Randomized Reachability Analysis in Uppaal: Fast Error Detection in Timed Systems

Andrej Kiviriga
kiviriga@cs.aau.dk

Kim G. Larsen
kgl@cs.aau.dk

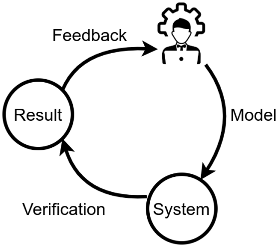
Ulrik Nyman
ulrik@cs.aau.dk

AALBORG UNIVERSITY
DENMARK

August 12, 2021

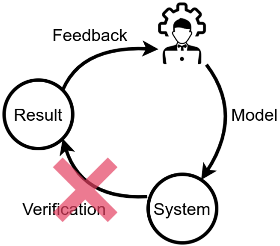
Motivation

- Formal methods and model correctness



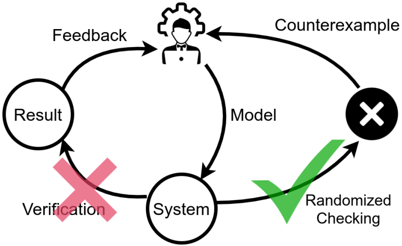
Motivation

- Formal methods and model correctness
- Hampered development due to the state-space explosion problem



Motivation

- Formal methods and model correctness
- Hampered development due to the state-space explosion problem
- Need for an efficient and scalable method for error detection



Main Contributions

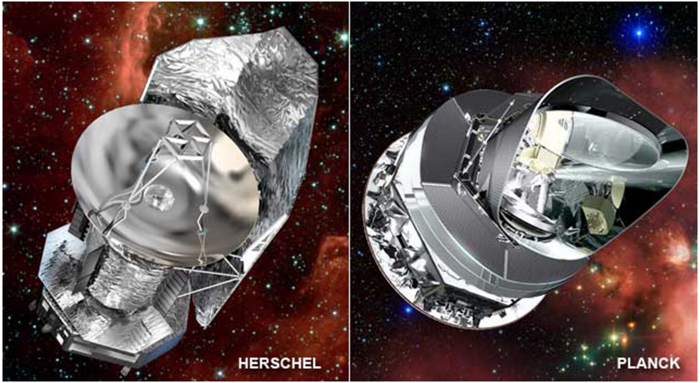
- Randomized Reachability Analysis in UPPAAL
- Detection of "rare events" up to several orders of magnitude faster (23 hours $\rightarrow$ 23 seconds)
- Possibility to analyze previously intractable models
- Searching for *shorter* or *faster* trace

Herschel-Planck Case Study

- System consisting of two satellites: Herschel and Planck
- The architecture consists of a single processor
- 32 individual tasks being executed with the policy of fixed priority preemptive scheduling
- A mixture of priority ceiling and priority inheritance protocols is used for resource sharing and deadlines extended beyond period.
- The control software developed by the Danish company Terma A/S
- Worst-case response time (WCRT) analysis has shown that one task my miss its deadline; though this has never been observed

Herschel-Planck Case Study

- Two satellites:
  Herschel and Planck
- Single processor architecture
- 32 individual tasks
- Preemptive scheduling
- The control software by Terma A/S
- Worst-case response time
  analysis performed by Terma A/S



Herschel-Planck Case Study

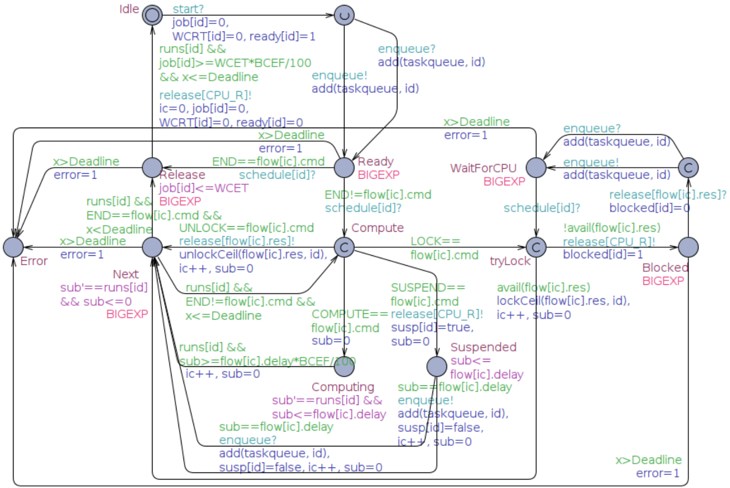
Model-based approach

- Model-based approach considers more parameters and provides more exact analysis.
- The preemption is encoded with stopwatches.
- First application of model-based approach on Herschel-Planck in 2010, but with the unrealistic assumption of fixed execution times (ET) for tasks.
- In 2012 an improved analysis was carried out with each task given a non-deterministic ET.
- Interval-based ET, preemption, shared resources, etc. unfortunately makes schedulability of HP undecidable.
- The symbolic model-checking (MC) of UPPAAL is over-approximate for stopwatch automata.

Herschel-Planck Case Study

Model-based approach

- Model-based approach
- Stopwatches to encode preemption
- Herschel-Planck Case Study [3] in 2010
- Herschel-Planck Case Study Revisited [1] in 2012



Herschel-Planck Case Study

Schedulability summary

Schedulability of Herschel-Planck Revisited Using Statistical Model Checking [1]

$f = \frac{\text{BCET}}{\text{WCET}}$	0-71%	72-80%	81-86%	87-90%	90-100%
Symbolic MC:	maybe	maybe	maybe	n/a	**Safe**
Statistical MC:	**Unsafe**	maybe	maybe	maybe	maybe

Herschel-Planck Case Study

Schedulability summary

Schedulability of Herschel-Planck Revisited Using Statistical Model Checking [1]

$f = \frac{\text{BCET}}{\text{WCET}}$	0-71%	72-80%	81-86%	87-90%	90-100%
Symbolic MC:	maybe	maybe	maybe	n/a	**Safe**
Statistical MC:	**Unsafe**	maybe	maybe	maybe	maybe

Herschel-Planck Case Study

Schedulability summary

Schedulability of Herschel-Planck Revisited Using Statistical Model Checking [1]

$f = \frac{\text{BCET}}{\text{WCET}}$	0-71%	72-80%	81-86%	87-90%	90-100%
Symbolic MC:	maybe	maybe	maybe	n/a	**Safe**
Statistical MC:	**Unsafe**	maybe	maybe	maybe	maybe

Herschel-Planck Case Study

Schedulability summary

$f = \frac{\text{BCET}}{\text{WCET}}$	0-71%	72-80%	81-86%	87-90%	90-100%
Symbolic MC:	maybe	maybe	maybe	n/a	**Safe**
Statistical MC:	**Unsafe**	maybe	maybe	maybe	maybe
Randomized MC:	**Unsafe**	**Unsafe**	maybe	maybe	maybe

Randomized Reachability Analysis

- Exploration based on random walks
- Operating on concrete semantics
- No states stored (memory efficient), but no termination guarantee
- Several randomized heuristics (SEM, RET, RLC, RLC-A)
- Adaptive delay choice mechanism in RET, RLC and RLC-A
- Dynamic depth up to a specified constant
- Search for “shorter” or “faster” trace

Randomized Heuristics

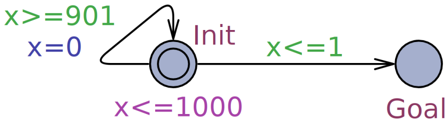
Semantic Exploration (SEM)

Choice of a meaningful delay (that leads to an enabled transition) uniformly at random followed by a uniform choice of available transitions after the selected delay was made.

Randomized Heuristics

Semantic Exploration (SEM)

Choice of a meaningful delay (that leads to an enabled transition) uniformly at random followed by a uniform choice of available transitions after the selected delay was made.

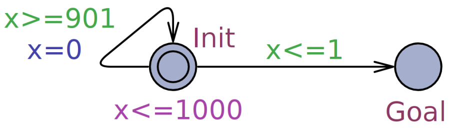


Randomized Heuristics

Semantic Exploration (SEM)

Choice of a meaningful delay (that leads to an enabled transition) uniformly at random followed by a uniform choice of available transitions after the selected delay was made.

$$P_{SEM_1}(\texttt{E<>Goal}) = \frac{1}{100}$$



Randomized Heuristics

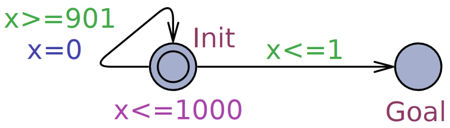
Semantic Exploration (SEM)

Choice of a meaningful delay (that leads to an enabled transition) uniformly at random followed by a uniform choice of available transitions after the selected delay was made.

Random Enabled Transition (RET)

Compute all eventually enabled transitions, i.e. transitions which are either currently available or will become such after a delay. Choose a single transition as a target uniformly at random. [2]

$$P_{SEM_1}(\texttt{E<>Goal}) = \frac{1}{100}$$



Randomized Heuristics

Semantic Exploration (SEM)

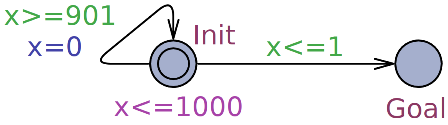
Choice of a meaningful delay (that leads to an enabled transition) uniformly at random followed by a uniform choice of available transitions after the selected delay was made.

Random Enabled Transition (RET)

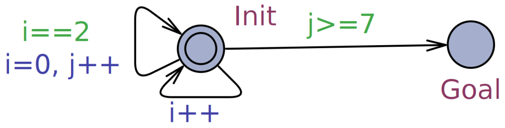
Compute all eventually enabled transitions, i.e. transitions which are either currently available or will become such after a delay. Choose a single transition as a target uniformly at random. [2]

$$P_{SEM_1}(\texttt{E<>Goal}) = \frac{1}{100}$$

$$P_{RET_1}(\texttt{E<>Goal}) = \frac{1}{2}$$

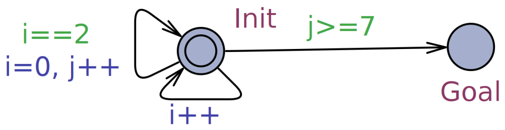


Randomized Heuristics



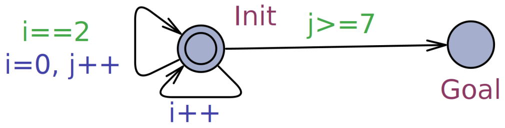
Randomized Heuristics

$P_{RET}(\texttt{E<>Goal}) < 1\%$



Randomized Heuristics

$P_{RET}(\texttt{E<>Goal}) < 1\%$



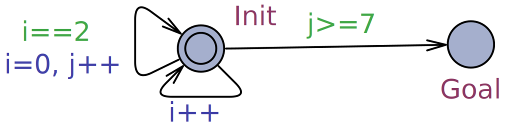
Random Least Coverage (RLC)

Uniformly at random choose of a transition with the least coverage for the sending edge. The coverage counters are reset after each random walk.

Randomized Heuristics

$P_{RET}(\texttt{E<>Goal}) < 1\%$

$P_{RLC}(\texttt{E<>Goal}) = 100\%$



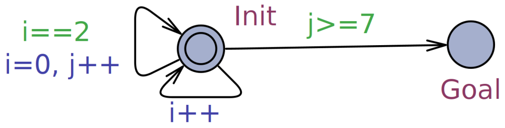
Random Least Coverage (RLC)

Uniformly at random choose of a transition with the least coverage for the sending edge. The coverage counters are reset after each random walk.

Randomized Heuristics

$P_{RET}(\texttt{E<>Goal}) < 1\%$

$P_{RLC}(\texttt{E<>Goal}) = 100\%$



Random Least Coverage (RLC)

Uniformly at random choose of a transition with the least coverage for the sending edge. The coverage counters are reset after each random walk.

Random Least Coverage - Accumulative (RLC-A)

Uniformly at random choose a transition with the least coverage for the sending edge. Keep the coverage counters shared between the random walks.

Adaptive delay choice and dynamic depth

How to choose delay values in RET, RLC and RLC-A?
“Randomized Refinement Checking” [2] hints at adaptive delays being more efficient in practice

Adaptive delay choice and dynamic depth

How to choose delay values in RET, RLC and RLC-A?
“Randomized Refinement Checking” [2] hints at adaptive delays being more efficient in practice

Sequence	1	2	3	4	5	6	7	8	9	10	11
Lower bound	60%	70%	80%	90%	100%	0%	10%	20%	30%	40%	40%
Uniform	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	20%
Upper bound	40%	30%	20%	10%	0%	100%	90%	80%	70%	60%	40%

Delay probability distributions used for RET, RLC and RLC-A.

Adaptive delay choice and dynamic depth

How to choose delay values in RET, RLC and RLC-A?
"Randomized Refinement Checking" [2] hints at adaptive delays being more efficient in practice

Sequence	1	2	3	4	5	6	7	8	9	10	11
Lower bound	60%	70%	80%	90%	100%	0%	10%	20%	30%	40%	40%
Uniform	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	20%
Upper bound	40%	30%	20%	10%	0%	100%	90%	80%	70%	60%	40%

Delay probability distributions used for RET, RLC and RLC-A.

Dynamic depth of random walks:

- Start at 2^4 steps
- Double the amount of steps after a full sequence of random walks is completed
- Repeat until the maximum depth specified by the user

Herschel-Planck Case Study

$f(\%)$	SMC(160)	SMC(640)	SMC(1280)	SEM	RET	RLC	RLC-A
68	3378.82	3656.0	2626.11	nf	**14.1**	14.35	14.48
69	6087.64	3258.13	3565.49	nf	15.91	14.32	**13.7**
70	19408.04	16875.89	24322.69	nf	17.59	14.47	**14.77**
71	85837.23	nf	nf	nf	22.54	**16.56**	16.75
72	nf	nf	nf	nf	27.81	**18.42**	18.96
73	nf	nf	nf	nf	31.56	**20.66**	20.68
74	nf	nf	nf	nf	52.53	**38.08**	40.31
75	nf	nf	nf	nf	72.16	**61.98**	68.35
76	nf	nf	nf	nf	**83.12**	328.03	327.32
77	nf	nf	nf	nf	**375.08**	nf	nf
78	nf	nf	nf	nf	**1155.50**	nf	nf
79	nf	nf	nf	nf	**2009.01**	nf	nf
80	nf	nf	nf	nf	**11194.43**	nf	nf
81	nf	nf	nf	nf	nf	nf	nf

Average time to detect non-schedulability in Herschel-Planck (in seconds). SMC search is limited to 160, 640 or 1280 cycles of 250ms.

Herschel-Planck Case Study

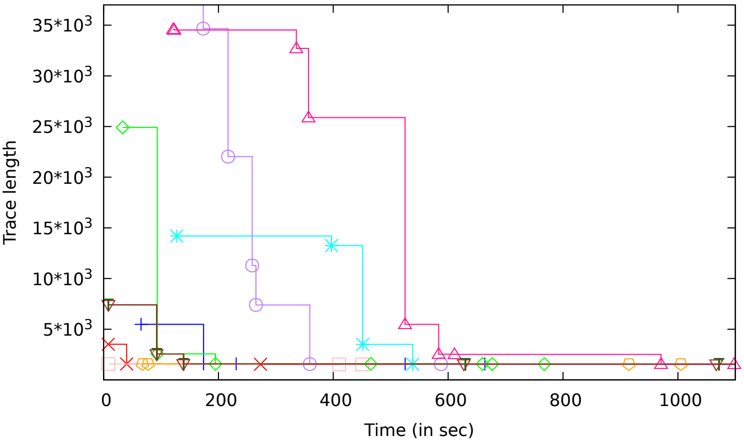
f(%)	RET	RET-S	Timeout
68	6882	560	1h
69	7619	568	1h
70	8285	572	1h
71	10411	570	1h
72	12394	571	1h
73	15937	578	1h
74	26605	1549	1h
75	41003	1546	1h
76	40154	1529	1h
77	97258	1536	1h
78	119939	1540	5h
79	129387	1536	5h
80	145493	6455	20h

Trace length comparison.

Herschel-Planck Case Study

f(%)	RET	RET-S	Timeout
68	6882	560	1h
69	7619	568	1h
70	8285	572	1h
71	10411	570	1h
72	12394	571	1h
73	15937	578	1h
74	26605	1549	1h
75	41003	1546	1h
76	40154	1529	1h
77	97258	1536	1h
78	119939	1540	5h
79	129387	1536	5h
80	145493	6455	20h

Trace length comparison.



10 runs of RET-S for Herschel-Planck with $f = 75\%$.

More schedulability

Model	#loc	BFS	DFS	RDFS	SMC	SEM	RET	RLC	RLC-A
IMAOptim-0	88	0.09	0.1	0.07	**0.04**	0.07	0.1	0.1	0.08
IMAOptim-1	88	0.21	0.2	0.08	**0.05**	**0.05**	0.08	0.08	0.06
IMAOptim-2	88	0.21	0.26	0.09	**0.06**	0.08	0.11	0.11	0.1
md5-jop	594	0.25	10.8	6.53	n/a	0.15	0.18	0.18	**0.12**
md5-hvmimp	476	0.41	0.85	0.49	n/a	0.1	0.14	0.14	**0.09**
md5-hvmexp	11901	oom	oom	oom	n/a	14.17	19.85	20.18	**8.71**
MP-jop	371	0.39	0.14	0.12	n/a	**0.08**	0.12	0.12	0.09
MP-hvmimp	371	0.35	0.14	0.12	n/a	**0.08**	0.12	0.12	0.09
MP-hvmexp	4388	oom	oom	oom	n/a	13.49	22.95	21.99	**8.59**
simplerts-opt	409	oom	oom	oom	n/a	2.43	**1.48**	nf	nf

Average time to find target state in stopwatch automata models.

Symbolic MC techniques provide potentially spurious traces.

More schedulability

Model	#loc	BFS	DFS	RDFS	SMC	SEM	RET	RLC	RLC-A
IMAOptim-0	88	0.09	0.1	0.07	**0.04**	0.07	0.1	0.1	0.08
IMAOptim-1	88	0.21	0.2	0.08	**0.05**	**0.05**	0.08	0.08	0.06
IMAOptim-2	88	0.21	0.26	0.09	**0.06**	0.08	0.11	0.11	0.1
md5-jop	594	0.25	10.8	6.53	n/a	0.15	0.18	0.18	**0.12**
md5-hvmimp	476	0.41	0.85	0.49	n/a	0.1	0.14	0.14	**0.09**
md5-hvmexp	11901	oom	oom	oom	n/a	14.17	19.85	20.18	**8.71**
MP-jop	371	0.39	0.14	0.12	n/a	**0.08**	0.12	0.12	0.09
MP-hvmimp	371	0.35	0.14	0.12	n/a	**0.08**	0.12	0.12	0.09
MP-hvmexp	4388	oom	oom	oom	n/a	13.49	22.95	21.99	**8.59**
simplerts-opt	409	oom	oom	oom	n/a	2.43	**1.48**	nf	nf

Average time to find target state in stopwatch automata models.

Symbolic MC techniques provide potentially spurious traces.

Gossiping Girls

- Each girl knows a distinct secret
- The secrets can be shared with other girls through calls
- Organized as total graph
- A string with at most 2^{n^2} values for n girls
- We use models developed by Master's Thesis students at Aalborg University

Gossiping Girls

All secrets known

Model	BFS	DFS	RDFS	SEM	RET	RLC	RLC-A
Gosgirls-1	oom	oom	697.13	nf	**0.39**	6949.95	nf
Gosgirls-2	oom	oom	**0.02**	nf	0.04	0.04	0.04
Gosgirls-3	oom	oom	44.49	nf	**0.02**	**0.02**	0.09
Gosgirls-4	oom	oom	28.35	nf	**0.03**	0.03	nf
Gosgirls-5	oom	oom	229.98	nf	**0.02**	**0.02**	**0.02**
Gosgirls-6	oom	oom	64.00	nf	**3.71**	167.44	1530.99
Gosgirls-7	oom	oom	55.61	nf	**0.17**	15.16	15.6
Gosgirls-8	oom	oom	13.96	nf	0.04	**0.03**	**0.03**
Gosgirls-9	oom	oom	2.08	nf	0.08	**0.07**	0.08
Gosgirls-10	oom	oom	598.64	nf	**0.24**	1.72	nf

Gossiping Girls with 8 nodes. Each cell represent avg. time for each found trace within 2 hours.
Searching for a state with all secrets known within a certain time..

Gossiping Girls

Particular cluster configuration

Model	BFS	DFS	RDFS	SEM	RET	RLC	RLC-A
Gosgirls-1	16.98	oom	oom	2.17	**1.35**	1.60	0.23
Gosgirls-2	**0.04**	oom	360.43	**0.04**	**0.04**	**0.04**	**0.04**
Gosgirls-3	77.96	oom	oom	nf	1.44	0.19	**0.10**
Gosgirls-4	oom	oom	oom	nf	0.03	**0.02**	nf
Gosgirls-5	oom	oom	oom	nf	**0.02**	**0.02**	**0.02**
Gosgirls-6	oom	244.66	2596.62	**5.92**	7.10	nf	nf
Gosgirls-7	oom	oom	oom	nf	**0.14**	75.44	141.20
Gosgirls-8	32.63	oom	oom	nf	**0.11**	3.24	505.99
Gosgirls-9	oom	oom	199.77	**0.10**	13.04	3.65	2.07
Gosgirls-10	oom	oom	209.36	nf	**0.02**	0.03	0.04

Gossiping Girls with 6 nodes. Each cell represent avg. time for each found trace within 2 hours.
Searching for a particular configuration of secrets known.

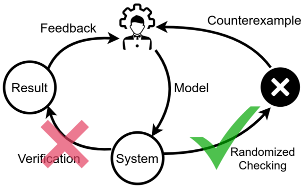
Scalability Experiments

Model	BFS	DFS	RDFS	SEM	RET	RLC	RLC-A
csma-cd-20N	20.2	oom	**0.02**	0.03	0.07	0.06	0.21
csma-cd-22N	37.48	oom	oom	**0.03**	0.08	0.08	0.31
csma-cd-25N	91.0	oom	oom	**0.05**	0.09	0.1	0.55
csma-cd-30N	313.54	oom	oom	**0.05**	0.12	0.19	1.43
csma-cd-50N	oom	oom	oom	**0.46**	0.84	1.19	15.29
Fischer-10N	0.9	22.84	4.3	**0.04**	0.05	1.21	nf
Fischer-15N	8.35	6037.63	9038.96	**0.09**	**0.09**	5.06	nf
Fischer-20N	72.61	oom	oom	0.3	**0.28**	17.28	nf
Fischer-25N	452.45	oom	oom	**0.64**	0.73	36.93	nf
Fischer-50N	oom	oom	90.01	**21.78**	23.79	233.67	nf
FischerME-10N	7.15	0.14	0.02	0.01	0.02	**0.01**	0.02
FischerME-15N	oom	11.45	0.05	0.04	0.04	**0.03**	0.16
FischerME-20N	oom	970.33	0.4	0.11	0.09	0.05	**0.04**
FischerME-25N	oom	oom	83.29	0.25	0.21	0.08	**0.07**
FischerME-50N	oom	oom	174.32	14.87	15.26	**0.49**	4.04
LE-Chan-3N	0.03	0.35	0.04	**0.01**	**0.01**	**0.01**	**0.01**
LE-Chan-4N	oom	oom	107.7	0.95	0.54	4.36	**0.07**
LE-Chan-5N	oom	oom	1167.41	53.21	**31.38**	102.08	nf
LE-Hops-3N	0.02	0.02	0.02	**0.01**	**0.01**	**0.01**	**0.01**
LE-Hops-4N	oom	oom	oom	49.40	**14.57**	428.96	1588.33
LE-Hops-5N	oom	oom	1108.15	63.44	**35.15**	36.49	49.00
Milner-N100	0.45	0.16	2.72	nf	**0.11**	**0.11**	0.12
Milner-N500	44.44	10.56	1619.75	nf	**1.19**	1.2	1.43
Milner-N1000	488.41	110.35	36455.73	nf	**4.44**	4.45	4.59
Train-200N	oom	5.64	6.06	5.91	**5.4**	16699.98	nf
Train-300N	oom	28.19	30.28	**25.62**	26.53	nf	nf
Train-400N	oom	85.22	90.66	**67.91**	70.87	nf	nf
Train-500N	oom	210.89	223.13	**181.99**	188.9	nf	nf
Train-1000N	nf	3461.17	3542.08	**2192.12**	2541.57	nf	nf
Train-2000N	nf	71286.92	oom	**19229.02**	23233.21	nf	nf

Future Work

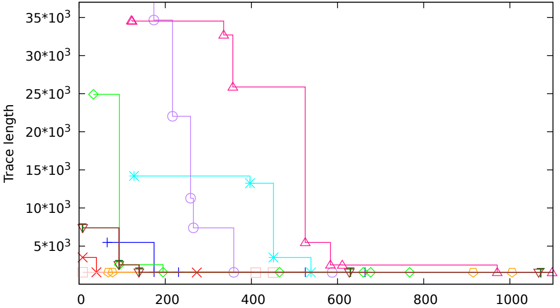
- Further investigations into tokenized, coverage-based and guided methods
- Static analysis and look-ahead in random walks
- Automatic sanity check for quick feedback in UPPAAL

Summary



$f(\%)$	SMC(160)	SMC(640)	SMC(1280)	SEM	RET	RLC	RLC-A
68	3378.82	3656.0	2626.11	nf	**14.1**	14.35	14.48
69	6087.64	3258.13	3565.49	nf	15.91	14.32	**13.7**
70	19408.04	16875.89	24322.69	nf	17.59	14.47	**14.77**
71	85837.23	nf	nf	nf	22.54	**16.56**	16.75
72	nf	nf	nf	nf	27.81	**18.42**	18.96
73	nf	nf	nf	nf	31.56	**20.66**	20.68
74	nf	nf	nf	nf	52.53	**38.08**	40.31
75	nf	nf	nf	nf	72.16	**61.98**	68.35
76	nf	nf	nf	nf	**83.12**	328.03	327.32
77	nf	nf	nf	nf	**375.08**	nf	nf
78	nf	nf	nf	nf	**1155.50**	nf	nf
79	nf	nf	nf	nf	**2009.01**	nf	nf
80	nf	nf	nf	nf	**11194.43**	nf	nf
81	nf	nf	nf	nf	nf	nf	nf

f(%)	RET	RET-S	Timeout
68	6882	560	1h
69	7619	568	1h
70	8285	572	1h
71	10411	570	1h
72	12394	571	1h
73	15937	578	1h
74	26605	1549	1h
75	41003	1546	1h
76	40154	1529	1h
77	97258	1536	1h
78	119939	1540	5h
79	129387	1536	5h
80	145493	6455	20h



References I

[1] A. David, K. G. Larsen, A. Legay, and M. Mikucionis.
Schedulability of herschel-planck revisited using statistical model checking.
In T. Margaria and B. Steffen, editors, *Leveraging Applications of Formal Methods, Verification and Validation. Applications and Case Studies - 5th International Symposium, ISoLA 2012, Heraklion, Crete, Greece, October 15-18, 2012, Proceedings, Part II*, volume 7610 of *Lecture Notes in Computer Science*, pages 293–307. Springer, 2012.

[2] A. Kiviriga, K. G. Larsen, and U. Nyman.
Randomized refinement checking of timed I/O automata.
In J. Pang and L. Zhang, editors, *Dependable Software Engineering. Theories, Tools, and Applications - 6th International Symposium, SETTA 2020, Guangzhou, China, November 24-27, 2020, Proceedings*, volume 12153 of *Lecture Notes in Computer Science*, pages 70–88. Springer, 2020.

[3] M. Mikučionis, K. G. Larsen, J. I. Rasmussen, B. Nielsen, A. Skou, S. U. Palm, J. S. Pedersen, and P. Hougaard.
Schedulability analysis using uppaal: Herschel-planck case study.
In T. Margaria and B. Steffen, editors, *Leveraging Applications of Formal Methods, Verification, and Validation*, pages 175–190, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.