Benchmarking of Java Verification Tools at the Software Verification Competition (SV-COMP)

Lucas Cordeiro
Daniel Kroening
Peter Schrammel

JPF Workshop 2018

What is SV-COMP?

https://sv-comp.sosy-lab.org

Annual comparative evaluation of fully automatic software verifiers

- Reflect state of the art w.r.t. effectiveness and efficiency
- Promote reproducibility and validity of experimental results
- Increase the visibility and credits for tool developers
- Establish set of benchmarks for software verification community

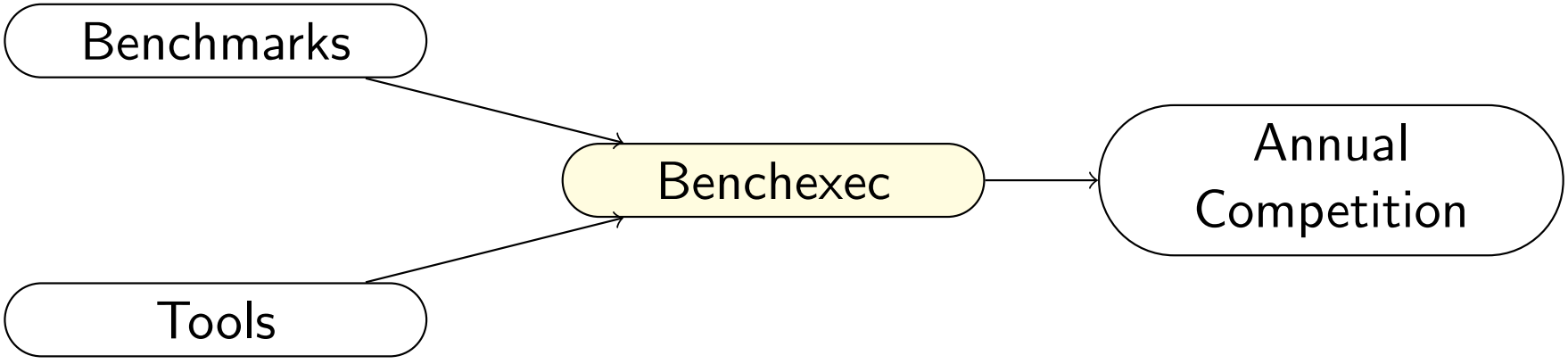
Started in 2012 for C programs, 8th edition in 2019:

- More than 10000 benchmarks
- More than 30 participants
- NEW: Java track

Goals of this talk

- Understand how SV-COMP works
- Know about the Java track at SV-COMP 2019
- Be able to use it for your own research
- Be able to contribute tools and benchmarks

How does SV-COMP work?



https://dblp.org/db/conf/tacas/tacas2017-2

Search

SV COMP

Dirk Beyer:
Software Verification with Validation of Results - (Report on SV-COMP 2017). 331-349

Jera Hensel, Frank Emrich, Florian Frohn, Thomas Ströder, Jürgen Giesl:
AProVE: Proving and Disproving Termination of Memory-Manipulating C Programs - (Competition Contribution). 350-354

Pavel Andrianov, Karlheinz Friedberger, Mikhail U. Mandrykin, Vadim S. Mutilin, Anton Volkov:
CPA-BAM-BnB: Block-Abstraction Memoization and Region-Based Memory Models for Predicate Abstractions - (Competition Contribution). 355-359

Williame Rocha, Herbert Rocha, Hussama Ismail, Lucas C. Cordeiro, Bernd Fischer:
DepthK: A k-Induction Verifier Based on Invariant Inference for C Programs - (Competition Contribution). 360-364

Lukás Holík, Martin Hruska, Ondrej Lengál, Adam Rogalewicz, Jirí Simácek, Tomás Vojnar:
Forester: From Heap Shapes to Automata Predicates - (Competition Contribution). 365-369

Benchmarks

`https://github.com/sosy-lab/sv-benchmarks`

For each **verification task** (aka benchmark)

- Source files (open source license)
- Descriptor (`.yml` file)
  - File name is name of the benchmark
  - Reference to one or more **properties** (`.prp` files)
    - unreach-call:
      `CHECK(init(main()), LTL(G !call(__VERIFIER_error())))`
    - termination:
      `CHECK(init(main()), LTL(F end))`
    - no-overflow, valid-memsafety, ...
  - Expected **answer**: true, false(property), unknown

Benchmarks

https://github.com/sosy-lab/sv-benchmarks

For each **verification task** (aka benchmark)

- Source files (open source license)
- Descriptor (`.yml` file)
  - File name is name of the benchmark
  - Reference to one or more **properties** (`.prp` files)
    - unreach-call:
      `CHECK(init(main()), LTL(G !call(__VERIFIER_error())))`
    - termination:
      `CHECK(init(main()), LTL(F end))`
    - no-overflow, valid-memsafety, ...
  - Expected **answer**: true, false(property), unknown

Categories defined as subsets (`.set` files)

- ReachSafety, ConcurrencySafety, MemorySafety, NoOverflows, Termination, ...
- There are sub-categories (loops, arrays, heap, ...).

Benchexec

https://github.com/sosy-lab/benchexec

Benchmark execution software (Beyer et al SPIN'15)

- Implemented in Python 3
- Resource-limited execution (8 cores, 15GB, 900s CPU time)
- Interfaces to **competition candidates** (aka tools) via **tool-info** modules
  - Name, version
  - Build command line
  - Parse answer (true, false, unknown) from tool output

Benchexec

https://github.com/sosy-lab/benchexec

Benchmark execution software (Beyer et al SPIN'15)

- Implemented in Python 3
- Resource-limited execution (8 cores, 15GB, 900s CPU time)
- Interfaces to **competition candidates** (aka tools) via **tool-info** modules
  - Name, version
  - Build command line
  - Parse answer (true, false, unknown) from tool output
- Table-generator to generate HTML table of results

Benchexec

https://github.com/sosy-lab/benchexec

Benchmark execution software (Beyer et al SPIN'15)

- Implemented in Python 3
- Resource-limited execution (8 cores, 15GB, 900s CPU time)
- Interfaces to **competition candidates** (aka tools) via **tool-info** modules
  - Name, version
  - Build command line
  - Parse answer (true, false, unknown) from tool output
- Table-generator to generate HTML table of results
- SV-COMP 2019 runs on Ubuntu 18.04

Benchexec

https://github.com/sosy-lab/benchexec

Benchmark execution software (Beyer et al SPIN'15)

- Implemented in Python 3
- Resource-limited execution (8 cores, 15GB, 900s CPU time)
- Interfaces to **competition candidates** (aka tools) via **tool-info** modules
  - Name, version
  - Build command line
  - Parse answer (true, false, unknown) from tool output
- Table-generator to generate HTML table of results
- SV-COMP 2019 runs on Ubuntu 18.04

E.g. run CBMC on sub-category ReachSafety-BitVectors:

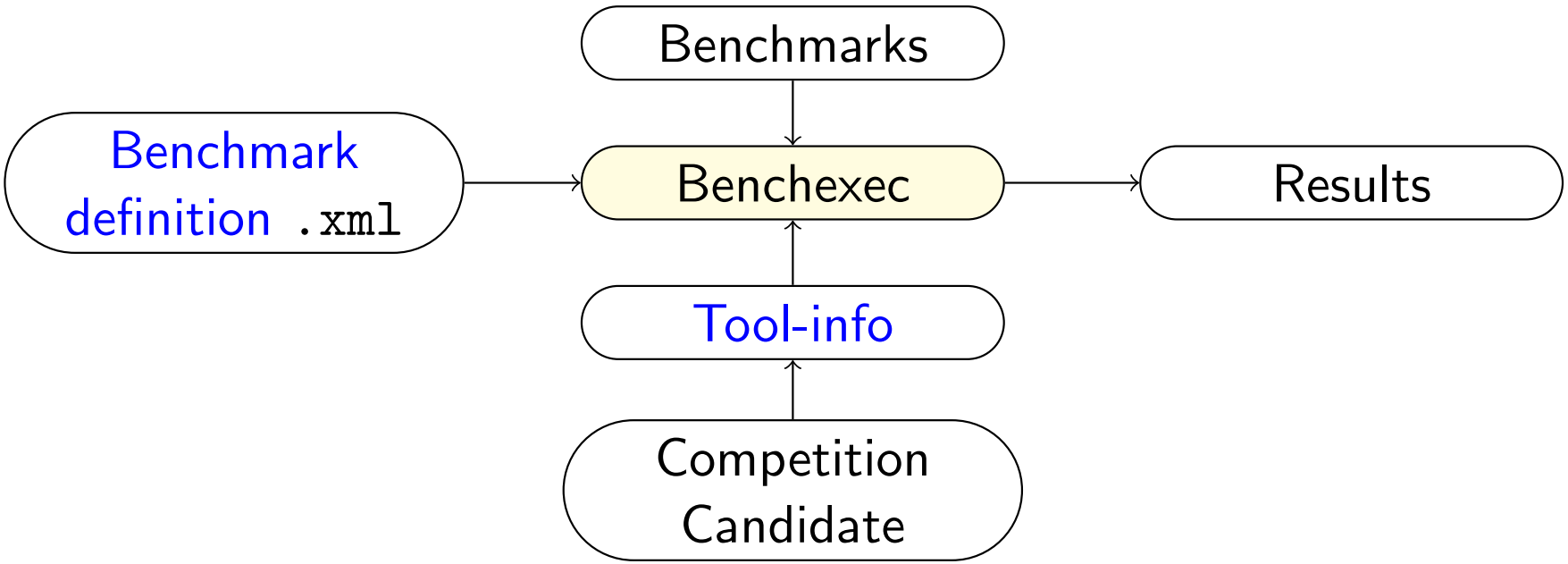
```
bin/benchexec cbmc.xml -t ReachSafety-BitVectors
```

Benchmark definition

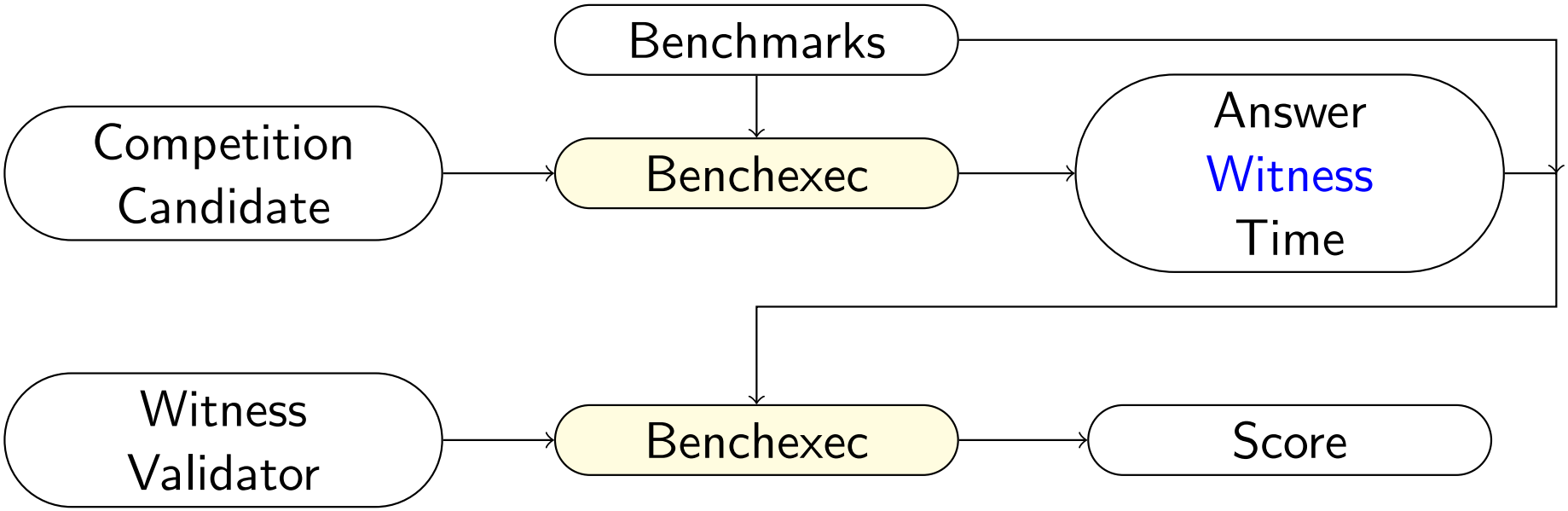
https://github.com/sosy-lab/sv-comp

Benchmark definition (tool.xml):

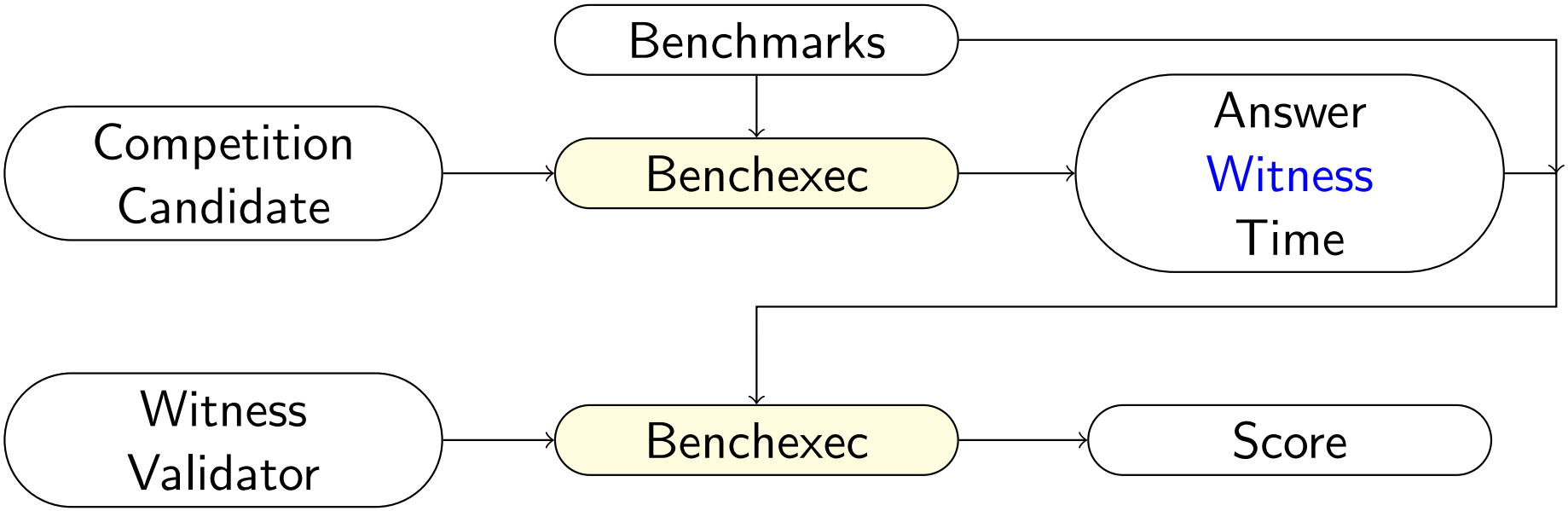
- Reference to tool-info module
- Resource limits
- Global options for tool
- Definition of (sub-)categories:
  - Reference to category .set files
  - Reference to property .prp file



Annual Competition



Annual Competition



Witness Validation

(Beyer et al FSE'15, FSE'16)

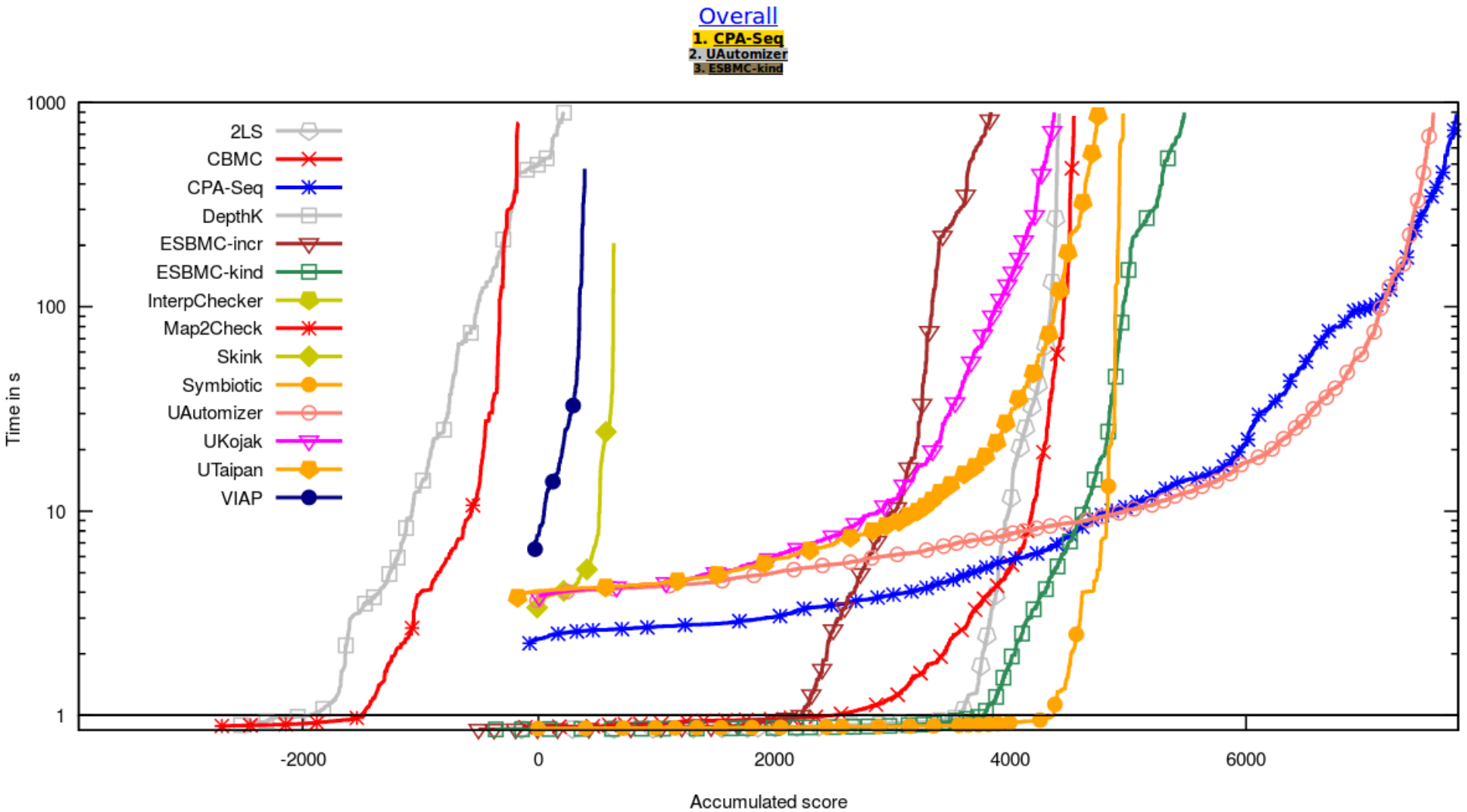
- Correctness and refutation witnesses (`.graphml`)
- Validated by witness validation tools

Scores

Points	Reported result	Description
0	UNKNOWN	Failure to compute verification result, out of resources, program crash.
+1	FALSE correct	The error in the program was found and a violation witness was confirmed.
−16	FALSE incorrect	An error is reported for a program that fulfills the specification (false alarm, incomplete analysis).
+2	TRUE correct	The program was analyzed to be free of errors and a correctness witness was confirmed.
+1	TRUE correct, witness unconfirmed	The program was analyzed to be free of errors but the correctness witness was not confirmed.
−32	TRUE incorrect	The program had an error but the competition candidate did not find it (missed bug, unsound analysis).

https://sv-comp.sosy-lab.org/2019/rules.php

Annual Competition



https://sv-comp.sosy-lab.org/2018/results/results-verified/

Annual Competition

Tool	JPF 32				JayHorn 5.1				JBMC 5.8			
Limits	timelimit: 180 s, memlimit: 15000 MB, CPU core limit: 8											
Host	localhost											
OS	Linux 4.4.0-116-generic x86_64											
System	CPU: Intel Core i7-6700HQ CPU @ 2.60GHz with 8 cores, frequency: 3500 MHz, Turbo Boost enabled; RAM: 16014056 kB											
Date of execution	2018-03-10 14:09:57 GMT				2018-03-10 11:39:56 GMT				2018-03-10 16:54:00 GMT			
Run set	jpf.sv-comp18				jayhorn.sv-comp18				jbmc.sv-comp18			
../git-sv-benchmarks/java/	status	cputime	walltime	memUsage	status	cputime	walltime	memUsage	status	cputime	walltime	memUsage
jbmc-regression/ArithmeticException1_false-assert.jar	false(reach)	1.252s	0.492s	86929408	true	3.748s	1.352s	203776000	false(reach)	0.207s	0.224s	17719296
jbmc-regression/ArithmeticException5_true-assert.jar	false(reach)	1.208s	0.479s	85815296	true	3.564s	1.169s	184410112	true	0.212s	0.226s	12709888
jbmc-regression/ArithmeticException6_false-assert.jar	true	1.172s	0.452s	82345984	true	3.448s	1.164s	174071808	false(reach)	0.251s	0.264s	17477632
jbmc-regression/ArrayIndexOutOfBoundsException1_false-assert.jar	true	1.164s	0.475s	83734528	false(reach)	6.516s	1.934s	228200448	false(reach)	0.256s	0.268s	16728064
jbmc-regression/ArrayIndexOutOfBoundsException2_false-assert.jar	true	1.140s	0.443s	81854464	false(reach)	6.748s	2.030s	230465536	false(reach)	0.250s	0.257s	18628608
jbmc-regression/ArrayIndexOutOfBoundsException3_false-assert.jar	true	1.256s	0.501s	82014208	false(reach)	7.204s	2.023s	233783296	false(reach)	0.223s	0.236s	18378752
jbmc-regression/BufferedReaderReadLine_false-assert.jar	true	1.200s	0.464s	85073920	false(reach)	11.484s	2.992s	264642560	false(reach)	0.373s	0.381s	29315072
jbmc-regression/CharSequenceBug_false-assert.jar	true	1.136s	0.444s	79372288	false(reach)	7.336s	2.191s	233238528	false(reach)	1.407s	1.422s	26890240
jbmc-regression/CharSequenceToString_true-assert.jar	true	1.136s	0.501s	80220160	false(reach)	7.412s	2.186s	235905024	true	1.047s	1.059s	36552704
jbmc-regression/ClassCastException1_false-assert.jar	false(reach)	1.172s	0.462s	86487040	false(reach)	6.124s	1.898s	230752256	false(reach)	0.305s	0.319s	18665472
jbmc-regression/ClassCastException2_true-assert.jar	true	1.216s	0.506s	77561856	true	6.512s	1.907s	233218048	true	0.233s	0.253s	13275136
jbmc-regression/ClassCastException3_false-assert.jar	false(reach)	1.148s	0.441s	84291584	false(reach)	6.852s	2.080s	233496576	false(reach)	0.235s	0.242s	17874944
jbmc-regression/Class_method1_true-assert.jar	true	1.160s	0.457s	80973824	true	5.148s	1.609s	236470272	true	0.236s	0.256s	13553664
jbmc-regression/Inheritance1_true-assert.jar	true	1.180s	0.482s	83697664	true	7.240s	2.046s	230694912	true	0.260s	0.276s	13492224
jbmc-regression/NegativeArraySizeException1_false-assert.jar	false(reach)	1.236s	0.492s	87400448	true	3.720s	1.230s	185139200	false(reach)	0.213s	0.225s	16687104
jbmc-regression/NegativeArraySizeException2_false-assert.jar	false(reach)	1.176s	0.472s	83542016	true	3.908s	1.245s	185753600	false(reach)	0.241s	0.247s	18546688
jbmc-regression/NullPointerException1_false-assert.jar	false(reach)	1.152s	0.440s	84705280	unknown	1.864s	0.686s	136036352	false(reach)	0.243s	0.245s	16834560
jbmc-regression/NullPointerException2_false-assert.jar	false(reach)	1.212s	0.466s	85893120	unknown	1.740s	0.676s	134901760	false(reach)	0.244s	0.251s	16654336
jbmc-regression/NullPointerException3_false-assert.jar	false(reach)	1.368s	0.532s	88522752	unknown	1.668s	0.616s	133578752	false(reach)	0.243s	0.255s	16982016
jbmc-regression/NullPointerException4_false-assert.jar	false(reach)	1.272s	0.535s	77479936	unknown	1.636s	0.626s	128970752	false(reach)	0.240s	0.248s	18771968
jbmc-regression/RegexMatches01_true-assert.jar	true	1.300s	0.537s	80244736	false(reach)	8.344s	2.393s	246411264	true	1.280s	1.301s	39464960
jbmc-regression/RegexMatches02_false-assert.jar	true	1.192s	0.481s	85598208	false(reach)	9.824s	2.621s	254140416	false(reach)	1.033s	1.044s	64978944
jbmc-regression/RegexSubstitution01_true-assert.jar	true	1.252s	0.503s	83034112	false(reach)	14.268s	3.535s	394878976	true	72.774s	72.850s	757309440
jbmc-regression/RegexSubstitution02_false-assert.jar	true	1.116s	0.442s	81616896	false(reach)	17.136s	4.284s	390799360	true	40.973s	41.011s	526553088
jbmc-regression/RegexSubstitution03_true-assert.jar	true	1.232s	0.465s	86216704	false(reach)	12.680s	3.158s	338685952	true	1.754s	1.766s	46804992
jbmc-regression/StaticCharMethods01_true-assert.jar	true	1.176s	0.502s	82903040	false(reach)	6.744s	2.041s	228970496	true	0.261s	0.276s	14376960
jbmc-regression/StaticCharMethods02_false-assert.jar	true	1.196s	0.484s	83034112	false(reach)	8.524s	2.476s	237215744	false(reach)	0.829s	0.854s	50974720
jbmc-regression/StaticCharMethods03_false-assert.jar	true	1.192s	0.486s	82206720	false(reach)	10.188s	2.888s	243417088	false(reach)	0.847s	0.858s	50806784
jbmc-regression/StaticCharMethods04_false-assert.jar	true	1.196s	0.487s	84496384	false(reach)	8.664s	2.529s	231280640	false(reach)	0.821s	0.829s	51412992

Competition Timeline

- September: Contribution of benchmarks
- October: Tool registration and qualification
- November: Tool submission
- December: Announcement of winners
- January: Tool paper submission
- April: SV-COMP session at ETAPS

Java Track at SV-COMP

Objectives:

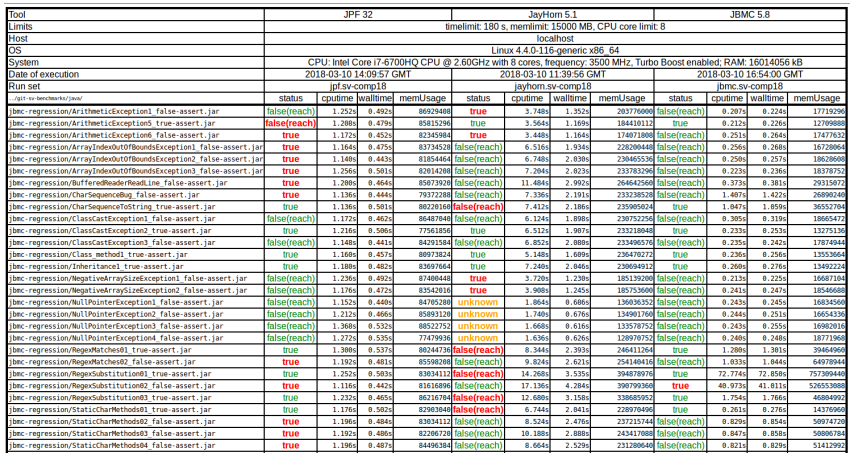
- More languages in SV-COMP
- Standard benchmark set
- Comparability
- Reproducibility
- Re-use existing benchmarking infrastructure

Tool	JPF 32				JayHorn 5.1				JBMC 5.8			
Limits	timelimit: 180 s, memlimit: 15000 MB, CPU core limit: 8											
Host	localhost											
OS	Linux 4.4.0-116-generic x86_64											
System	CPU: Intel Core i7-6700HQ CPU @ 2.60GHz with 8 cores, frequency: 3500 MHz, Turbo Boost enabled: RAM: 16014056 kB											
Date of execution	2018-03-10 14:09:57 GMT				2018-03-10 11:39:56 GMT				2018-03-10 16:54:00 GMT			
Run set	jpf.sv-comp18				jayhorn.sv-comp18				jbmc.sv-comp18			
.../git-sv-benchmarks/java/	status	cputime	walltime	memUsage	status	cputime	walltime	memUsage	status	cputime	walltime	memUsage
jbmc-regression/ArithmeticException1_false-assert.jar	false(reach)	1.252s	0.492s	86929408	**true**	3.748s	1.352s	203776000	false(reach)	0.207s	0.224s	17719296
jbmc-regression/ArithmeticException5_true-assert.jar	**false(reach)**	1.208s	0.479s	85815296	true	3.564s	1.169s	184410112	true	0.212s	0.226s	12709888
jbmc-regression/ArithmeticException6_false-assert.jar	**true**	1.172s	0.452s	82345984	**true**	3.448s	1.164s	174071808	false(reach)	0.251s	0.264s	17477632
jbmc-regression/ArrayIndexOutOfBoundsException1_false-assert.jar	**true**	1.164s	0.475s	83734528	false(reach)	6.516s	1.934s	228200448	false(reach)	0.256s	0.268s	16728064
jbmc-regression/ArrayIndexOutOfBoundsException2_false-assert.jar	**true**	1.140s	0.443s	81854464	false(reach)	6.748s	2.030s	230465536	false(reach)	0.250s	0.257s	18628608
jbmc-regression/ArrayIndexOutOfBoundsException3_false-assert.jar	**true**	1.256s	0.501s	82614288	false(reach)	7.204s	2.023s	233783296	false(reach)	0.223s	0.236s	18378752
jbmc-regression/BufferedReaderReadLine_false-assert.jar	**true**	1.208s	0.464s	85073920	false(reach)	11.484s	2.992s	264642560	false(reach)	0.373s	0.381s	29315072
jbmc-regression/CharSequenceBug_false-assert.jar	**true**	1.136s	0.444s	79372288	false(reach)	7.336s	2.191s	233238528	false(reach)	1.407s	1.422s	26896240
jbmc-regression/CharSequenceToString_true-assert.jar	true	1.136s	0.501s	80220160	**false(reach)**	7.412s	2.186s	235905024	true	1.047s	1.059s	36552704
jbmc-regression/ClassCastException1_false-assert.jar	false(reach)	1.172s	0.462s	86487040	false(reach)	6.124s	1.898s	230752256	false(reach)	0.305s	0.319s	18665472
jbmc-regression/ClassCastException2_true-assert.jar	true	1.216s	0.506s	77561856	true	6.512s	1.907s	233218048	true	0.233s	0.253s	13275136
jbmc-regression/ClassCastException3_false-assert.jar	false(reach)	1.148s	0.441s	84291584	false(reach)	6.852s	2.080s	233496576	false(reach)	0.235s	0.242s	17874944
jbmc-regression/Class_method1_true-assert.jar	true	1.160s	0.457s	80973824	true	5.148s	1.609s	236470272	true	0.236s	0.256s	13553664
jbmc-regression/Inheritance1_true-assert.jar	true	1.188s	0.482s	83697664	true	7.240s	2.046s	230694912	true	0.269s	0.276s	13492224
jbmc-regression/NegativeArraySizeException1_false-assert.jar	false(reach)	1.236s	0.492s	87400448	**true**	3.720s	1.230s	185139200	false(reach)	0.213s	0.225s	16687104
jbmc-regression/NegativeArraySizeException2_false-assert.jar	false(reach)	1.176s	0.472s	83542016	**true**	3.908s	1.245s	185753600	false(reach)	0.241s	0.247s	18546688
jbmc-regression/NullPointerException1_false-assert.jar	false(reach)	1.152s	0.440s	84705280	unknown	1.864s	0.686s	136036352	false(reach)	0.243s	0.245s	16834560
jbmc-regression/NullPointerException2_false-assert.jar	false(reach)	1.212s	0.466s	85893120	unknown	1.740s	0.676s	134901760	false(reach)	0.244s	0.251s	16654336
jbmc-regression/NullPointerException3_false-assert.jar	false(reach)	1.368s	0.532s	88522752	unknown	1.668s	0.616s	133578752	false(reach)	0.243s	0.255s	16982016
jbmc-regression/NullPointerException4_false-assert.jar	false(reach)	1.272s	0.535s	77479936	unknown	1.636s	0.626s	128970752	false(reach)	0.240s	0.248s	18771968
jbmc-regression/RegexMatches01_true-assert.jar	true	1.308s	0.537s	80244736	**false(reach)**	8.344s	2.393s	246411264	true	1.200s	1.301s	39464960
jbmc-regression/RegexMatches02_false-assert.jar	**true**	1.192s	0.481s	85598208	false(reach)	9.824s	2.621s	254140416	false(reach)	1.033s	1.044s	64978944
jbmc-regression/RegexSubstitution01_true-assert.jar	true	1.252s	0.503s	83034112	**false(reach)**	14.268s	3.535s	394878976	true	72.774s	72.858s	757309440
jbmc-regression/RegexSubstitution02_false-assert.jar	**true**	1.116s	0.442s	81616896	false(reach)	17.136s	4.284s	390799360	**true**	40.973s	41.011s	526553088
jbmc-regression/RegexSubstitution03_true-assert.jar	true	1.232s	0.465s	86216704	**false(reach)**	12.680s	3.158s	338685952	true	1.754s	1.766s	46804992
jbmc-regression/StaticCharMethods01_true-assert.jar	true	1.176s	0.502s	82903040	**false(reach)**	6.744s	2.041s	228978496	true	0.261s	0.276s	14376960
jbmc-regression/StaticCharMethods02_false-assert.jar	**true**	1.196s	0.484s	83034112	false(reach)	8.524s	2.476s	237215744	false(reach)	0.829s	0.854s	50974720
jbmc-regression/StaticCharMethods03_false-assert.jar	**true**	1.192s	0.486s	82206720	false(reach)	10.188s	2.888s	243417088	false(reach)	0.847s	0.858s	50806784
jbmc-regression/StaticCharMethods04_false-assert.jar	**true**	1.196s	0.487s	84496384	false(reach)	8.664s	2.529s	231280640	false(reach)	0.821s	0.829s	51412992

Java Track at SV-COMP

Objectives:

- More languages in SV-COMP
- Standard benchmark set
- Comparability
- Reproducibility
- Re-use existing benchmarking infrastructure



Tool	JPF 32				JayHorn 5.1				JBMC 5.8			
Limits	timelimit: 180 s, memlimit: 15000 MB, CPU core limit: 8											
Host	localhost											
OS	Linux 4.4.0-116-generic x86_64											
System	CPU: Intel Core i7-6700HQ CPU @ 2.60GHz with 8 cores, frequency: 3500 MHz, Turbo Boost enabled: RAM: 16014056 kB											
Date of execution	2018-03-10 14:09:57 GMT				2018-03-10 11:39:56 GMT				2018-03-10 16:54:00 GMT			
Run set	jpf.sv-comp18				jayhorn.sv-comp18				jbmc.sv-comp18			
.../git-sv-benchmarks/java/	status	cputime	walltime	memUsage	status	cputime	walltime	memUsage	status	cputime	walltime	memUsage
jbmc-regression/ArithmeticException1_false-assert.jar	false(reach)	1.252s	0.492s	86929408	true	3.748s	1.352s	203776000	false(reach)	0.207s	0.224s	17719296
jbmc-regression/ArithmeticException5_true-assert.jar	false(reach)	1.208s	0.479s	85815296	true	3.564s	1.169s	184410112	true	0.212s	0.226s	12709888
jbmc-regression/ArithmeticException6_false-assert.jar	true	1.172s	0.452s	82345984	true	3.448s	1.164s	174071808	false(reach)	0.251s	0.264s	17477632
jbmc-regression/ArrayIndexOutOfBoundsException1_false-assert.jar	true	1.164s	0.475s	83734528	false(reach)	6.516s	1.934s	228200448	false(reach)	0.256s	0.268s	16728064
jbmc-regression/ArrayIndexOutOfBoundsException2_false-assert.jar	true	1.140s	0.443s	81854464	false(reach)	6.748s	2.030s	230465536	false(reach)	0.250s	0.257s	18628608
jbmc-regression/ArrayIndexOutOfBoundsException3_false-assert.jar	true	1.256s	0.501s	82014208	false(reach)	7.204s	2.023s	233783296	false(reach)	0.223s	0.236s	18378752
jbmc-regression/BufferedReaderReadLine_false-assert.jar	true	1.200s	0.464s	85073920	false(reach)	11.484s	2.992s	264642560	false(reach)	0.373s	0.381s	29315072
jbmc-regression/CharSequenceBug_false-assert.jar	true	1.136s	0.444s	79372288	false(reach)	7.336s	2.191s	233238528	false(reach)	1.407s	1.422s	26890240
jbmc-regression/CharSequenceToString_true-assert.jar	true	1.136s	0.501s	80220160	false(reach)	7.412s	2.186s	235905024	true	1.047s	1.059s	36552704
jbmc-regression/ClassCastException1_false-assert.jar	false(reach)	1.172s	0.462s	86487040	false(reach)	6.124s	1.898s	230752256	false(reach)	0.305s	0.319s	18665472
jbmc-regression/ClassCastException2_true-assert.jar	true	1.216s	0.506s	77561856	true	6.512s	1.907s	233218048	true	0.233s	0.253s	13275136
jbmc-regression/ClassCastException3_false-assert.jar	false(reach)	1.148s	0.441s	84291584	false(reach)	6.852s	2.080s	233496576	false(reach)	0.235s	0.242s	17874944
jbmc-regression/Class_method1_true-assert.jar	true	1.168s	0.457s	80973824	true	5.148s	1.699s	236470272	true	0.236s	0.256s	13553664
jbmc-regression/Inheritance1_true-assert.jar	true	1.188s	0.482s	83697664	true	7.240s	2.046s	230694912	true	0.269s	0.276s	13492224
jbmc-regression/NegativeArraySizeException1_false-assert.jar	false(reach)	1.236s	0.492s	87400448	true	3.720s	1.230s	185139200	false(reach)	0.213s	0.225s	16687104
jbmc-regression/NegativeArraySizeException2_false-assert.jar	false(reach)	1.176s	0.472s	83542016	true	3.908s	1.245s	185753600	false(reach)	0.241s	0.247s	18546688
jbmc-regression/NullPointerException1_false-assert.jar	false(reach)	1.152s	0.440s	84705280	unknown	1.864s	0.686s	136036352	false(reach)	0.243s	0.245s	16834560
jbmc-regression/NullPointerException2_false-assert.jar	false(reach)	1.212s	0.466s	85893120	unknown	1.740s	0.676s	134901760	false(reach)	0.244s	0.251s	16654336
jbmc-regression/NullPointerException3_false-assert.jar	false(reach)	1.368s	0.532s	88522752	unknown	1.668s	0.616s	133578752	false(reach)	0.243s	0.255s	16982016
jbmc-regression/NullPointerException4_false-assert.jar	false(reach)	1.272s	0.535s	77479936	unknown	1.636s	0.626s	128970752	false(reach)	0.240s	0.248s	18771968
jbmc-regression/RegexMatches01_true-assert.jar	true	1.308s	0.537s	80244736	false(reach)	8.344s	2.393s	246411264	true	1.280s	1.301s	39464960
jbmc-regression/RegexMatches02_false-assert.jar	true	1.192s	0.481s	85598208	false(reach)	9.824s	2.621s	254140416	false(reach)	1.033s	1.044s	64978944
jbmc-regression/RegexSubstitution01_true-assert.jar	true	1.252s	0.503s	83034112	false(reach)	14.268s	3.535s	394878976	true	72.774s	72.858s	757309440
jbmc-regression/RegexSubstitution02_false-assert.jar	true	1.116s	0.442s	81616896	false(reach)	17.136s	4.284s	390799360	true	40.973s	41.011s	526553088
jbmc-regression/RegexSubstitution03_true-assert.jar	true	1.232s	0.465s	86216704	false(reach)	12.680s	3.158s	338685952	true	1.754s	1.766s	46804992
jbmc-regression/StaticCharMethods01_true-assert.jar	true	1.176s	0.502s	82903040	false(reach)	6.744s	2.041s	228978496	true	0.261s	0.276s	14376960
jbmc-regression/StaticCharMethods02_false-assert.jar	true	1.196s	0.484s	83034112	false(reach)	8.524s	2.476s	237215744	false(reach)	0.829s	0.854s	50974720
jbmc-regression/StaticCharMethods03_false-assert.jar	true	1.192s	0.486s	82206720	false(reach)	10.188s	2.888s	243417088	false(reach)	0.847s	0.858s	50806784
jbmc-regression/StaticCharMethods04_false-assert.jar	true	1.196s	0.487s	84496384	false(reach)	8.664s	2.529s	231280640	false(reach)	0.821s	0.829s	51412992

Current status:

- March 2018: Initial benchmarks collected, proof-of-concept tools integrated, potential participants contacted

Java Track at SV-COMP

Objectives:

- More languages in SV-COMP
- Standard benchmark set
- Comparability
- Reproducibility
- Re-use existing benchmarking infrastructure

Tool	JPF 32				JayHorn 5.1				JBMC 5.8			
Limits	timelimit: 180 s, memlimit: 15000 MB, CPU core limit: 8											
Host	localhost											
OS	Linux 4.4.0-116-generic x86_64											
System	CPU: Intel Core i7-6700HQ CPU @ 2.60GHz with 8 cores, frequency: 3500 MHz, Turbo Boost enabled: RAM: 16014056 kB											
Date of execution	2018-03-10 14:09:57 GMT				2018-03-10 11:39:56 GMT				2018-03-10 16:54:00 GMT			
Run set	jpf.sv-comp18				jayhorn.sv-comp18				jbmc.sv-comp18			
.../git-sv-benchmarks/java/	status	cputime	walltime	memUsage	status	cputime	walltime	memUsage	status	cputime	walltime	memUsage
jbmc-regression/ArithmeticException1_false-assert.jar	false(reach)	1.252s	0.492s	86929408	true	3.748s	1.352s	203776000	false(reach)	0.207s	0.224s	17719296
jbmc-regression/ArithmeticException5_true-assert.jar	false(reach)	1.208s	0.479s	85815296	true	3.564s	1.169s	184410112	true	0.212s	0.226s	12709888
jbmc-regression/ArithmeticException6_false-assert.jar	true	1.172s	0.452s	82345984	true	3.448s	1.164s	174071808	false(reach)	0.251s	0.264s	17477632
jbmc-regression/ArrayIndexOutOfBoundsException1_false-assert.jar	true	1.164s	0.475s	83734528	false(reach)	6.516s	1.934s	228200448	false(reach)	0.256s	0.268s	16728064
jbmc-regression/ArrayIndexOutOfBoundsException2_false-assert.jar	true	1.148s	0.443s	81854464	false(reach)	6.748s	2.030s	230465536	false(reach)	0.250s	0.257s	18628608
jbmc-regression/ArrayIndexOutOfBoundsException3_false-assert.jar	true	1.256s	0.501s	82014208	false(reach)	7.204s	2.023s	233783296	false(reach)	0.223s	0.236s	18378752
jbmc-regression/BufferedReaderReadLine_false-assert.jar	true	1.208s	0.464s	85073920	false(reach)	11.484s	2.992s	264642560	false(reach)	0.373s	0.381s	29315072
jbmc-regression/CharSequenceBug_false-assert.jar	true	1.136s	0.444s	79372288	false(reach)	7.336s	2.191s	233238528	false(reach)	1.407s	1.422s	26890240
jbmc-regression/CharSequenceToString_true-assert.jar	true	1.136s	0.501s	80220160	false(reach)	7.412s	2.186s	235905024	true	1.047s	1.059s	36552704
jbmc-regression/ClassCastException1_false-assert.jar	false(reach)	1.172s	0.462s	86487040	false(reach)	6.124s	1.898s	230752256	false(reach)	0.305s	0.319s	18665472
jbmc-regression/ClassCastException2_true-assert.jar	true	1.216s	0.506s	77561856	true	6.512s	1.907s	233218048	true	0.233s	0.253s	13275136
jbmc-regression/ClassCastException3_false-assert.jar	false(reach)	1.148s	0.441s	84291584	false(reach)	6.852s	2.080s	233496576	false(reach)	0.235s	0.242s	17874944
jbmc-regression/Class_method1_true-assert.jar	true	1.168s	0.457s	80973824	true	5.148s	1.699s	236470272	true	0.236s	0.256s	13553664
jbmc-regression/Inheritance1_true-assert.jar	true	1.188s	0.482s	83697664	true	7.240s	2.046s	230694912	true	0.269s	0.276s	13492224
jbmc-regression/NegativeArraySizeException1_false-assert.jar	false(reach)	1.236s	0.492s	87400448	true	3.720s	1.230s	185139200	false(reach)	0.213s	0.225s	16687104
jbmc-regression/NegativeArraySizeException2_false-assert.jar	false(reach)	1.176s	0.472s	83542016	true	3.908s	1.245s	185753600	false(reach)	0.241s	0.247s	18546688
jbmc-regression/NullPointerException1_false-assert.jar	false(reach)	1.152s	0.440s	84705280	unknown	1.864s	0.686s	136036352	false(reach)	0.243s	0.245s	16834560
jbmc-regression/NullPointerException2_false-assert.jar	false(reach)	1.212s	0.466s	85893120	unknown	1.740s	0.676s	134901760	false(reach)	0.244s	0.251s	16654336
jbmc-regression/NullPointerException3_false-assert.jar	false(reach)	1.368s	0.532s	88522752	unknown	1.668s	0.616s	133578752	false(reach)	0.243s	0.255s	16982016
jbmc-regression/NullPointerException4_false-assert.jar	false(reach)	1.272s	0.535s	77479936	unknown	1.636s	0.626s	128970752	false(reach)	0.240s	0.248s	18771968
jbmc-regression/RegexMatches01_true-assert.jar	true	1.308s	0.537s	80244736	false(reach)	8.344s	2.393s	246411264	true	1.280s	1.301s	39464960
jbmc-regression/RegexMatches02_false-assert.jar	true	1.192s	0.481s	85598208	false(reach)	9.824s	2.621s	254140416	false(reach)	1.033s	1.044s	64978944
jbmc-regression/RegexSubstitution01_true-assert.jar	true	1.252s	0.503s	83034112	false(reach)	14.268s	3.535s	394878976	true	72.774s	72.858s	757309440
jbmc-regression/RegexSubstitution02_false-assert.jar	true	1.116s	0.442s	81616896	false(reach)	17.136s	4.284s	390799360	true	40.973s	41.011s	526553088
jbmc-regression/RegexSubstitution03_true-assert.jar	true	1.232s	0.465s	86216704	false(reach)	12.680s	3.158s	338685952	true	1.754s	1.766s	46804992
jbmc-regression/StaticCharMethods01_true-assert.jar	true	1.176s	0.502s	82903040	false(reach)	6.744s	2.041s	228978496	true	0.261s	0.276s	14376960
jbmc-regression/StaticCharMethods02_false-assert.jar	true	1.196s	0.484s	83034112	false(reach)	8.524s	2.476s	237215744	false(reach)	0.829s	0.854s	50974720
jbmc-regression/StaticCharMethods03_false-assert.jar	true	1.192s	0.486s	82206720	false(reach)	10.188s	2.888s	243417088	false(reach)	0.847s	0.858s	50806784
jbmc-regression/StaticCharMethods04_false-assert.jar	true	1.196s	0.487s	84496384	false(reach)	8.664s	2.529s	231280640	false(reach)	0.821s	0.829s	51412992

Current status:

- March 2018: Initial benchmarks collected, proof-of-concept tools integrated, potential participants contacted
- April 2018: Proposal accepted at Jury meeting at TACAS

Java Track at SV-COMP

Objectives:

- More languages in SV-COMP
- Standard benchmark set
- Comparability
- Reproducibility
- Re-use existing benchmarking infrastructure

Tool	JPF 32				JayHorn 5.1				JBMC 5.8			
Limits	timelimit: 180 s, memlimit: 15000 MB, CPU core limit: 8											
Host	localhost											
OS	Linux 4.4.0-116-generic x86_64											
System	CPU: Intel Core i7-6700HQ CPU @ 2.60GHz with 8 cores, frequency: 3500 MHz, Turbo Boost enabled: RAM: 16014056 kB											
Date of execution	2018-03-10 14:09:57 GMT				2018-03-10 11:39:56 GMT				2018-03-10 16:54:00 GMT			
Run set	jpf.sv-comp18				jayhorn.sv-comp18				jbmc.sv-comp18			
.../git-sv-benchmarks/java/	status	cputime	walltime	memUsage	status	cputime	walltime	memUsage	status	cputime	walltime	memUsage
jbmc-regression/ArithmeticException1_false-assert.jar	false(reach)	1.252s	0.492s	86929408	true	3.748s	1.352s	203776000	false(reach)	0.207s	0.224s	17719296
jbmc-regression/ArithmeticException5_true-assert.jar	false(reach)	1.208s	0.479s	85815296	true	3.564s	1.169s	184410112	true	0.212s	0.226s	12709888
jbmc-regression/ArithmeticException6_false-assert.jar	true	1.172s	0.452s	82345984	true	3.448s	1.164s	174071808	false(reach)	0.251s	0.264s	17477632
jbmc-regression/ArrayIndexOutOfBoundsException1_false-assert.jar	true	1.164s	0.475s	83734528	false(reach)	6.516s	1.934s	228200448	false(reach)	0.256s	0.268s	16728064
jbmc-regression/ArrayIndexOutOfBoundsException2_false-assert.jar	true	1.140s	0.443s	81854464	false(reach)	6.748s	2.030s	230465536	false(reach)	0.250s	0.257s	18628608
jbmc-regression/ArrayIndexOutOfBoundsException3_false-assert.jar	true	1.256s	0.501s	82014208	false(reach)	7.204s	2.023s	233783296	false(reach)	0.223s	0.236s	18378752
jbmc-regression/BufferedReaderReadLine_false-assert.jar	true	1.200s	0.464s	85073920	false(reach)	11.484s	2.992s	264642560	false(reach)	0.373s	0.381s	29315072
jbmc-regression/CharSequenceBug_false-assert.jar	true	1.136s	0.444s	79372288	false(reach)	7.336s	2.191s	233238528	false(reach)	1.407s	1.422s	26890240
jbmc-regression/CharSequenceToString_true-assert.jar	true	1.136s	0.501s	80220160	false(reach)	7.412s	2.186s	235905024	true	1.047s	1.059s	36552704
jbmc-regression/ClassCastException1_false-assert.jar	false(reach)	1.172s	0.462s	86487040	false(reach)	6.124s	1.898s	230752256	false(reach)	0.305s	0.319s	18665472
jbmc-regression/ClassCastException2_true-assert.jar	true	1.216s	0.506s	77561856	true	6.512s	1.907s	233218048	true	0.233s	0.253s	13275136
jbmc-regression/ClassCastException3_false-assert.jar	false(reach)	1.148s	0.441s	84291584	false(reach)	6.852s	2.080s	233496576	false(reach)	0.235s	0.242s	17874944
jbmc-regression/Class_method1_true-assert.jar	true	1.168s	0.457s	80973824	true	5.148s	1.699s	236470272	true	0.236s	0.256s	13553664
jbmc-regression/Inheritance1_true-assert.jar	true	1.188s	0.482s	83697664	true	7.240s	2.046s	230694912	true	0.269s	0.276s	13492224
jbmc-regression/NegativeArraySizeException1_false-assert.jar	false(reach)	1.236s	0.492s	87400448	true	3.720s	1.230s	185139200	false(reach)	0.213s	0.225s	16687104
jbmc-regression/NegativeArraySizeException2_false-assert.jar	false(reach)	1.176s	0.472s	83542016	true	3.908s	1.245s	185753600	false(reach)	0.241s	0.247s	18546688
jbmc-regression/NullPointerException1_false-assert.jar	false(reach)	1.152s	0.440s	84705280	unknown	1.864s	0.686s	136036352	false(reach)	0.243s	0.245s	16834560
jbmc-regression/NullPointerException2_false-assert.jar	false(reach)	1.212s	0.466s	85893120	unknown	1.740s	0.676s	134901760	false(reach)	0.244s	0.251s	16654336
jbmc-regression/NullPointerException3_false-assert.jar	false(reach)	1.368s	0.532s	88522752	unknown	1.668s	0.616s	133578752	false(reach)	0.243s	0.255s	16982016
jbmc-regression/NullPointerException4_false-assert.jar	false(reach)	1.272s	0.535s	77479936	unknown	1.636s	0.626s	128970752	false(reach)	0.240s	0.248s	18771968
jbmc-regression/RegexMatches01_true-assert.jar	true	1.308s	0.537s	80244736	false(reach)	8.344s	2.393s	246411264	true	1.280s	1.301s	39464960
jbmc-regression/RegexMatches02_false-assert.jar	true	1.192s	0.481s	85598208	false(reach)	9.824s	2.621s	254140416	false(reach)	1.033s	1.044s	64978944
jbmc-regression/RegexSubstitution01_true-assert.jar	true	1.252s	0.503s	83034112	false(reach)	14.268s	3.535s	394878976	true	72.774s	72.858s	757309440
jbmc-regression/RegexSubstitution02_false-assert.jar	true	1.116s	0.442s	81616896	false(reach)	17.136s	4.284s	390799360	true	40.973s	41.011s	526553088
jbmc-regression/RegexSubstitution03_true-assert.jar	true	1.232s	0.465s	86216704	false(reach)	12.680s	3.158s	338685952	true	1.754s	1.766s	46804992
jbmc-regression/StaticCharMethods01_true-assert.jar	true	1.176s	0.502s	82903040	false(reach)	6.744s	2.041s	228970496	true	0.261s	0.276s	14376960
jbmc-regression/StaticCharMethods02_false-assert.jar	true	1.196s	0.484s	83034112	false(reach)	8.524s	2.476s	237215744	false(reach)	0.829s	0.854s	50974720
jbmc-regression/StaticCharMethods03_false-assert.jar	true	1.192s	0.486s	82206720	false(reach)	10.188s	2.888s	243417088	false(reach)	0.847s	0.858s	50806784
jbmc-regression/StaticCharMethods04_false-assert.jar	true	1.196s	0.487s	84496384	false(reach)	8.664s	2.529s	231280640	false(reach)	0.821s	0.829s	51412992

Current status:

- March 2018: Initial benchmarks collected, proof-of-concept tools integrated, potential participants contacted
- April 2018: Proposal accepted at Jury meeting at TACAS
- July 2018: Announced at CAV

Java Track at SV-COMP

Objectives:

- More languages in SV-COMP
- Standard benchmark set
- Comparability
- Reproducibility
- Re-use existing benchmarking infrastructure

Tool	JPF 32				JayHorn 5.1				JBMC 5.8			
Limits	timelimit: 180 s, memlimit: 15000 MB, CPU core limit: 8											
Host	localhost											
OS	Linux 4.4.0-116-generic x86_64											
System	CPU: Intel Core i7-6700HQ CPU @ 2.60GHz with 8 cores, frequency: 3500 MHz, Turbo Boost enabled: RAM: 16014056 kB											
Date of execution	2018-03-10 14:09:57 GMT				2018-03-10 11:39:56 GMT				2018-03-10 16:54:00 GMT			
Run set	jpf.sv-comp18				jayhorn.sv-comp18				jbmc.sv-comp18			
.../git-sv-benchmarks/java/	status	cputime	walltime	memUsage	status	cputime	walltime	memUsage	status	cputime	walltime	memUsage
jbmc-regression/ArithmeticException1_false-assert.jar	false(reach)	1.252s	0.492s	86929408	true	3.748s	1.352s	203776000	false(reach)	0.207s	0.224s	17719296
jbmc-regression/ArithmeticException5_true-assert.jar	false(reach)	1.208s	0.479s	85815296	true	3.564s	1.169s	184410112	true	0.212s	0.226s	12709888
jbmc-regression/ArithmeticException6_false-assert.jar	true	1.172s	0.452s	82345984	true	3.448s	1.164s	174071808	false(reach)	0.251s	0.264s	17477632
jbmc-regression/ArrayIndexOutOfBoundsException1_false-assert.jar	true	1.164s	0.475s	83734528	false(reach)	6.516s	1.934s	228200448	false(reach)	0.256s	0.268s	16728064
jbmc-regression/ArrayIndexOutOfBoundsException2_false-assert.jar	true	1.140s	0.443s	81854464	false(reach)	6.748s	2.030s	230465536	false(reach)	0.250s	0.257s	18628608
jbmc-regression/ArrayIndexOutOfBoundsException3_false-assert.jar	true	1.256s	0.501s	82614288	false(reach)	7.204s	2.023s	233783296	false(reach)	0.223s	0.236s	18378752
jbmc-regression/BufferedReaderReadLine_false-assert.jar	true	1.200s	0.464s	85073920	false(reach)	11.484s	2.992s	264642560	false(reach)	0.373s	0.381s	29315072
jbmc-regression/CharSequenceBug_false-assert.jar	true	1.136s	0.444s	79372288	false(reach)	7.336s	2.191s	233238528	false(reach)	1.407s	1.422s	26890240
jbmc-regression/CharSequenceToString_true-assert.jar	true	1.136s	0.501s	80220160	false(reach)	7.412s	2.186s	235905024	true	1.047s	1.059s	36552704
jbmc-regression/ClassCastException1_false-assert.jar	false(reach)	1.172s	0.462s	86487040	false(reach)	6.124s	1.898s	230752256	false(reach)	0.305s	0.319s	18665472
jbmc-regression/ClassCastException2_true-assert.jar	true	1.216s	0.506s	77561856	true	6.512s	1.907s	233218048	true	0.233s	0.253s	13275136
jbmc-regression/ClassCastException3_false-assert.jar	false(reach)	1.148s	0.441s	84291584	false(reach)	6.852s	2.080s	233496576	false(reach)	0.235s	0.242s	17874944
jbmc-regression/Class_method1_true-assert.jar	true	1.160s	0.457s	80973824	true	5.148s	1.699s	236470272	true	0.236s	0.256s	13553664
jbmc-regression/Inheritance1_true-assert.jar	true	1.180s	0.482s	83697664	true	7.240s	2.046s	230694912	true	0.269s	0.276s	13492224
jbmc-regression/NegativeArraySizeException1_false-assert.jar	false(reach)	1.236s	0.492s	87400448	true	3.720s	1.230s	185139200	false(reach)	0.213s	0.225s	16687104
jbmc-regression/NegativeArraySizeException2_false-assert.jar	false(reach)	1.176s	0.472s	83542016	true	3.908s	1.245s	185753600	false(reach)	0.241s	0.247s	18546688
jbmc-regression/NullPointerException1_false-assert.jar	false(reach)	1.152s	0.440s	84705280	unknown	1.864s	0.686s	136036352	false(reach)	0.243s	0.245s	16834560
jbmc-regression/NullPointerException2_false-assert.jar	false(reach)	1.212s	0.466s	85893120	unknown	1.740s	0.676s	134901760	false(reach)	0.244s	0.251s	16654336
jbmc-regression/NullPointerException3_false-assert.jar	false(reach)	1.368s	0.532s	88522752	unknown	1.668s	0.616s	133578752	false(reach)	0.243s	0.255s	16982016
jbmc-regression/NullPointerException4_false-assert.jar	false(reach)	1.272s	0.535s	77479936	unknown	1.636s	0.626s	128970752	false(reach)	0.240s	0.248s	18771968
jbmc-regression/RegexMatches01_true-assert.jar	true	1.308s	0.537s	80244736	false(reach)	8.344s	2.393s	246411264	true	1.280s	1.301s	39464960
jbmc-regression/RegexMatches02_false-assert.jar	true	1.192s	0.481s	85598208	false(reach)	9.824s	2.621s	254140416	false(reach)	1.033s	1.044s	64978944
jbmc-regression/RegexSubstitution01_true-assert.jar	true	1.252s	0.503s	83034112	false(reach)	14.268s	3.535s	394878976	true	72.774s	72.858s	757309440
jbmc-regression/RegexSubstitution02_false-assert.jar	true	1.116s	0.442s	81616896	false(reach)	17.136s	4.284s	390799360	true	40.973s	41.011s	526553088
jbmc-regression/RegexSubstitution03_true-assert.jar	true	1.232s	0.465s	86216704	false(reach)	12.680s	3.158s	338685952	true	1.754s	1.766s	46804992
jbmc-regression/StaticCharMethods01_true-assert.jar	true	1.176s	0.502s	82903040	false(reach)	6.744s	2.041s	228978496	true	0.261s	0.276s	14376960
jbmc-regression/StaticCharMethods02_false-assert.jar	true	1.196s	0.484s	83034112	false(reach)	8.524s	2.476s	237215744	false(reach)	0.829s	0.854s	50974720
jbmc-regression/StaticCharMethods03_false-assert.jar	true	1.192s	0.486s	82206720	false(reach)	10.188s	2.888s	243417088	false(reach)	0.847s	0.858s	50806784
jbmc-regression/StaticCharMethods04_false-assert.jar	true	1.196s	0.487s	84496384	false(reach)	8.664s	2.529s	231280640	false(reach)	0.821s	0.829s	51412992

Current status:

- March 2018: Initial benchmarks collected, proof-of-concept tools integrated, potential participants contacted
- April 2018: Proposal accepted at Jury meeting at TACAS
- July 2018: Announced at CAV
- Since September: detailed discussions with 4 registered participants (JPF, SPF, JayHorn, JBMC)

Java Track at SV-COMP

Tool	JPF 32				JayHorn 5.1				JBMC 5.8			
Limits	timelimit: 180 s, memlimit: 15000 MB, CPU core limit: 8											
Host	localhost											
OS	Linux 4.4.0-116-generic x86_64											
System	CPU: Intel Core i7-6700HQ CPU @ 2.60GHz with 8 cores, frequency: 3500 MHz, Turbo Boost enabled: RAM: 16014056 kB											
Date of execution	2018-03-10 14:09:57 GMT				2018-03-10 11:39:56 GMT				2018-03-10 16:54:00 GMT			
Run set	jpf.sv-comp18				jayhorn.sv-comp18				jbmc.sv-comp18			
.../git-sv-benchmarks/java/	status	cputime	walltime	memUsage	status	cputime	walltime	memUsage	status	cputime	walltime	memUsage
jbmc-regression/ArithmeticException1_false-assert.jar	false(reach)	1.252s	0.492s	86929408	true	3.748s	1.352s	203776000	false(reach)	0.207s	0.224s	17719296
jbmc-regression/ArithmeticException5_true-assert.jar	false(reach)	1.208s	0.479s	85815296	true	3.564s	1.169s	184410112	true	0.212s	0.226s	12709888
jbmc-regression/ArithmeticException6_false-assert.jar	true	1.172s	0.452s	82345984	true	3.448s	1.164s	174071808	false(reach)	0.251s	0.264s	17477632
jbmc-regression/ArrayIndexOutOfBoundsException1_false-assert.jar	true	1.164s	0.475s	83734528	false(reach)	6.516s	1.934s	228200448	false(reach)	0.256s	0.268s	16728064
jbmc-regression/ArrayIndexOutOfBoundsException2_false-assert.jar	true	1.140s	0.443s	81854464	false(reach)	6.748s	2.030s	230465536	false(reach)	0.250s	0.257s	18628608
jbmc-regression/ArrayIndexOutOfBoundsException3_false-assert.jar	true	1.256s	0.501s	82014208	false(reach)	7.204s	2.023s	233783296	false(reach)	0.223s	0.236s	18378752
jbmc-regression/BufferedReaderReadLine_false-assert.jar	true	1.200s	0.464s	85073920	false(reach)	11.484s	2.992s	264642560	false(reach)	0.373s	0.381s	29315072
jbmc-regression/CharSequenceBug_false-assert.jar	true	1.136s	0.444s	79372288	false(reach)	7.336s	2.191s	233238528	false(reach)	1.407s	1.422s	26890240
jbmc-regression/CharSequenceToString_true-assert.jar	true	1.136s	0.501s	80220160	false(reach)	7.412s	2.186s	235905024	true	1.047s	1.059s	36552704
jbmc-regression/ClassCastException1_false-assert.jar	false(reach)	1.172s	0.462s	86487040	false(reach)	6.124s	1.898s	230752256	false(reach)	0.305s	0.319s	18665472
jbmc-regression/ClassCastException2_true-assert.jar	true	1.216s	0.506s	77561856	true	6.512s	1.907s	233218048	true	0.233s	0.253s	13275136
jbmc-regression/ClassCastException3_false-assert.jar	false(reach)	1.148s	0.441s	84291584	false(reach)	6.852s	2.080s	233496576	false(reach)	0.235s	0.242s	17874944
jbmc-regression/Class_method1_true-assert.jar	true	1.168s	0.457s	80973824	true	5.148s	1.699s	236470272	true	0.236s	0.256s	13553664
jbmc-regression/Inheritance1_true-assert.jar	true	1.188s	0.482s	83697664	true	7.240s	2.046s	230694912	true	0.269s	0.276s	13492224
jbmc-regression/NegativeArraySizeException1_false-assert.jar	false(reach)	1.236s	0.492s	87400448	true	3.720s	1.230s	185139200	false(reach)	0.213s	0.225s	16687104
jbmc-regression/NegativeArraySizeException2_false-assert.jar	false(reach)	1.176s	0.472s	83542016	true	3.908s	1.245s	185753600	false(reach)	0.241s	0.247s	18546688
jbmc-regression/NullPointerException1_false-assert.jar	false(reach)	1.152s	0.440s	84705280	unknown	1.864s	0.686s	136036352	false(reach)	0.243s	0.245s	16834560
jbmc-regression/NullPointerException2_false-assert.jar	false(reach)	1.212s	0.466s	85893120	unknown	1.740s	0.676s	134901760	false(reach)	0.244s	0.251s	16654336
jbmc-regression/NullPointerException3_false-assert.jar	false(reach)	1.368s	0.532s	88522752	unknown	1.668s	0.616s	133578752	false(reach)	0.243s	0.255s	16982016
jbmc-regression/NullPointerException4_false-assert.jar	false(reach)	1.272s	0.535s	77479936	unknown	1.636s	0.626s	128970752	false(reach)	0.240s	0.248s	18771968
jbmc-regression/RegexMatches01_true-assert.jar	true	1.308s	0.537s	80244736	false(reach)	8.344s	2.393s	246411264	true	1.280s	1.301s	39464960
jbmc-regression/RegexMatches02_false-assert.jar	true	1.192s	0.481s	85598208	false(reach)	9.824s	2.621s	254140416	false(reach)	1.033s	1.044s	64978944
jbmc-regression/RegexSubstitution01_true-assert.jar	true	1.252s	0.503s	83034112	false(reach)	14.268s	3.535s	394878976	true	72.774s	72.858s	757309440
jbmc-regression/RegexSubstitution02_false-assert.jar	true	1.116s	0.442s	81616896	false(reach)	17.136s	4.284s	390799360	true	40.973s	41.011s	526553088
jbmc-regression/RegexSubstitution03_true-assert.jar	true	1.232s	0.465s	86216704	false(reach)	12.680s	3.158s	338685952	true	1.754s	1.766s	46804992
jbmc-regression/StaticCharMethods01_true-assert.jar	true	1.176s	0.502s	82903040	false(reach)	6.744s	2.041s	228978496	true	0.261s	0.276s	14376960
jbmc-regression/StaticCharMethods02_false-assert.jar	true	1.196s	0.484s	83034112	false(reach)	8.524s	2.476s	237215744	false(reach)	0.829s	0.854s	50974720
jbmc-regression/StaticCharMethods03_false-assert.jar	true	1.192s	0.486s	82200720	false(reach)	10.188s	2.888s	243417088	false(reach)	0.847s	0.858s	50806784
jbmc-regression/StaticCharMethods04_false-assert.jar	true	1.196s	0.487s	84496384	false(reach)	8.664s	2.529s	231280640	false(reach)	0.821s	0.829s	51412992

Objectives:

- More languages in SV-COMP
- Standard benchmark set
- Comparability
- Reproducibility
- Re-use existing benchmarking infrastructure

Current status:

- March 2018: Initial benchmarks collected, proof-of-concept tools integrated, potential participants contacted
- April 2018: Proposal accepted at Jury meeting at TACAS
- July 2018: Announced at CAV
- Since September: detailed discussions with 4 registered participants (JPF, SPF, JayHorn, JBMC)
- Planned start of competition runs: 20 November

Benchmarks and Properties

368 benchmarks (40LOC on average, 250LOC max)

- jayhorn-recursive, jbmc-regression, jpf-regression, MinePump

Java 1.8

Calls to Java standard library (`java.*`, `javax.*`) allowed;
sources of other dependencies must be part of the benchmark.

1 category for violation of `asserts` ("ReachSafety")
Property: `CHECK(init(Main.main), LTL(G assert))`

Benchmarks and Properties

java/jbmc-regression/StringStartEnd02/Main.java

java/jbmc-regression/StringStartEnd02.yml

```
import org.sosy_lab.sv_benchmarks.Verifier;

public class Main {
  public static void main(String[] args) {
    String[] strings = new String[4];
    strings[0] = Verifier.nondetString();
    strings[1] = Verifier.nondetString();
    strings[2] = Verifier.nondetString();
    strings[3] = Verifier.nondetString();

    int i = 0;
    for (String string : strings) {
      if (string.startsWith("te"))
        ++i;
    }
    assert i == 1;
  }
}
```

```
format_version: "0.1"
input_files:
  - ../common/
  - StringStartEnd02/
properties:
  - property_file: ../properties/assert.prp
    expected_verdict: false
```

If a tool requires class files as input it is responsible for compiling the benchmark.

A benchmark must be compilable by passing all `.java` files within the directories listed in `input_files` to `javac`.

Rules for Nondeterminism

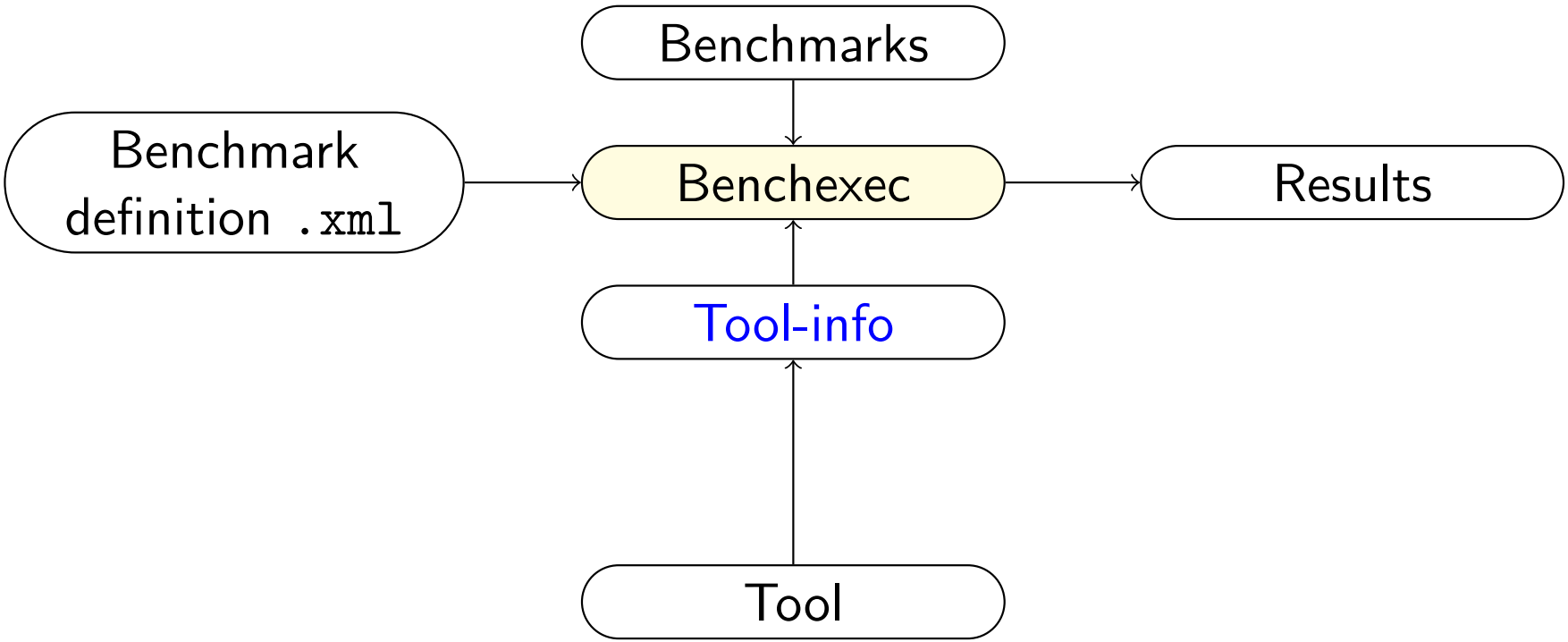
Only source of nondeterminism: return values of methods defined in `org.sosy_lab.sv_benchmarks.Verifier` class.

Must not be used:

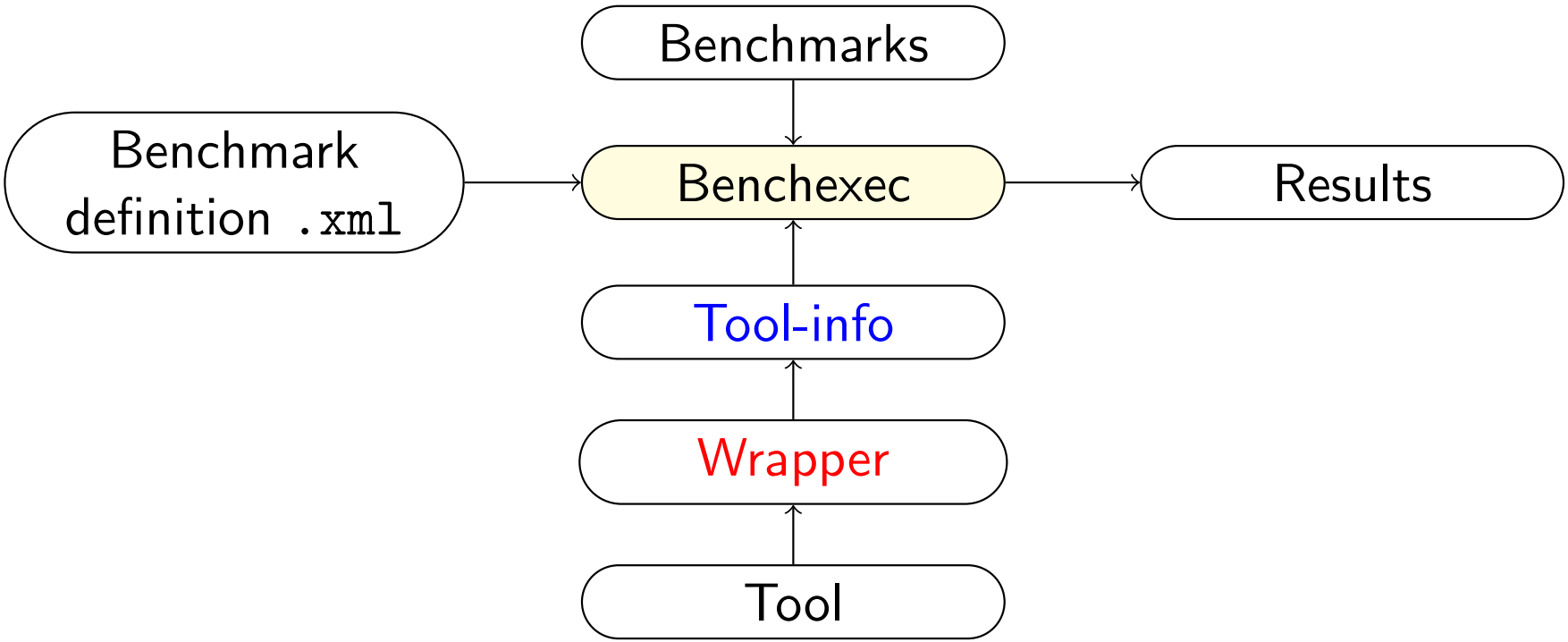
- Arguments of `main`
- Library methods that make system calls

```
package org.sosy_lab.sv_benchmarks;

import java.util.Random;

public final class Verifier
{
  public static void assume(boolean condition)
  {
    if(!condition) {
      Runtime.getRuntime().halt(1);
    }
  }

  public static boolean nondetBoolean()
  {
    return new Random().nextBoolean();
  }

  public static byte nondetByte()
  {
    return (byte)(new Random().nextInt());
  }

...
```

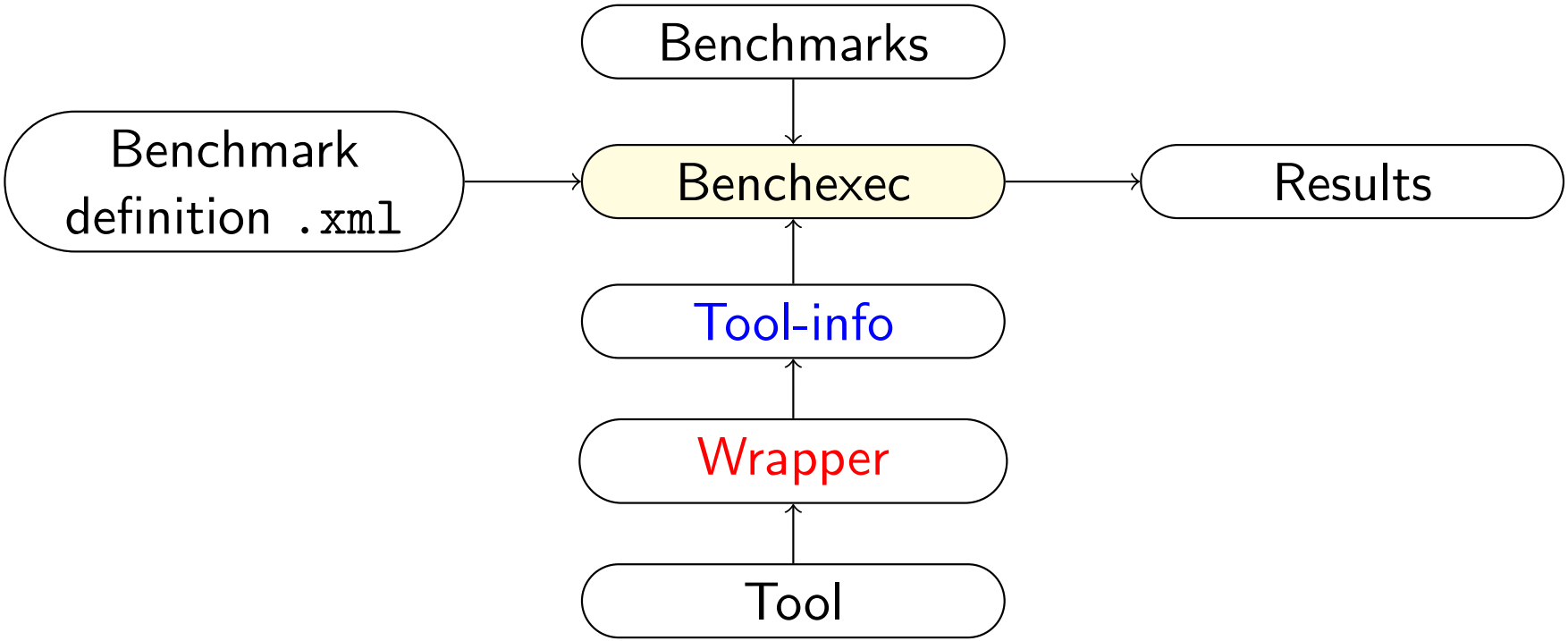
Tool-info and Wrapper Scripts



Tool-info and Wrapper Scripts



Tool-info and Wrapper Scripts



E.g. command line produced for JPF:

```
./jpf-sv-comp
  --graphml-witness witness.graphml
  --propertyfile ../sv-benchmarks/java/properties/assert.prp
  ../sv-benchmarks/java/common/org/sosy_lab/sv_benchmarks/Verifier.java
  ../sv-benchmarks/java/jbmc-regression/StringStartEnd02/Main.java
```

How can I use it?

Re-use existing benchmarking infrastructure

- Stop writing benchmarking scripts
- Use it for running your tests

'Standard' benchmark set

- Use it for running comparisons
- Contribute your benchmarks

Compare with the 'best' configuration of a tool

- Take the most recent competition candidate:
  Download from
  `https://sv-comp.sosy-lab.org/2018/systems.php`

How can I use it?

Reproduce the competition results:

- Download:

```
git clone https://github.com/sosy-lab/sv-benchmarks
git clone https://github.com/sosy-lab/benchexec
git clone https://github.com/sosy-lab/sv-comp
git clone https://gitlab.com/sosy-lab/sv-comp/archives-2019
```

- Run:

```
cd benchexec
for tool in jpf spf jayhorn jbmc
do
  unzip ../archives-2019/2019/$tool.zip; mv $tool/* .
  bin/benchexec ../sv-comp/benchmark-defs/$tool.xml
done
bin/table-generator results/*.xml.bz2
```

Currently (5 Nov 2018) only runs with benchexec's `yaml` branch.

How can I use it?

Run JPF on your own benchmarks:

- Add descriptor `yml` file for each benchmark
- List descriptor files in `MyBenchmarks.set`
- Add `MyBenchmarks.set` to benchmark definition `jpf.xml`:

```
<tasks name="MyBench">
  <includesfile>
    ../sv-benchmarks/java/MyBenchmarks.set</includesfile>
  <propertyfile>
    ../sv-benchmarks/java/properties/assert.prp</propertyfile>
</tasks>
```

- Run with `-t MyBench`

```
bin/benchexec ../sv-comp/benchmark-defs/jpf.xml -t MyBench
bin/table-generator results/*.xml.bz2
```

Outlook

What is needed for 2020:

- More benchmarks:
  Fork `https://github.com/sosy-lab/sv-benchmarks` and create PR with your benchmarks

Outlook

What is needed for 2020:

- More benchmarks:
  Fork `https://github.com/sosy-lab/sv-benchmarks` and create PR with your benchmarks
- Witness validators

Witness Validation for Java

Refutation witnesses (for ReachSafety property):

- Witness contains counterexample trace
  annotated with evaluated assignments and conditionals
  $\rightarrow$ Check whether counterexample trace is feasible and violates the property
- Proposed implementation: generate harness, compile and execute

Correctness witnesses (for ReachSafety property):

- Witness contains dynamic CFG annotated with invariants
  $\rightarrow$ Check whether invariants hold and imply properties
- ???

Outlook

What is needed for 2020:

- More benchmarks:
  Fork `https://github.com/sosy-lab/sv-benchmarks` and create PR with your benchmarks
- Witness validators

Outlook

What is needed for 2020:

- More benchmarks:
  Fork `https://github.com/sosy-lab/sv-benchmarks` and create PR with your benchmarks
- Witness validators
- Encourage participation of more tools and tool variants

Subscribe to `sv-comp@googlegroups.com`

Up-to-date version of paper: `http://arxiv.org/abs/1809.03739`

Outlook

What is needed for 2020:

- More benchmarks:
  Fork `https://github.com/sosy-lab/sv-benchmarks` and create PR with your benchmarks
- Witness validators
- Encourage participation of more tools and tool variants

Subscribe to `sv-comp@googlegroups.com`

Up-to-date version of paper: `http://arxiv.org/abs/1809.03739`
