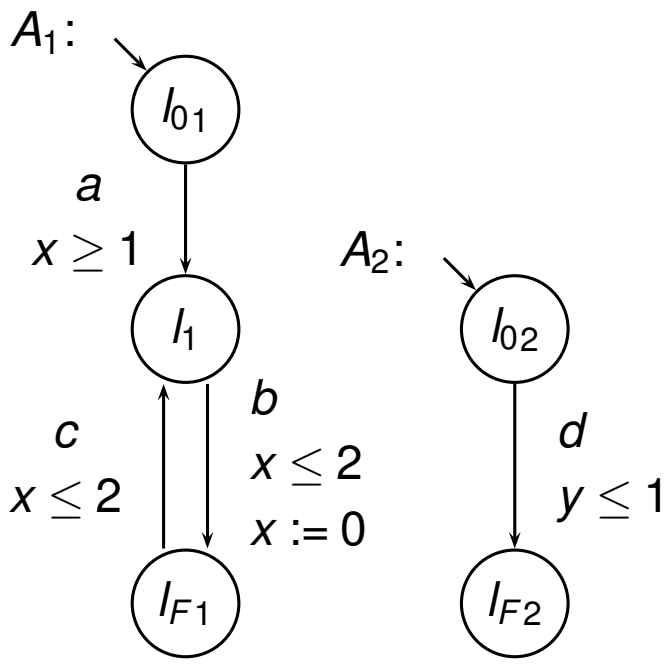
Reduction from infinite-state to finite-state models by counting rounds in C *modulo* a phase size.

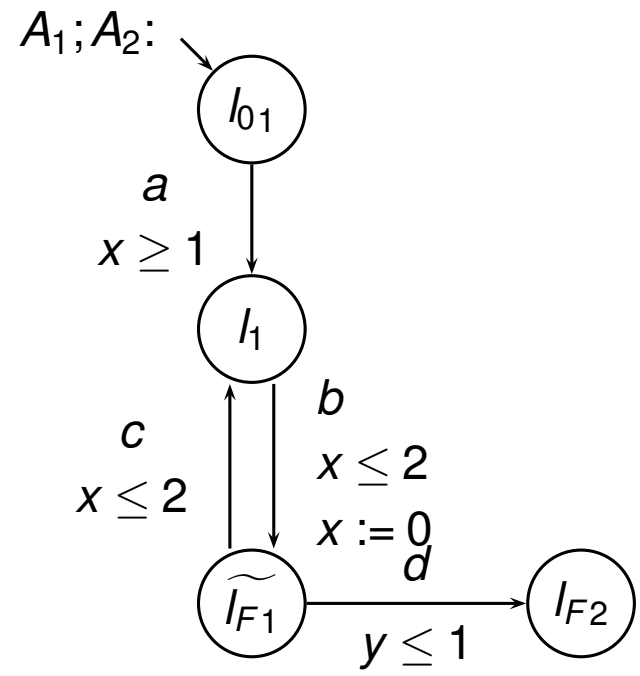
Timed Automata with Data

Olderog & Swaminathan (2010)

- Definition of Layered Composition $\bullet$ on TA
- CCL laws established in this setting
- Input/output and partial order equivalence
- Application to Collision Avoidance Protocol of B&O

Sequential Composition

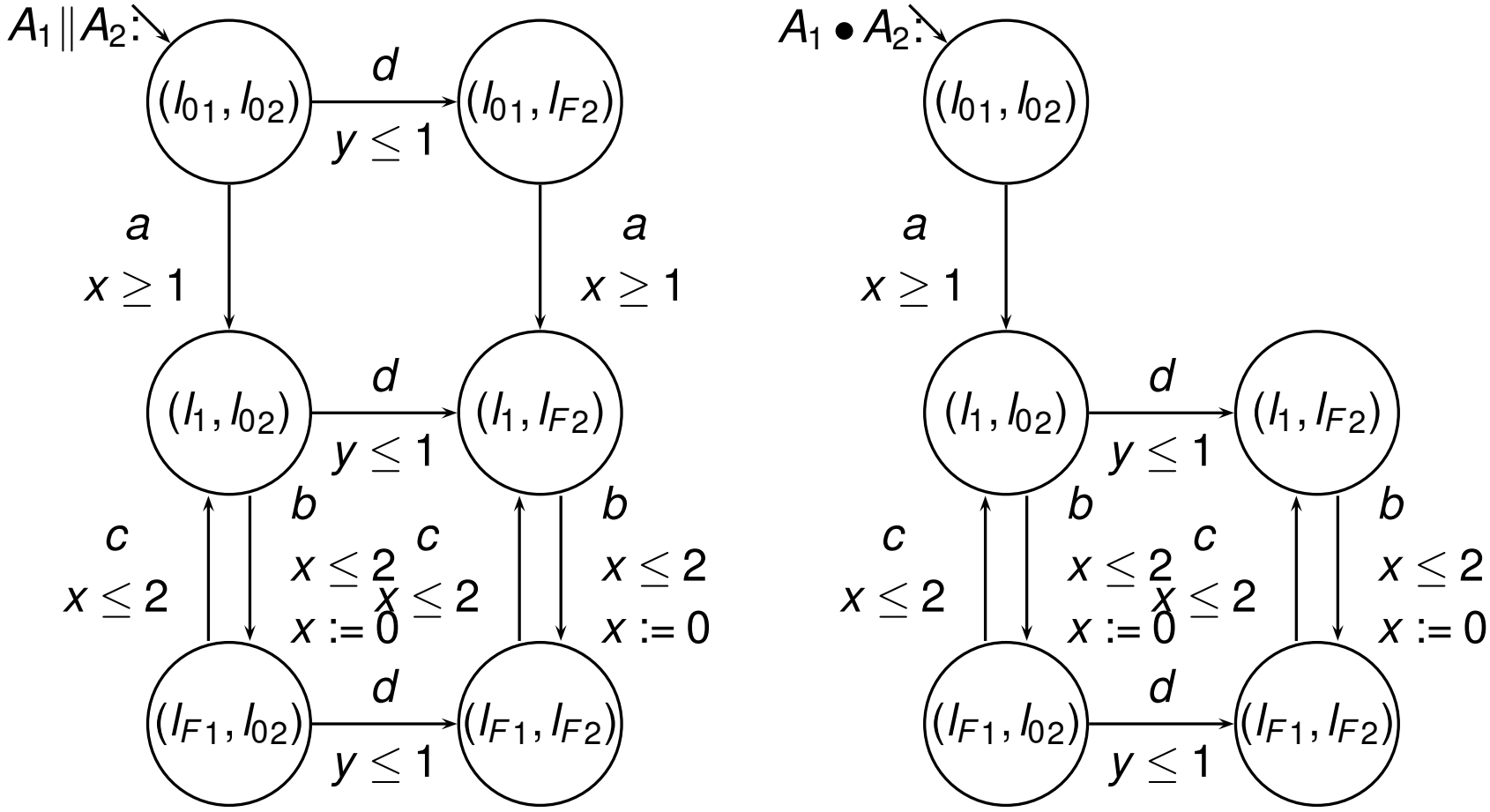




i

Parallel and Layered Composition

Assuming the dependency $a \leftrightsquigarrow d$ we obtain:



CCL Laws for TA

Theorem 1. For all timed automata A_1, A_2, B_1, B_2 with $A_1 \leftrightsquigarrow B_2$ and $A_2 \leftrightsquigarrow B_1$ the following *communication closed layer* laws hold:

1. $A_1 \bullet B_2 \equiv A_1 \| B_2$ (IND)
2. $(A_1 \bullet A_2) \| B_2 \equiv A_1 \bullet (A_2 \| B_2)$ (CCL-L)
3. $(A_1 \bullet A_2) \| B_1 \equiv (A_1 \| B_1) \bullet A_2$ (CCL-R)
4. $(A_1 \bullet A_2) \| (B_1 \bullet B_2) \equiv (A_1 \| B_1) \bullet (A_2 \| B_2)$ (CCL)

Here $\equiv$ denotes reachability equivalence, i.e., equal sets of reachable states after each iteration of the transition relation.

Layered and Sequential Composition

Repacing $\bullet$ by ;

Theorem 2. For any two TA A_1 and A_2 we have
$A_1 \bullet A_2 \equiv_{i/o} A_1 ; A_2$ and $A_1 \bullet A_2 \equiv_{po} A_1 ; A_2$.

Corollary 1. Replacing $\bullet$ by ; within the expressions in Theorem 1 yields i/o and po equivalences.

Thus (timed) LTL and CTL properties without *next* operator are preserved, see e.g. Baier & Katoen (2008).

Verification Methodology

Suppose $A_2 \leftrightsquigarrow B_1$ holds. Then

$$
\begin{array}{ll}
 & (A_1 ; A_2) \parallel B_1 \\
\equiv_{po} & \quad \{ \text{ Corollary 1 } \} \\
 & (A_1 \bullet A_2) \parallel B_1 \\
\equiv & \quad \{ \text{ CCL-R } \} \\
 & (A_1 \parallel B_1) \bullet A_2 \\
\equiv_{po} & \quad \{ \text{ Corollary 1 } \} \\
 & (A_1 \parallel B_1) ; A_2
\end{array}
$$

If each of A_1, A_2, B_1 has 10 locations
then top system has 200 locations whereas bottom system has 110 locations.

Case Study:

Layering a UPPAAL Audio/Video Protocol yields reduction by a factor of 300 (modulo reachablity).

Probabilistic Automata

Swaminathan, Katoen & Olderog (2012)

- Definition of Layered Composition $\bullet$ on PA
- CCL laws for independence and precedence established in this setting
- Input/output and partial order equivalence
- Application to Rabin's Mutual Exclusion Algorithm

Precedence $\prec$ for PA:

Given PA $\mathcal{P}_1$ and $\mathcal{P}_2$, it holds that

$$\mathcal{P}_1 \prec \mathcal{P}_2 \quad \text{iff} \quad \mathcal{P}_1 \| \mathcal{P}_2 \equiv_{TD} \mathcal{P}_1 ; \mathcal{P}_2 .$$

CCL Laws for PA

For all probabilistic automata $\mathcal{P}_1$, $\mathcal{P}_2$, Q_1, Q_2
with ($\mathcal{P}_1 \leftrightsquigarrow Q_2$ or $\mathcal{P}_1 \prec Q_2$) and ($Q_1 \leftrightsquigarrow \mathcal{P}_2$ or $Q_1 \prec \mathcal{P}_2$)
the following CCL laws hold:

1. $\mathcal{P}_1 \bullet Q_2 \equiv_{TD} \mathcal{P}_1 \| Q_2$ (IND)
2. $(\mathcal{P}_1 \bullet \mathcal{P}_2) \| Q_2 \equiv_{TD} \mathcal{P}_1 \bullet (\mathcal{P}_2 \| Q_2)$ (CCL-L)
3. $(\mathcal{P}_1 \bullet \mathcal{P}_2) \| Q_1 \equiv_{TD} (\mathcal{P}_1 \| Q_1) \bullet \mathcal{P}_2$ (CCL-R)
4. $(\mathcal{P}_1 \bullet \mathcal{P}_2) \| (Q_1 \bullet Q_2) \equiv_{TD} (\mathcal{P}_1 \| Q_1) \bullet (\mathcal{P}_2 \| Q_2)$ (CCL)

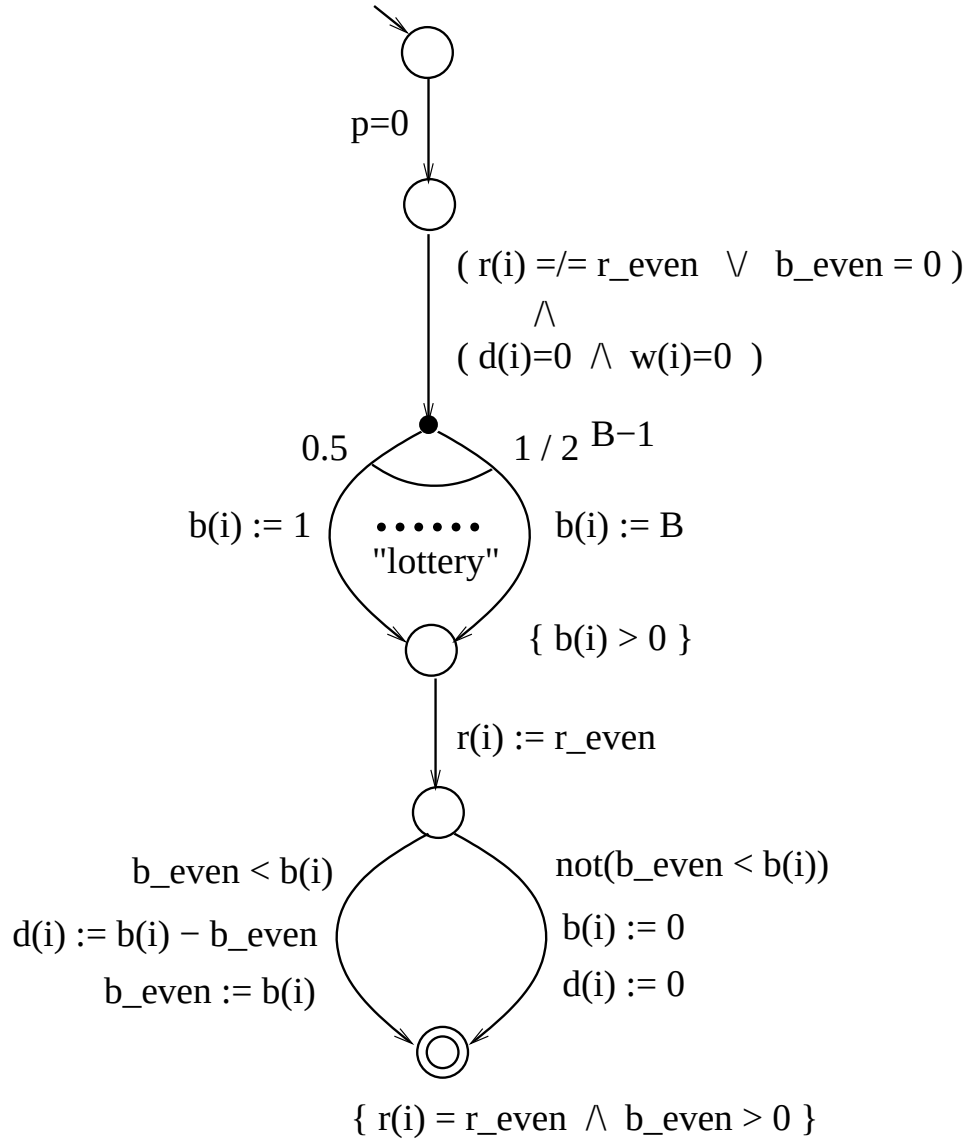
Randomized Mutual Exclusion

Original algorithm by Rabin (1982) found erroneous by Saias (1992).
Revised algorithm by Kushilevitz & Rabin (1992):

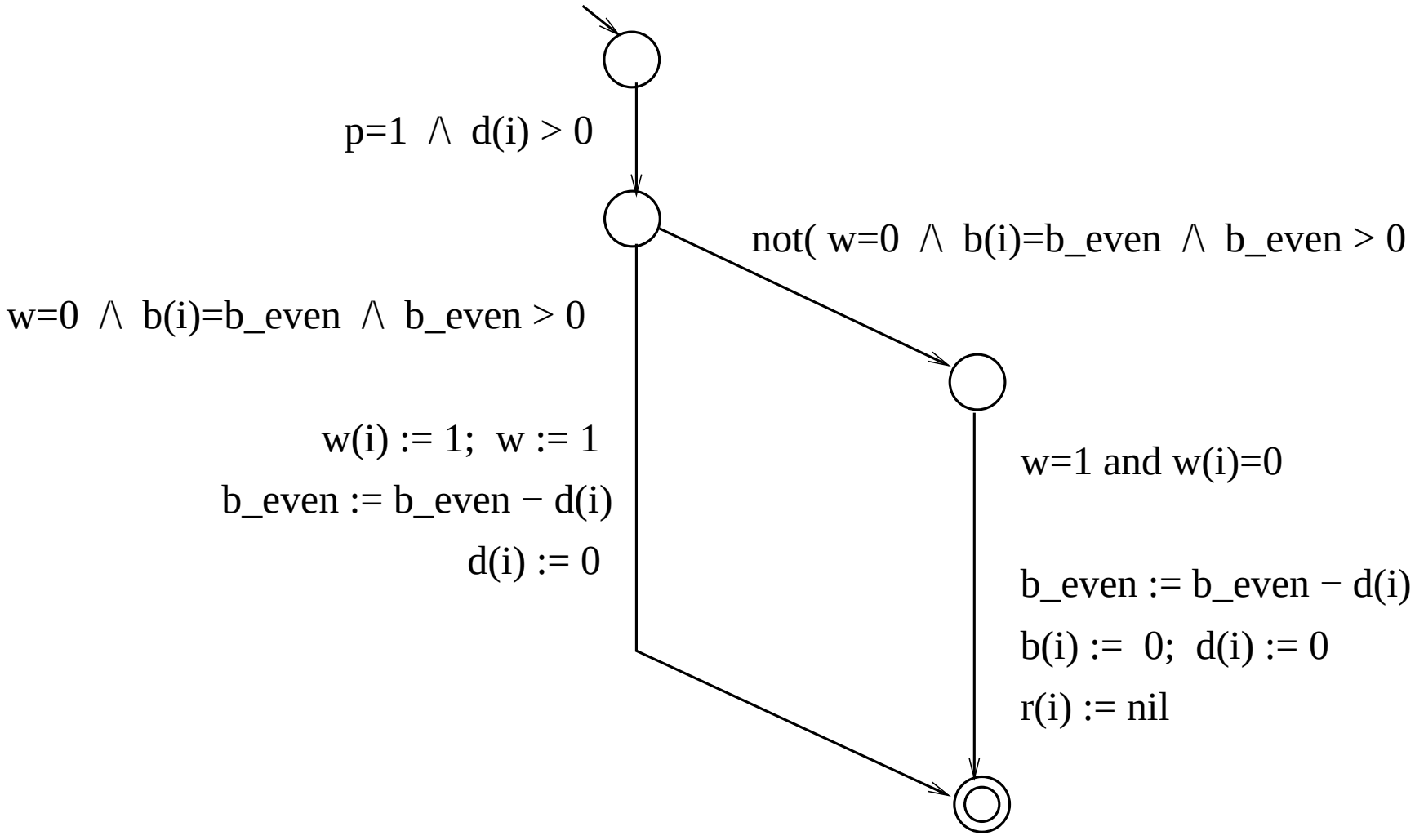
$$MUTEX = (\mathcal{P}_1 \,||\, \ldots \,||\, \mathcal{P}_N) \text{ with } P_i = \left(\begin{array}{c} odd\ \mathcal{V}_i + I\mathcal{V}_i + even\ \mathcal{C}_i \\ ; \\ odd\ \mathcal{T}_i + I\mathcal{T}_i + \mathcal{R}_i \\ ; \\ even\ \mathcal{V}_i + I\mathcal{V}_i + odd\ \mathcal{C}_i \\ ; \\ even\ \mathcal{T}_i + I\mathcal{T}_i + \mathcal{R}_i \end{array} \right)^*$$

McIver, Gonzalia, Cohen, and Morgan (2008) analysed parts of the algorithm in terms of probabilistic Kleene Algebra.

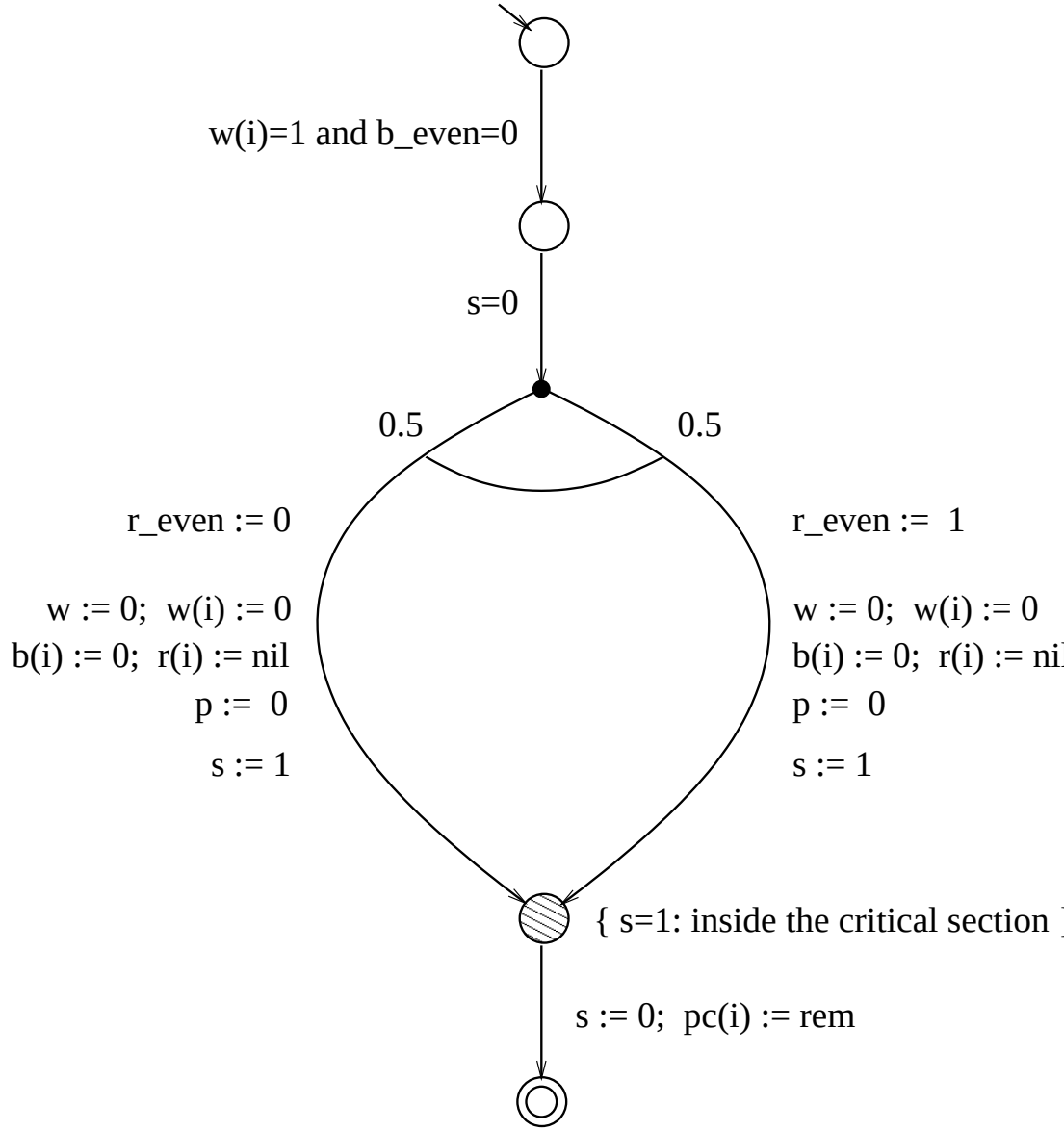
PA for even Voting $\mathcal{V}_i$



PA for even notification $\mathcal{T}_i$



PA for even critical section C_i



Unfolding up to Round 4

$$
\begin{array}{lcccc}
\text{odd } (r=1): & \{p=1\}\mathcal{V}_1^1 + I\mathcal{V}_1^1 + \mathcal{C}_1^0 & \mathcal{V}_2^1 + I\mathcal{V}_2^1 + \mathcal{C}_2^0 & \cdots & \mathcal{V}_N^1 + I\mathcal{V}_N^1 + \mathcal{C}_N^0 \\
 & ; & ; & & ; \\
 & \{p=0\}\mathcal{T}_1^1 + I\mathcal{T}_1^1 + \mathcal{R}_1^0 & \mathcal{T}_2^1 + I\mathcal{T}_2^1 + \mathcal{R}_2^0 & \cdots & \mathcal{T}_N^1 + I\mathcal{T}_N^1 + \mathcal{R}_N^0 \\
 & ; & ; & & ; \\
\text{even } (r=2): & \{p=0\}\mathcal{V}_1^2 + I\mathcal{V}_1^2 + \mathcal{C}_1^1 & \mathcal{V}_2^2 + I\mathcal{V}_2^2 + \mathcal{C}_2^1 & \cdots & \mathcal{V}_N^2 + I\mathcal{V}_N^2 + \mathcal{C}_N^1 \\
 & ; & ; & & ; \\
 & \{p=1\}\mathcal{T}_1^2 + I\mathcal{T}_1^2 + \mathcal{R}_1^1 & \mathcal{T}_2^2 + I\mathcal{T}_2^2 + \mathcal{R}_2^1 & \cdots & \mathcal{T}_N^2 + I\mathcal{T}_N^2 + \mathcal{R}_N^1 \\
 & ; & ; & & ; \\
\text{odd } (r=3): & \{p=1\}\mathcal{V}_1^3 + I\mathcal{V}_1^3 + \mathcal{C}_1^2 & \mathcal{V}_2^3 + I\mathcal{V}_2^3 + \mathcal{C}_2^2 & \cdots & \mathcal{V}_N^3 + I\mathcal{V}_N^3 + \mathcal{C}_N^2 \\
 & ; & ; & & ; \\
 & \{p=0\}\mathcal{T}_1^3 + I\mathcal{T}_1^3 + \mathcal{R}_1^2 & \mathcal{T}_2^3 + I\mathcal{T}_2^3 + \mathcal{R}_2^2 & \cdots & \mathcal{T}_N^3 + I\mathcal{T}_N^3 + \mathcal{R}_N^2 \\
 & ; & ; & & ; \\
\text{even } (r=4): & \{p=0\}\mathcal{V}_1^4 + I\mathcal{V}_1^4 + \mathcal{C}_1^3 & \mathcal{V}_2^4 + I\mathcal{V}_2^4 + \mathcal{C}_2^3 & \cdots & \mathcal{V}_N^4 + I\mathcal{V}_N^4 + \mathcal{C}_N^3 \\
 & ; & ; & & ; \\
 & \{p=1\}\mathcal{T}_1^4 + I\mathcal{T}_1^4 + \mathcal{R}_1^3 & \mathcal{T}_2^4 + I\mathcal{T}_2^4 + \mathcal{R}_2^3 & \cdots & \mathcal{T}_N^4 + I\mathcal{T}_N^4 + \mathcal{R}_N^3
\end{array}
$$

Analysis of unfolded *MUTEX*

The phases of unfolded *MUTEX* satisfy the following independence and precedence conditions, where $r, r1, r2 > 0$ and $i, j \in \{1, \ldots, N\}$:

1. Remainder $\mathcal{R}_i^r$, idle voting $I\mathcal{V}_i^r$ and idle notif. $I\mathcal{T}_i^r$ are *independent* ($\leftrightsquigarrow$) of any other phase.
2. If one of $r1$ and $r2$ is *even* and the other *odd* then

   $\mathcal{V}_i^{r1} \leftrightsquigarrow \mathcal{V}_j^{r2}$, $\mathcal{V}_i^{r1} \leftrightsquigarrow \mathcal{T}_j^{r2}$, $\mathcal{T}_i^{r1} \leftrightsquigarrow \mathcal{V}_j^{r2}$, $\mathcal{T}_i^{r1} \leftrightsquigarrow \mathcal{T}_j^{r2}$.
3. A critical section $\mathcal{C}_i^r$ can only start if notification $\mathcal{T}_i^r$ has been completed, which in turn can only start if voting $\mathcal{V}_i^r$ has been completed:

   $\mathcal{V}_i^r \prec \mathcal{T}_i^r$ and $\mathcal{T}_i^r \prec \mathcal{C}_i^r$.
4. – 6. $\cdots$

Layered Unfolding up to Round 4

$$(\mathcal{V}_1^1 + I\mathcal{V}_1^1 + \mathcal{C}_1^0 \parallel \mathcal{V}_2^1 + I\mathcal{V}_2^1 + \mathcal{C}_2^0 \parallel \cdots \parallel \mathcal{V}_N^1 + I\mathcal{V}_N^1 + \mathcal{C}_N^0)$$

;

$$(\mathcal{T}_1^1 + I\mathcal{T}_1^1 + \mathcal{R}_1^0 \parallel \mathcal{T}_2^1 + I\mathcal{T}_2^1 + \mathcal{R}_2^0 \parallel \cdots \parallel \mathcal{T}_N^1 + I\mathcal{T}_N^1 + \mathcal{R}_N^0)$$

;

$$(\mathcal{V}_1^2 + I\mathcal{V}_1^2 + \mathcal{C}_1^1 \parallel \mathcal{V}_2^2 + I\mathcal{V}_2^2 + \mathcal{C}_2^1 \parallel \cdots \parallel \mathcal{V}_N^2 + I\mathcal{V}_N^2 + \mathcal{C}_N^1)$$

;

$$(\mathcal{T}_1^2 + I\mathcal{T}_1^2 + \mathcal{R}_1^1 \parallel \mathcal{T}_2^2 + I\mathcal{T}_2^2 + \mathcal{R}_2^1 \parallel \cdots \parallel \mathcal{T}_N^2 + I\mathcal{T}_N^2 + \mathcal{R}_N^1)$$

;

$$(\mathcal{V}_1^3 + I\mathcal{V}_1^3 + \mathcal{C}_1^2 \parallel \mathcal{V}_2^3 + I\mathcal{V}_2^3 + \mathcal{C}_2^2 \parallel \cdots \parallel \mathcal{V}_N^3 + I\mathcal{V}_N^3 + \mathcal{C}_N^2)$$

;

$$(\mathcal{T}_1^3 + I\mathcal{T}_1^3 + \mathcal{R}_1^2 \parallel \mathcal{T}_2^3 + I\mathcal{T}_2^3 + \mathcal{R}_2^2 \parallel \cdots \parallel \mathcal{T}_N^3 + I\mathcal{T}_N^3 + \mathcal{R}_N^2)$$

;

$$(\mathcal{V}_1^4 + I\mathcal{V}_1^4 + \mathcal{C}_1^3 \parallel \mathcal{V}_2^4 + I\mathcal{V}_2^4 + \mathcal{C}_2^3 \parallel \cdots \parallel \mathcal{V}_N^4 + I\mathcal{V}_N^4 + \mathcal{C}_N^3)$$

;

$$(\mathcal{T}_1^4 + I\mathcal{T}_1^4 + \mathcal{R}_1^3 \parallel \mathcal{T}_2^4 + I\mathcal{T}_2^4 + \mathcal{R}_2^3 \parallel \cdots \parallel \mathcal{T}_N^4 + I\mathcal{T}_N^4 + \mathcal{R}_N^3)$$

Layered MUTEX

Under the assumption of fairness the parallel *MUTEX* is po equivalent to the following layered version:

$$
\textit{layered\_MUTEX} = \left(
\begin{array}{c}
(\textit{odd}\ \mathcal{V}_1 + I\mathcal{V}_1 + \textit{even}\ \mathcal{C}_1 \parallel \cdots \parallel \textit{odd}\ \mathcal{V}_N + I\mathcal{V}_N + \textit{even}\ \mathcal{C}_N) \\
; \\
(\textit{odd}\ \mathcal{T}_1 + I\mathcal{T}_1 + \mathcal{R}_1 \parallel \cdots \parallel \textit{odd}\ \mathcal{T}_N + I\mathcal{T}_N + \mathcal{R}_N) \\
; \\
(\textit{even}\ \mathcal{V}_1 + I\mathcal{V}_1 + \textit{odd}\ \mathcal{C}_1 \parallel \cdots \parallel \textit{even}\ \mathcal{V}_N + I\mathcal{V}_N + \textit{odd}\ \mathcal{C}_N) \\
; \\
(\textit{even}\ \mathcal{T}_1 + I\mathcal{T}_1 + \mathcal{R}_1 \parallel \cdots \parallel \textit{even}\ \mathcal{T}_N + I\mathcal{T}_N + \mathcal{R}_N)
\end{array}
\right)^*
$$

Reduction in Locations

	number of locations in ...		
N	... *MUTEX*	... *layered_MUTEX*	reduction factor
3	15,625	1,453	≥ 10
4	390,625	10,781	≥ 36
5	9,765,625	81,085	≥ 120

Conclusion

Semantic methods, e.g., layering transformations, simplify system verification.

References

- T. Elrad & N. Francez,
  Decomposition of Distributed Programs into Communication Closed Layers.
  Science of Computer Programming 2 (1982) 155–173
- F.A. Stomp & W.P. de Roever,
  A Principle for Sequential Reasoning about Distributed Algorithms.
  Formal Aspects of Computing 6 (1994) 716–737
- W. Janssen,
  Layered Design of Parallel Systems.
  PhD Dissertation, Universiteit Twente (1994)

Refs cont'd

- B. Charron-Bost & A. Schiper,
  The Heard-Of Model:
  Computing in Distributed Systems with Benign Faults.
  Distributed Computing 22 (2009) 49–71
- M. Chaouch-Saad, B. Charron-Bost & S. Merz,
  A Reduction Theorem for the Verification of
  Round-Based Distributed Algorithms.
  In *Proc. Reachability Problems*, LNCS 5797 (2009) 93–106
- B. Charron-Bost & S. Merz,
  Formal Verification of a Consensus Algorithm in the Heard-Of Model.
  Int. J Software Informatics 3 (2009) 273–303

Refs cont'd

- E.-R. Olderog & M. Swaminathan,
  Layered Composition for Timed Automata.
  In *Proc. FORMATS*, LNCS 6246 (2010) 228–242
- K. Havelund, A. Skou, K. G. Larsen & K. Lund,
  Formal modeling and analysis of an audio/video protocol:
  an industrial case study using UPPAAL.
  In *Proc. IEEE RTSS* (1997) 2–13
- M. Swaminathan, J.-P. Katoen & E.-R. Olderog,
  Layered Reasoning for Randomized Distributed Algorithms.
  Formal Aspects of Computing 24 (2012) 477–496

Refs cont'd

- M. O, Rabin,
  n-process mutual exclusion with bounded waiting
  by $4 \log n$ shared variables.
  J. Comput. Syst. Sci. 25 (1982) 66–75

- I. Saias,
  Proving probabilistic correctness statements:
  the case of Rabin's algorithm for mutual exclusion.
  In: *Proc. ACM PODC* (1992) 263–274

- E. Kushilevitz & M. O. Rabin,
  Randomized mutual exclusion algorithms revisited.
  In: Proc. ACM PODC (1992) 275–283

- A. K. McIver, C. Gonzalia, E. Cohen & C. C. Morgan,
  Using probabilistic Kleene algebra pKA for protocol verification.
  J. Log. Algebr. Program. 76 (2008) 90–111