Adam: Causality-Based Synthesis of Distributed Systems

Bernd Finkbeiner, Manuel Gieseking, Ernst-Rüdiger Olderog

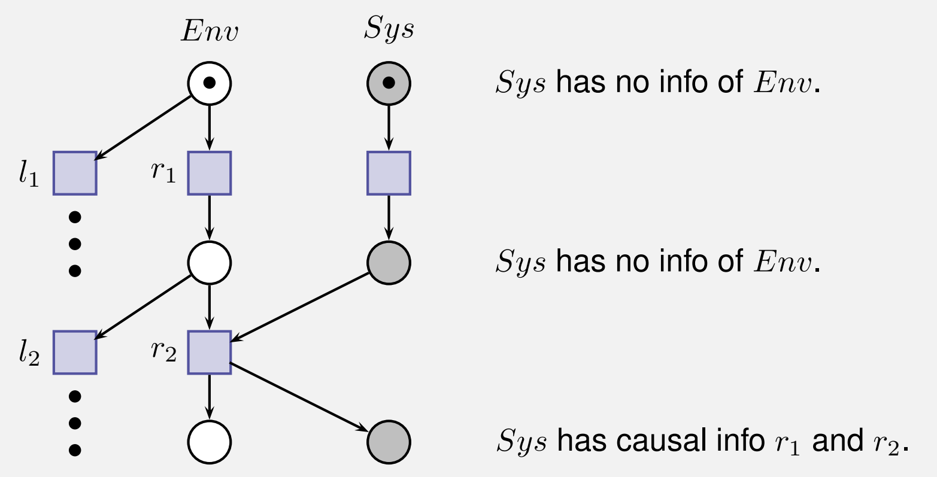
Adam*

Adam is a tool for the automatic synthesis of distributed systems with multiple concurrent processes. For each process, a local controller is synthesized that acts exclusively on locally available information, obtained through synchronization with the environment and with other system processes. Adam implements the first symbolic game solving algorithm for Petri games. Adam has been applied in case studies with up to 38 system processes.

* Adam is named in honor of Carl Adam Petri (1926–2010).

Petri Games

- Extension of Petri nets to multi-player games.
- Each token is a player.
- Players have causal memory:



Sys has no info of Env.

Sys has no info of Env.

Sys has causal info r_1 and r_2.

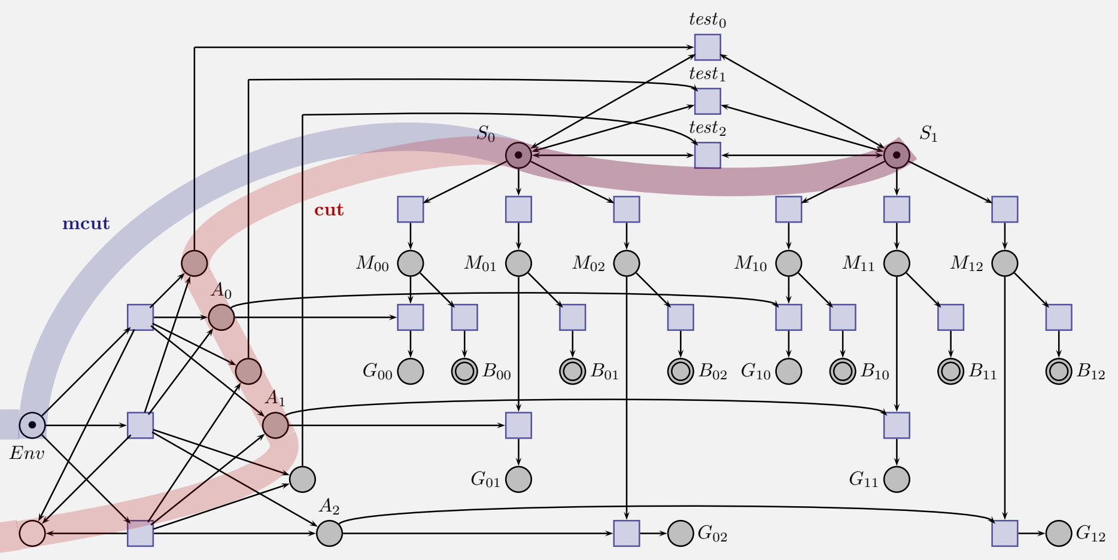
- Solving takes only* single-exponential time.

* For n system players, 1 environment player, local safety objective. Compare: Reachability in 1-safe Petri nets is already PSPACE-complete.

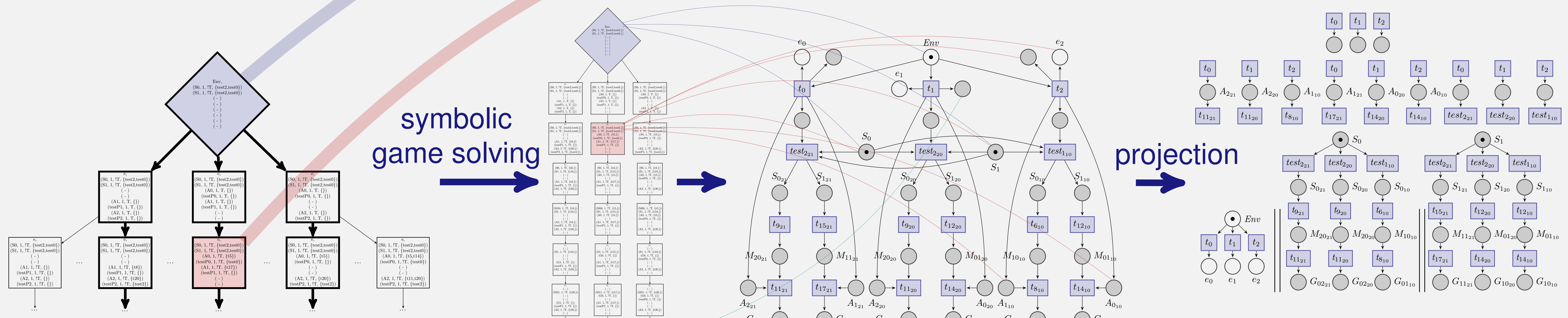
Finkbeiner & Olderog (2014): Petri Games: Synthesis of Distributed Systems with Causal Memory.

Example: Concurrent Machines (CM)

Synthesis of robot controllers in a production plant: the robots execute k manufacturing orders on n concurrent machines in a hostile environment, which may disable an arbitrary machine. Petri game model for $k = 2, n = 3$:



Solving Petri Games



symbolic game solving

projection

2-player game over finite graph

Strategy for Player$_\square$

Petri game strategy for system players

Distributed controllers

Benchmarks

CM: see above.

SR: Self-reconfiguration of n robots on which the environment destroys up to k tools.

JP: Processing of a job by a subset of n processors chosen by the environment.

DWs: Workflow of a document among n clerks starting at a clerk selected by the environment.

Size of the game: *#Tok* - number of token, *#Var* - number of BDD variables, *#P* - number of places, *#T* - number of transitions.
Size of the strategy: $\#P_s$ - number of places, $\#T_s$ - number of transitions (if a Petri game strategy exists).

Ben.	n/k	$\#Tok$	$\#Var$	$\#P$	$\#T$	time (s)	memory (GB)	$\#P_s$	$\#T_s$
CM	2/1	6	66	13	10	1.4	0.31	14	8
	2/2	7	96	18	16	1.3	0.29	-	-
	2/3	8	126	23	22	1.7	0.30	-	-
			. . .			. . .		. . .	
	4/3	12	224	41	44	14.4	0.8	68	32
	4/4	13	276	50	56	155.3	4.27	-	-
	5/1	12	146	28	25	4.0	0.38	62	20
	5/2	13	208	39	40	24.3	0.8	78	30
	5/3	14	270	50	55	468.3	3.5	94	40
	6/1	14	172	33	30	19.6	0.8	86	24
	6/2	15	244	46	48	1042.2	2.51	105	36
SR	2/1	5	86	18	17	1.3	0.29	32	16
	2/2	6	116	24	26	1.6	0.29	-	-
	2/3	7	144	30	35	4.4	0.39	-	-
	2/4	8	174	36	44	42.7	0.8	-	-
	3/1	6	204	34	49	1155.6	10.05	79.7	45
JP	2	3	46	12	13	1.1	0.31	16	13
	3	4	76	18	23	1.8	0.31	34	28
	4	5	112	25	35	1.5	0.29	62	50
	5	6	160	33	49	2.0	0.31	102	80
			. . .			. . .		. . .	
	9	10	486	75	125	55.3	2.85	422	300
	10	11	612	88	149	146.9	5.43	552	385
	11	12	762	102	175	434.8	16.62	706	484
DWs	1	3	36	11	6	0.8	0.31	8	3
	2	5	70	21	12	1.6	0.31	23	10
	3	7	102	31	18	1.4	0.30	46	21
			. . .			. . .		. . .	
	16	33	524	161	96	547.7	9.07	1073	528
	17	35	556	171	102	789.1	12.02	1208	595
	18	37	588	181	108	1027.3	11.94	1351	666
	19	39	620	191	114	1451.9	15.99	1502	741

'-' means no winning strategy exists.

Related Work: Other Frameworks for Distributed Synthesis

Pnueli-Rosner model: synchronous concurrency with partial observation of shared variables

- distributed synthesis: in general undecidable
- pipelines and rings: nonelementary

A. Pnueli & R. Rosner (1990): Distributed Reactive Systems are Hard to Synthesize.
B. Finkbeiner & S. Schewe (2005): Uniform Distributed Synthesis.

Zielonka automata: asynchronous concurrency with shared actions and causal memory

- distributed synthesis: in general decidability open
- tree architectures: nonelementary

Gastin, Lerman & Zeitoun (2004): Distributed Games with Causal Memory are Decidable for Series-Parallel Systems.
Genest, Gimbert, Muscholl & Walukiewicz (2013): Asynchronous Games over Tree Architectures.

Correct System Design
University of Oldenburg

Bernd Finkbeiner
finkbeiner@react.uni-saarland.de
Manuel Gieseking
manuel.gieseking@informatik.uni-oldenburg.de
Ernst-Rüdiger Olderog
olderog@informatik.uni-oldenburg.de

Carl von Ossietzky universität Oldenburg
Albert-Ludwigs-Universität Freiburg
Universität des Saarlandes

Reactive Systems

DFG: SFB / TR 14 AVACS