



Report on the 2021 edition

Philipp Rümmer & Grigory Fedyukovich

Outline

- Tracks and Benchmarks
- Teams and Solvers
- **Results**

Tracks

- Linear Integer Arithmetic (LIA), linear clauses (**LIA-Lin**)
- LIA, nonlinear clauses (**LIA-Nonlin**)
- LIA and Arrays, linear clauses (**LIA-Lin-Array**)
- LIA and Arrays, nonlinear clauses (**LIA-Nonlin-Array**)
- Linear Real Arithmetic (LRA), transition systems (**LRA-TS**)
- LRA, transition systems, parallel (**LRA-TS-Parallel**)
- Algebraic Data Types (ADT), nonlinear (**ADT-Nonlin**)

Tracks

- Linear Integer Arithmetic (LIA), linear clauses (**LIA-Lin**)
- LIA, nonlinear clauses (**LIA-Nonlin**)
- LIA and Arrays, linear clauses (**LIA-Lin-Array**)
- LIA and Arrays, nonlinear clauses (**LIA-Nonlin-Array**)
- Linear Real Arithmetic (LRA), transition systems (**LRA-TS**)
- LRA, transition systems, parallel (**LRA-TS-Parallel**)
- Algebraic Data Types (ADT), nonlinear (**ADT-Nonlin**)

New in 2021

The Benchmarks

New Benchmarks in 2021

- **Spacer**
 - ADT+LIA benchmarks from Rust verification problems (post-processed afterwards)
 - LIA-Nonlin-Arrays benchmarks from SeaHorn
- **RInGen**
 - (Pure) ADT benchmarks from well-known TIP/Isaplanner suites
- **FreqHorn**
 - LIA-Lin benchmarks from the paper
- **ADTCHC**
 - Benchmarks from the paper
- **NayHorn**
 - LIA-NonLin benchmarks encoding syntax-guided synthesis problems
- **SemGuS**
 - LIA-NonLin-Arrays benchmarks encoding synthesis problems

Benchmark Processing

- Benchmarks were selected among those available on <https://github.com/chc-comp>
- Processing for ADT-NonLin:
 - Purification e.g., to replace INT by NAT and SMT functions by recursive functions
 - Convert `define-fun-rec` to inductive CHCs
 - (thanks to Yurii Kostyukov)
- All other tracks (same processing as in 2020):
 - Run `format.py` script from <https://github.com/chc-comp/chc-tools> to format according to the CHC-COMP rules
 - Run `format-checker` from <https://github.com/chc-comp/chc-tools> to verify correct format, and categorize into the different tracks
 - Elimination of duplicate benchmarks

Benchmark Library (total / unique number of benchmarks)

Repository	LIA-nonlin	LIA-lin	LIA-nonlin-arrays	LIA-lin-arrays	LRA-TS	ADT-nonlin
adt-purified						67 / 67
aeval		54 / 54				
eldarica-misc	69 / 66	147 / 134				
extra-small-lia		55 / 55				
hcai	135 / 133	100 / 86	25 / 25	39 / 39		
hopv	68 / 67	49 / 48				
jayhorn	5138 / 5084	75 / 73				
kind2	851 / 738					
ldv-ant-med			79 / 79	10 / 10		
ldv-arrays			821 / 546	3 / 2		
llreve	59 / 57	44 / 44		31 / 31		
quic3				43 / 43		
ringen						439 / 439
sally					177 / 174	
seahorn	68 / 66	3396 / 2822				
synth/nay-horn	119 / 114					
synth/semgus			2357* / 2331*			
tricera	4 / 4	405 / 405				
vmt		905 / 802			99 / 98	
chc-comp19	271 / 265	325 / 313	15 / 15	290 / 290	228 / 226	
sv-comp	1643 / 1169	3150 / 2932	855 / 779	79 / 73		
Total	8425 / 7763	8705 / 7768	4152* / 3775*	495 / 488	504 / 498	506 / 506

Benchmark Selection (LIA-NonLin, LIA-Lin, LIA-NonLin-Arrays)

- Ran the two top solvers from 2020 on each of the benchmarks:
 - LIA-NonLin: Spacer (5s t/o), Eldarica-abs (10s t/o)
 - LIA-Lin: Spacer (5s t/o), Eldarica-abs (10s t/o)
 - LIA-NonLin-Arrays: Spacer (5s t/o), Ultimate Unihorn (10s t/o)
- This yields three classes of benchmarks:
 - Rated A: both top solvers from 2020 can solve it
 - Rated B: one top solver from 2020 can solve it
 - Rated C: both top solvers from 2020 time out
- For each benchmark family, up to $3*N$ benchmarks were randomly chosen for the competition, N for each rating A, B, C
- (Same approach was applied in 2020)

Hardness Statistics

Repository	LIA-nonlin			LIA-lin			LIA-nl-arrays		
	#A /	#B /	#C	#A /	#B /	#C	#A /	#B /	#C
aeval				11 /	15 /	28			
eldarica-misc	35 /	4 /	27	105 /	20 /	9			
extra-small-lia				21 /	24 /	10			
hcai	74 /	44 /	15	73 /	8 /	5	14 /	6 /	5
hopv	60 /	7 /		47 /	1 /				
jayhorn	2688 /	769 /	1627	73 /	/				
kind2	250 /	455 /	33						
ldv-ant-med							/	25 /	54
ldv-arrays							/	127 /	419
llreve	35 /	13 /	9	37 /	5 /	2			
seahorn	38 /	19 /	9	977 /	985 /	860			
synth/nay-horn	46 /	30 /	38						
synth/semgus							282 /	768 /	1281
tricera	4 /	/		28 /	14 /	363			
vmt				85 /	616 /	101			
chc-comp19	144 /	80 /	41	80 /	101 /	132	/	7 /	8
sv-comp	1013 /	144 /	12	2801 /	17 /	114	258 /	268 /	253
Total	4387 /	1565 /	1811	4338 /	1806 /	1624	554 /	1201 /	2020

Benchmark Selection

Repository	LIA-nonlin		LIA-lin		LIA-nl-arrays		LIA-lin-arrays	LRA-TS	ADT-nonlin
	N_r	#Sel	N_r	#Sel	N_r	#Sel	#Selected	#Selected	#Selected
adt-purified									67
aeval			10	30					
eldarica-misc	10	30	15	39					
extra-small-lia			10	30					
hcai	20	55	15	28	5	15	39		
hopv	10	17	10	11					
jayhorn	30	90	10	10					
kind2	30	90							
ldv-ant-med					20	60	10		
ldv-arrays					30	90	2		
llreve	15	37	10	17			31		
quic3							43		
ringen									111
sally								174	
seahorn	15	39	30	90					
synth/nay-horn	20	60							
synth/semgus	20	60			45	135			
tricera	1	1	20	60					
vmt			30	90				98	
chc-comp19	30	90	30	90	5	15	290	226	
sv-comp	30	72	30	90	45	135	73		
Total		581		585		450	488	498	178

The Solvers

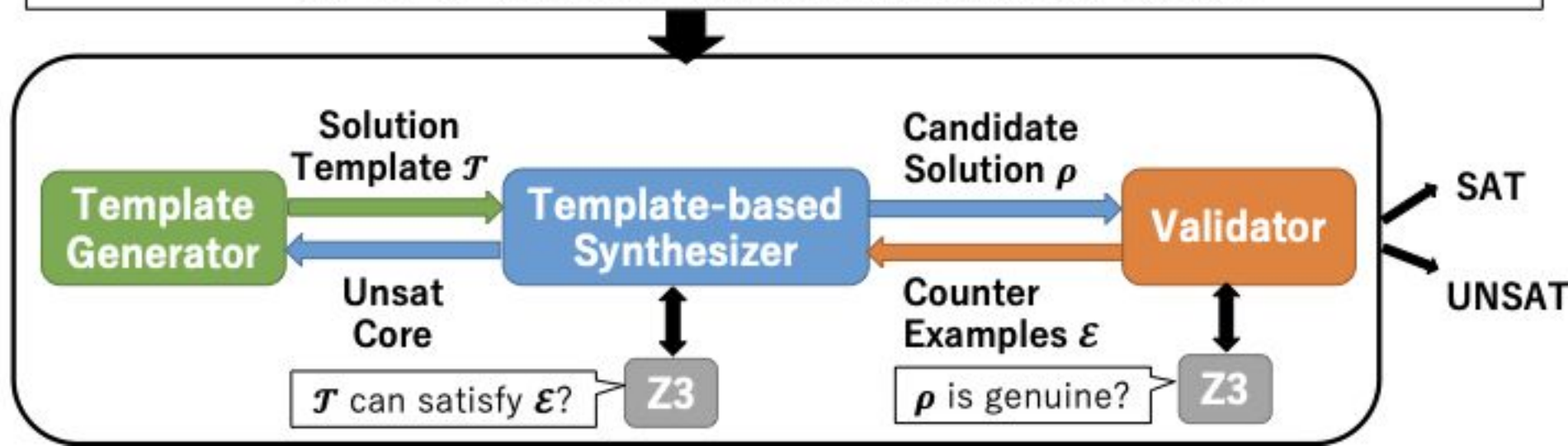
GOLEM CHC solver

- Recently started project at [USI Formal Verification and Security Lab](#)
 - as part of SNSF project *Beyond Symbolic Model Checking through Deep Modelling*
- Builds on top of the interpolating SMT solver [OpenSMT](#)
- Supports linear real and integer arithmetic
- Supports linear clauses
 - (support for non-linear clauses coming soon!)
- Implements Lazy Abstraction with Interpolants (also known as IMPACT) [[McMillan '06](#)]
- Current development:
<https://github.com/blishko/opensmt/tree/golem>

PCSat: A Predicate Constraint Solver

- **Algorithm:** CounterExample-Guided Inductive Synthesis + Template-based Synthesis + Unsat-core-based Refinement using Stratified Family of Solution Templates [ESOP'15]
- **Supported Theories:** Int, Real, ADT, Arrays

(1) CHC (2) extended CHC with head-disjunction/ \exists quantification and functional/well-founded predicates, (3) fixpoint logic constraints, (4) LTS safety/termination verification, (5) OCaml safety verification, (6) C LTL and CTL model checking (7) SyGuS



RINGEN: *Regular Invariant Generator*

- RINGEN supports (nonlinear) CHCs over ADTs
- Invariants are represented as *automata over ADT terms*
- Uses CVC4 as a backend finite-model finder
- Source (in F#): <https://github.com/Columpio/RInGen>
- Developed by Yurii Kostyukov (JetBrains Research, SPbU)

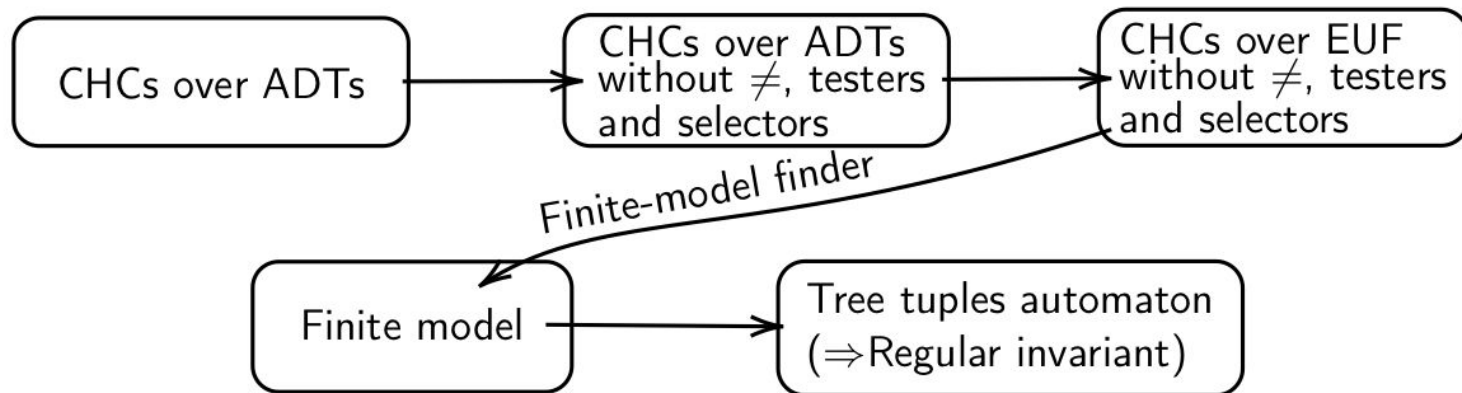
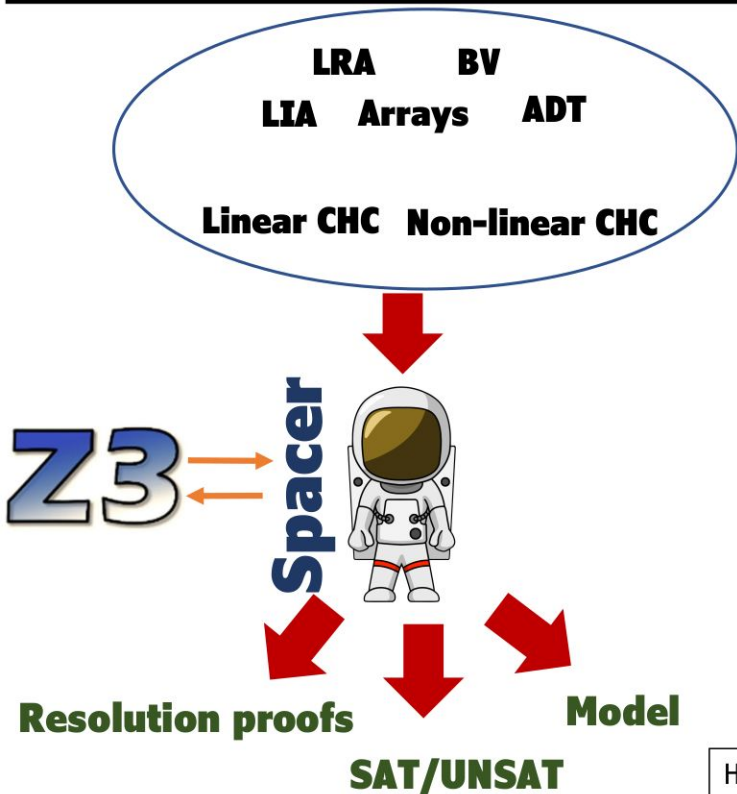
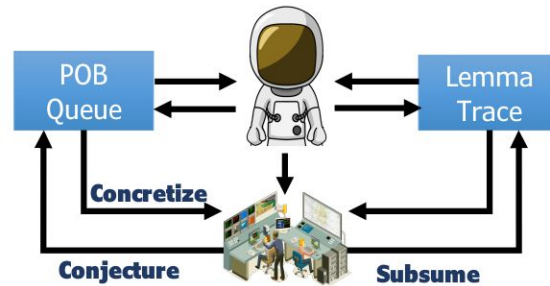


Figure: Obtaining regular model of a CHC system over ADTs

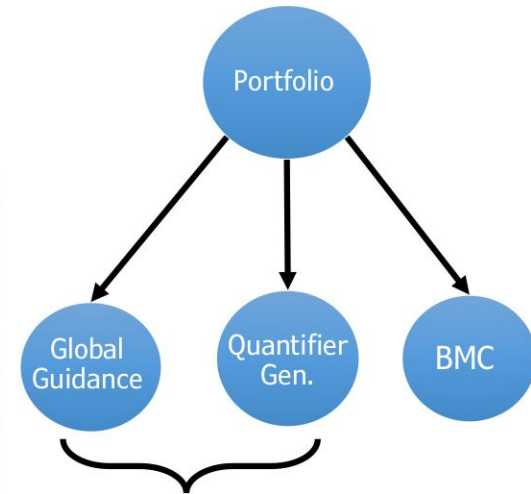


The main algorithm



IC3/PDR and many lemma generalization techniques

Competition submission



Both have inductive generalization and interpolation



- ▶ Horn solver developed since 2011
(authors: Hossein Hojjat and Philipp Rümmer)
- ▶ Open-source, implemented in Scala, running in JVM.
- ▶ **Input formats:**
SMT-LIB, Prolog, C, timed automata
- ▶ **Theories:**
LIA, NIA, arrays, algebraic data-types, bit-vectors
- ▶ **New in 2021 (v2.0.6):**
Better support for the extensional theory of arrays, support more theory combinations (e.g., arrays + ADTs)
- ▶ Scala/Java API
- ▶ Support for linear + non-linear clauses
- ▶ <https://github.com/uuverifiers/eldarica>

Ultimate Unihorn, Ultimate TreeAutomizer

- Entered in the same version as in 2020

Settings in the Competition Runs

- Normal tracks:
 - Timeout 1800s, CPU time
 - Memory restricted to 64GB
 - Two jobs per StarExec node, two cores for each job

- Parallel track (LRA-TS-parallel)
 - Timeout 1800s, **wall-clock time**
 - Same memory limit
 - One job per StarExec node, four cores for each job

Issues Encountered during the Competition

- Spacer
 - StarExec trouble, fixed within a few hours
- Golem
 - Incorrect answer on one LIA-Lin benchmark
 - Authors submitted a fixed version
- PCSat
 - Incorrect answer on one ADT-NonLin benchmarks
 - Authors submitted a fixed version
- RInGen
 - Some frontend issues; solved quickly before the deadline

The Results

Results in LIA-Lin

solver	cnt	ok	sat	uns	fld	to	mo	time	real	space	uniq
Spacer	585	484	310	174	101	101	0	196494	98301	1437	41
Eldarica	585	397	258	139	188	188	0	367839	200046	8964	27
Golem	585	312	179	133	273	247	0	475962	481432	1145	2
Ultimate Unihorn	585	301	175	126	284	258	0	498519	410585	21128	0
PCSat	585	278	181	97	307	307	0	570522	538048	430	1
Ultimate TreeAutomizer	585	207	106	101	378	313	0	580853	547293	20750	0

cnt	number of benchmarks
ok	number of solved benchmarks
sat	number of SAT solved
uns	number of UNSAT solved
fld	number of failed unsolved
to	number of timeouts
mo	number of memory outs
time	sum of total cpu time in seconds
real	sum of total wall-clock time in seconds
space	sum of memory used in MB
uniq	number of unique benchmarks solved

Results in LIA-NonLin

solver	cnt	ok	sat	uns	fld	to	mo	time	real	space	uniq
Spacer	581	550	352	198	31	26	0	70325	35613	358	80
Eldarica	581	461	285	176	120	120	0	252300	120997	8941	5
Ultimate Unihorn	581	295	175	120	286	280	0	562355	471683	21116	0
PCSat	581	280	172	108	301	294	0	555680	553500	433	0
Ultimate TreeAutomizer	581	56	21	35	525	251	0	463165	433393	20610	0

cnt	number of benchmarks
ok	number of solved benchmarks
sat	number of SAT solved
uns	number of UNSAT solved
fld	number of failed unsolved
to	number of timeouts
mo	number of memory outs
time	sum of total cpu time in seconds
real	sum of total wall-clock time in seconds
space	sum of memory used in MB
uniq	number of unique benchmarks solved

Results in LIA-Lin-Arrays

solver	cnt	ok	sat	uns	fld	to	mo	time	real	space	uniq
Spacer	488	288	214	74	200	195	0	353882	177769	399	89
Eldarica	488	225	149	76	263	240	0	441240	208321	7502	11
Ultimate Unihorn	488	219	146	73	269	241	0	462492	402093	17718	1
Ultimate TreeAutomizer	488	147	100	47	341	212	0	387636	369112	17311	0

cnt	number of benchmarks
ok	number of solved benchmarks
sat	number of SAT solved
uns	number of UNSAT solved
fld	number of failed unsolved
to	number of timeouts
mo	number of memory outs
time	sum of total cpu time in seconds
real	sum of total wall-clock time in seconds
space	sum of memory used in MB
uniq	number of unique benchmarks solved

Results in LIA-NonLin-Arrays

solver	cnt	ok	sat	uns	fld	to	mo	time	real	space	uniq
Spacer	450	379	224	155	71	52	0	117861	62320	461	135
Eldarica	450	225	135	90	225	217	0	415182	200473	6917	10
Ultimate Unihorn	450	205	108	97	245	149	0	312415	250274	16221	0
Ultimate TreeAutomizer	450	88	21	67	362	161	0	299039	273337	15966	1

cnt	number of benchmarks
ok	number of solved benchmarks
sat	number of SAT solved
uns	number of UNSAT solved
fld	number of failed unsolved
to	number of timeouts
mo	number of memory outs
time	sum of total cpu time in seconds
real	sum of total wall-clock time in seconds
space	sum of memory used in MB
uniq	number of unique benchmarks solved

Results in LRA-TS

solver	cnt	ok	sat	uns	fld	to	mo	time	real	space	uniq
Spacer	498	311	228	83	187	178	0	364422	188832	515	51
Golem	498	276	200	76	222	222	0	416173	416193	1645	11
Ultimate TreeAutomizer	498	169	132	37	329	328	0	632058	606468	17662	15
Ultimate Unihorn	498	160	103	57	338	335	0	646526	545329	18029	1

Results in LRA-TS-parallel *(4 cores, 1800s wall-clock time)*

solver	cnt	ok	sat	uns	fld	to	mo	time	real	space	uniq
Spacer	498	335	250	85	163	153	0	642476	329157	739	61
Golem	498	276	200	76	222	222	0	416152	416229	1637	6
Ultimate TreeAutomizer	498	169	132	37	329	328	0	652793	626474	17663	14
Ultimate Unihorn	498	166	109	57	332	329	0	804867	630927	18034	1

Results in ADT-Nonlin

solver	cnt	ok	sat	uns	fld	to	mo	time	real	space	uniq
Spacer	178	83	30	53	95	95	0	173478	173499	139	10
Eldarica	178	77	38	39	101	98	0	182413	87101	2741	8
RInGen	178	71	26	45	107	107	0	194180	194199	698	6
PCSat	178	64	33	31	114	114	0	209696	209725	150	6

cnt	number of benchmarks
ok	number of solved benchmarks
sat	number of SAT solved
uns	number of UNSAT solved
fld	number of failed unsolved
to	number of timeouts
mo	number of memory outs
time	sum of total cpu time in seconds
real	sum of total wall-clock time in seconds
space	sum of memory used in MB
uniq	number of unique benchmarks solved

Congratulations!

	LIA-Lin	LIA-NonLin	LIA-Lin-Ar	LIA-Nonlin-Ar	LRA-TS	LRA-TS-par	ADT-Nonlin
Winner:	Spacer	Spacer	Spacer	Spacer	Spacer	Spacer	Spacer
2nd:	Golem	Ultimate Unihorn	Ultimate Unihorn	Ultimate Unihorn	Golem	Golem	RInGen
3rd:	Ultimate Unihorn	PCSat	Ultimate Tree-Automizer	Ultimate Tree-Automizer	Ultimate Tree-Automizer	Ultimate Tree-Automizer	PCSat

+ Eldarica for many unofficial second places

Big Thanks to



Discussion

- New benchmarks / new tracks?
- Format extensions?
- New solvers?
- Solution validation?
- CHC-COMP 2022 organizers?



We want your Horn solver for 2022!