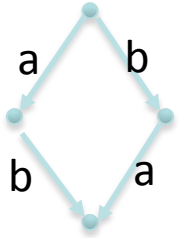
Petri Net (versus) State Spaces

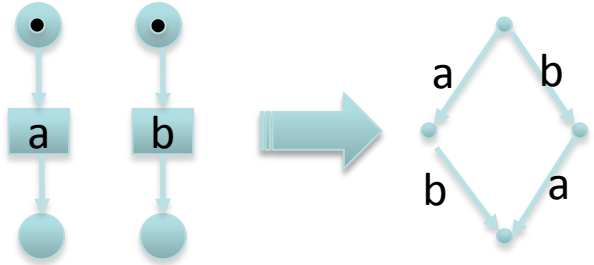
Karsten Wolf

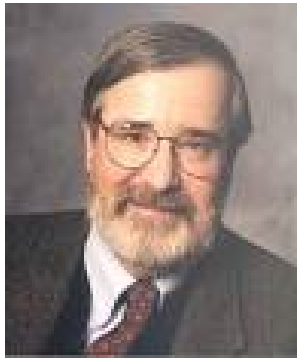
Why



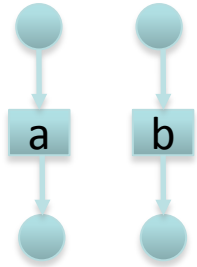
?

State Space:

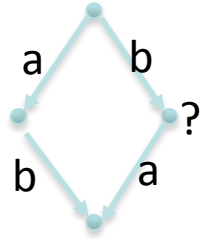




Why



?

Why 

- Asynchronous communication !

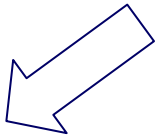
-Global changes → Local changes !

-Sequential order → Causality !

Petri net principles

Monotonicity of firing

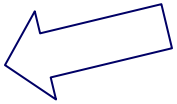
Variables → Ressources



Linearity of firing rule

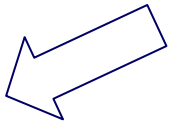
- Asynchronous communication !

Locality



-Global changes → Local changes !

Partially ordered
event structures



-Sequential order → Causality !

Petri net specific verification

Monotonicity of firing

Coverability graphs ... available in LoLA

Linearity of firing rule

Invariants ... used in LoLA

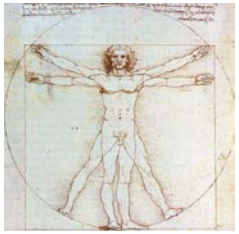
Locality

Net reduction

Partially ordered event structures

Branching prefixes

Explicit State Reduction Techniques in the LoLA tool



Symmetry

Sweep-Line

Coverability

PN Structure Theory

Partial Order

1. PN Structure Theory

- The Petri net state equation:

If $m \rightarrow^* m'$ then $N x = (m-m')$ has a solution

Tool Sara:

Search *state* space

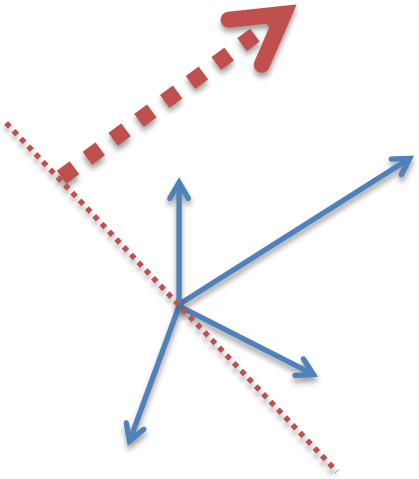
→

Search *solution* space

2. The sweep-line method

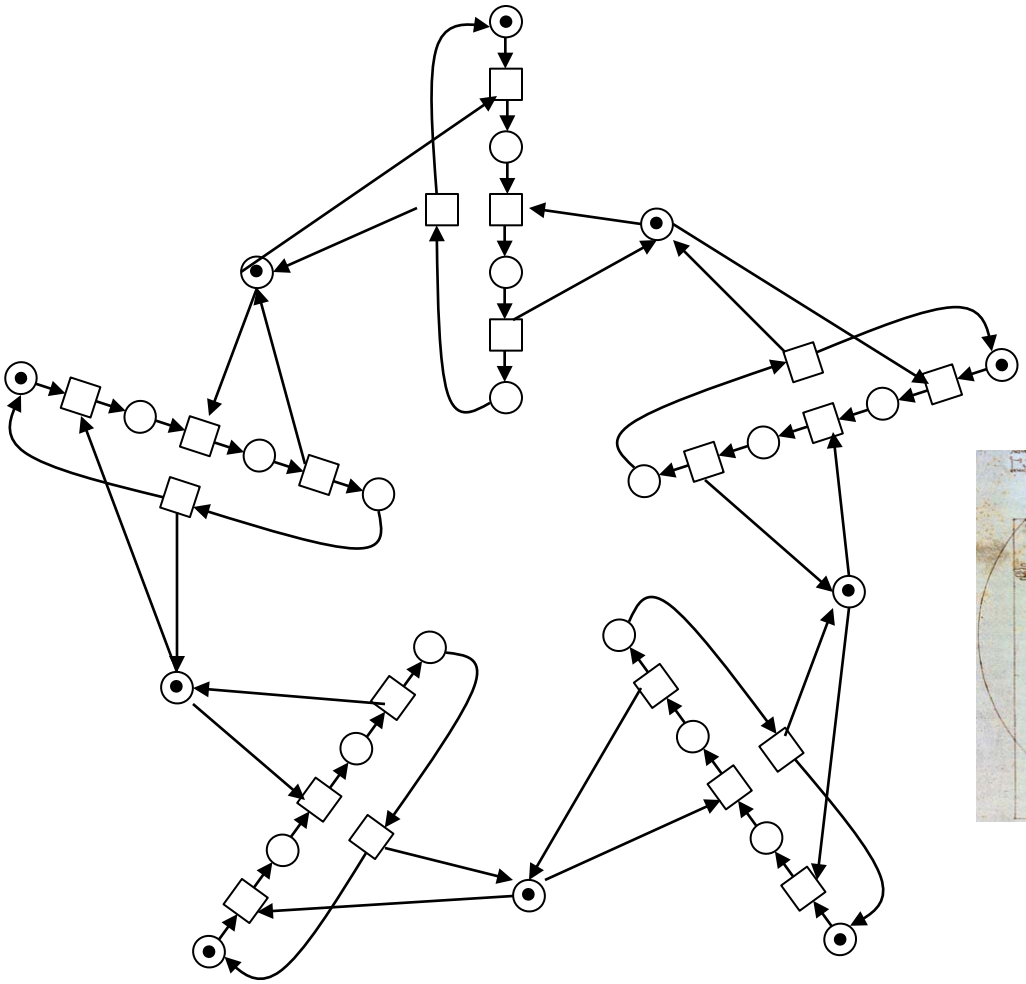
- Relies on progress measure

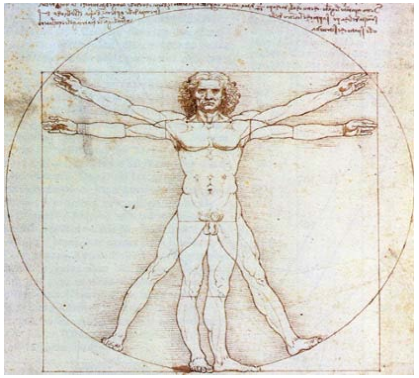
LoLA computes measure automatically:



3. The symmetry method

LoLA: Symmetry = graph automorphism of the PT-Net





Example

```
demo_munich — bash — 96×24
Karsten-Wolfs-MacBook-Pro:demo_munich karsten$ ./lola --check=full  --symmetry data17.llnet
lola: reading net from data17.llnet
lola: finished parsing
lola: closed net file data17.llnet
lola: 162/65536 symbol table entries, 0 collisions
lola: preprocessing net
lola: computing forward-conflicting sets
lola: computing back-conflicting sets
lola: 55 transition conflict sets
lola: finding significant places
lola: 90 places, 72 transitions, 36 significant places
lola: computing symmetries (--symmetry)
lola: computed 306 generators (35 in search tree, 271 by composition)
lola: representing 4.09904E+31 symmetries
lola: 0 dead branches visited in search tree
lola: using a bit-perfect encoder (--encoder)
lola: using 144 bytes per marking, with 0 unused bits
lola: using a prefix store (--store)
lola: killed reporter thread
lola: building the complete state space (--check=full)
lola: finished preprocessing
lola: result: no
lola: 20 markings, 361 edges
Karsten-Wolfs-MacBook-Pro:demo_munich karsten$
```

Example

```
demo_munich — bash — 96×24
Karsten-Wolfs-MacBook-Pro:demo_munich karsten$ ./lola --check=full  --symmetry data17.llnet
lola: reading net from data17.llnet
lola: finished parsing
lola: closed net file data17.llnet
lola: 162/65536 symbol table entries, 0 collisions
lola: preprocessing net
lola: computing forward-conflicting sets
lola: computing back-conflicting sets
lola: 55 transition conflict sets
lola: finding significant places
lola: 90 places, 72 transitions, 36 significant places
lola: computing symmetries (--symmetry)
lola: computed 306 generators (35 in search tree, 271 by composition)
lola: representing 4.09904E+31 symmetries
lola: 0 dead branches visited in search tree
lola: using a bit-perfect encoder (--encoder)
lola: using 144 bytes per marking, with 0 unused bits
lola: using a prefix store (--store)
lola: killed reporter thread
lola: building the complete state space (--check=full)
lola: finished preprocessing
lola: result: no
lola: 20 markings, 361 edges
Karsten-Wolfs-MacBook-Pro:demo_munich karsten$
```

Example

```
demo_munich — bash — 96×24
Karsten-Wolfs-MacBook-Pro:demo_munich karsten$ ./lola --check=full  --symmetry data17.llnet
lola: reading net from data17.llnet
lola: finished parsing
lola: closed net file data17.llnet
lola: 162/65536 symbol table entries, 0 collisions
lola: preprocessing net
lola: computing forward-conflicting sets
lola: computing back-conflicting sets
lola: 55 transition conflict sets
lola: finding significant places
lola: 90 places, 72 transitions, 36 significant places
lola: computing symmetries (--symmetry)
lola: computed 306 generators (35 in search tree, 271 by composition)
lola: representing 4.09904E+31 symmetries
lola: 0 dead branches visited in search tree
lola: using a bit-perfect encoder (--encoder)
lola: using 144 bytes per marking, with 0 unused bits
lola: using a prefix store (--store)
lola: killed reporter thread
lola: building the complete state space (--check=full)
lola: finished preprocessing
lola: result: no
lola: 20 markings, 361 edges
Karsten-Wolfs-MacBook-Pro:demo_munich karsten$
```

Example

```
demo_munich — bash — 96×24
Karsten-Wolfs-MacBook-Pro:demo_munich karsten$ ./lola --check=full  --symmetry data17.llnet
lola: reading net from data17.llnet
lola: finished parsing
lola: closed net file data17.llnet
lola: 162/65536 symbol table entries, 0 collisions
lola: preprocessing net
lola: computing forward-conflicting sets
lola: computing back-conflicting sets
lola: 55 transition conflict sets
lola: finding significant places
lola: 90 places, 72 transitions, 36 significant places
lola: computing symmetries (--symmetry)
lola: computed 306 generators (35 in search tree, 271 by composition)
lola: representing 4.09904E+31 symmetries
lola: 0 dead branches visited in search tree
lola: using a bit-perfect encoder (--encoder)
lola: using 144 bytes per marking, with 0 unused bits
lola: using a prefix store (--store)
lola: killed reporter thread
lola: building the complete state space (--check=full)
lola: finished preprocessing
lola: result: no
lola: 20 markings, 361 edges
Karsten-Wolfs-MacBook-Pro:demo_munich karsten$
```

Commercial I

Functional Verification of Task Partitioning for Multiprocessor Embedded Systems

DIPANKAR DAS, P. P. CHAKRABARTI, and RAJEEV KUMAR
Indian Institute of Technology Kharagpur

	nodes UML	places Petri Net	transitions Petri Net	time taken				
				Spin	LoLA	PROD	PEP	TraceMatch
crc32_T	70	90	30	580 ms	10 ms	2170 ms	30 ms	1 ms
crc32_F	70	90	30	600 ms	10 ms	2200 ms	30 ms	2 ms
adpcm_T	85	102	66	630 ms	10 ms	2340 ms	20 ms	2 ms
adpcm_F	85	102	66	590 ms	10 ms	2420 ms	20 ms	2 ms
fft_T	98	76	106	600 ms	<10 ms	2520 ms	20 ms	1 ms
fft_F	98	76	106	640 ms	20 ms	2660 ms	20 ms	3 ms
basicmath_T	265	320	157	1700 ms	20 ms	11300 ms	170 ms	9 ms
basicmath_F	265	320	157	1880 ms	40 ms	12420 ms	180 ms	27 ms
dijkstra_T	285	302	229	1340 ms	20 ms	8570 ms	140 ms	23 ms
dijkstra_F	285	302	229	1330 ms	80 ms	8750 ms	140 ms	27 ms
sha_T	309	314	239	2050 ms	20 ms	13640 ms	200 ms	8 ms
sha_F	309	314	239	2020 ms	50 ms	13550 ms	210 ms	15 ms

Commercial II

The Pathway Logic Assistant

Carolyn Talcott[1] and David L. Dill[2]

[1] SRI International
clt@csl.sri.com
[2] Stanford University
dill@cs.stanford.edu

Figure 1 shows the result of the Petri net query corresponding to `prop3` found by the LoLA [13] Petri net analysis tool, as displayed by PLA using the Pathalyzer tool. Lola uses "stubborn set reduction", which is a technique that exploits the ease of determining the independence of certain transitions in the Petri nets. For reachability queries on our nets, answering a reachability query that would have taken hours using a general purpose model-checking tool takes on the order of a second in LoLA—fast enough to permit interactive use.

Commercial III

Model Checking Contest @ Petri Nets 2013

Milano, Italy, June 25, 2013

Trophies for the ReachabilityMix Examination

Last Updated Apr. 26, 2013

Model Checking Contest @ Petri Nets 2014

Tunis, Tunisia, June 24, 2014

Winners for the Reachability Category

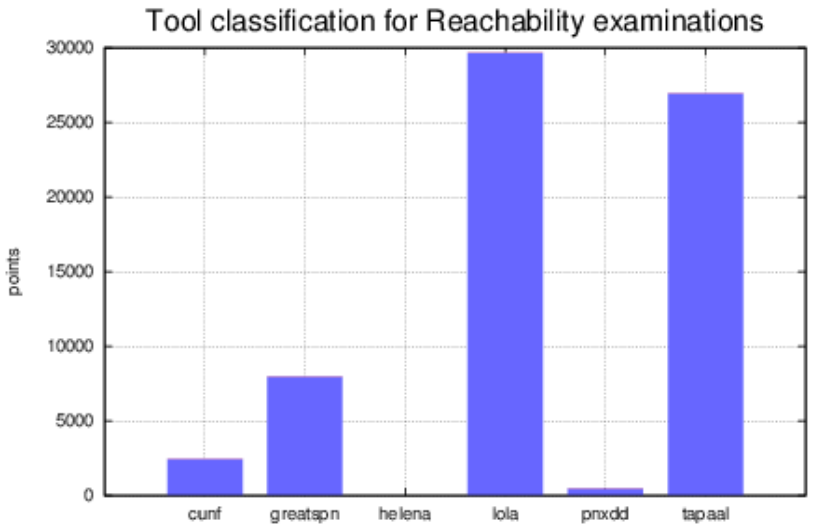
Last Updated Sept. 1, 2014

Global trophies for ReachabilityMix

For this trophy, we use the following formula: results on "Known" models + 2 x results on "Surprise" models.

Trophies for All Models

LoLA optimistic 213 (points)	LoLA 208 (points)	Marcie 166 (points)	LoLA optimistic incomplete 166 (points)



Applications

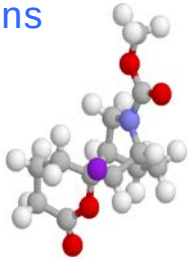
Hazards in asynchronous circuits

Information flow security in web services

AI planning for web service composition

Soundness in business processes

Biochemical reaction chains

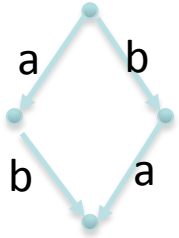


Multiprocessor embedded systems

Parameterized problems

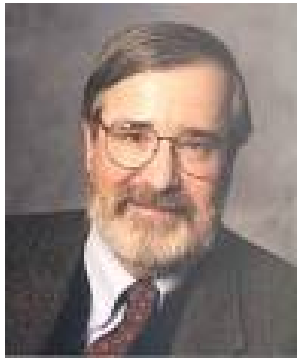
Conclusion

Why

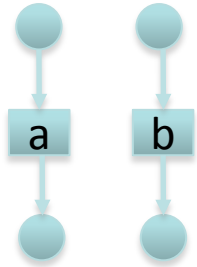


?

That‘s why



Why



?

Further reading:

- Tools: www.service-technology.org
- Group / Papers: www.informatik.uni-rostock.de/tpp/