Unit Testing Tool Competition — Round Four

Urko Rueda[+], René Just*, Juan P. Galeotti[‡], and Tanja E. J. Vos[+,♯]

[+]Universidad Politécnica
Valencia, ES
{urueda, tvos}@dsic.upv.es

[♯]Open Universiteit
Heerlen, NL
{tanja.vos}@ou.nl

*University of Massachusetts
Amherst, MA, USA
rjust@cs.umass.edu

[‡]University of Buenos Aires
Buenos Aires, Argentina
jgaleotti@dc.uba.ar

ABSTRACT

This paper describes the methodology and results of the 4th edition of the Java Unit Testing Tool Competition. This year's competition features a number of infrastructure improvements, new test effectiveness metrics, and the evaluation of the test generation tools for multiple time budgets. Overall, the competition evaluated four automated test generation tools. This paper details the methodology and contains the full results of the competition.

Keywords

tool competition, benchmark, mutation testing, automated unit testing, Java, Defects4J

1. INTRODUCTION

The objective of all four editions of the Java Unit Testing Tool Competition has been to evaluate tools that generate JUnit tests for Java classes. To that end, the competition first defines a benchmark of Java classes, selected from open source projects, and a scoring formula that takes into account the effectiveness (i.e., code coverage and fault finding capability) of the generated tests. Then, the competition executes the participating tools on the benchmark and ranks them according to the scoring formula.

Previous editions of the tool competition showed that the evaluation of tools in the form of a competition provides great feedback to the tool developers, which helps them to improve their tools and guides future development efforts. The rules of all competitions have been the same: the participants neither know the benchmark for which they need to generate tests beforehand nor do they know the exact scoring formula that is used to evaluate the generated tests—this avoids fine-tuning towards specific artefacts.

This year's edition of the tool competition evaluated 4 tools—3 tools from participating developers (EvoSuite, jTexPert, and T3) and 1 baseline tool (Randoop). Furthermore, this year's competition differs from previous editions [10] in the following four ways.

SBST16, May 16-17, 2016, Austin, TX, USA

ACM ISBN 978-1-4503-4166-0/16/05.

DOI: http://dx.doi.org/10.1145/2897010.2897018

Benchmark subjects.

While the benchmarks for the 2nd and the 3rd competition were the same, this year's competition uses a completely new benchmark. We selected all Java classes for the benchmark from the Defects4J data set [4]. Section 2 details the selection procedure and the selected benchmark subjects.

Benchmark infrastructure.

We improved the benchmark infrastructure by integrating parts of the Defects4J's test execution framework, which allowed us to include new test effectiveness metrics in the evaluation and scoring formula. In particular, the improvements made it possible to not only consider code coverage ratios and mutation scores but also fault finding capabilities for real faults, which come with the Defects4J data set. Section 4 details the benchmark infrastructure and the competition methodology.

Time budgets.

This year's competition evaluated the participating tools for different time budgets (1, 2, 4, and 8 minutes per benchmark subject). The idea is that in practice, one cannot spend infinite time on deriving tests. Moreover, controlling for the test generation time provides insights into the effectiveness of the tools for different time budgets and how the effectiveness of the generated tests increases for an increasing time budget. Section 4.2 details the test generation for different time budgets.

Flaky tests.

In contrast to previous editions, this year's competition detected and penalized the generation of flaky tests (i.e., a test that does not reliably pass when executed multiple times on the same program version) and uncompilable test classes. Moreover, the improved benchmark infrastructure automatically removed flaky tests to allow a reliable computation of the test effectiveness metrics. Section 4.4 details the procedure of detecting and removing flaky tests.

2. THE BENCHMARK SUBJECTS

This tool competition exclusively used benchmark subjects from the Defects4J data set [4], which contains 357 real faults from 5 open source projects. For each real fault, Defects4J provides a buggy and a fixed program version and a developer-written test suite of which at least one test case exposes the fault.

We arbitrarily selected 80 out of Defects4J's 357 real faults, uniformly distributed across the 5 open source projects.

Table 1: Configuration options for Randoop.

Option	Value
`clear`	10000
`string-maxlen`	5000
`forbid-null`	false
`null-ratio`	0.1
`no-error-revealing-tests`	true
`omitmethods`	random
`silently-ignore-bad-class-names`	true
`testsperfile`	100
`ignore-flaky-tests`	true
`only-test-public-members`	true

We discarded 12 faults[1] for which we couldn't compute all results, leaving a total of 68 faults—the benchmark subjects. In particular, we used the following Defects4J subjects for the competition (cf., [4]):

- Chart-{1, 2, 3, 4, 6, 7, 9, 11, 12, 16, 17, 20, 23, 24, 25, 26}
- Closure-{14, 16, 20, 46, 68, 74, 98, 99, 100, 124, 130, 132}
- Lang-{28, 33, 36, 37, 41, 43, 47, 50, 57, 58, 59, 60, 63, 65}
- Math-{2, 7, 18, 20, 21, 39, 44, 52, 56, 64, 67, 88, 91, 93, 103, 106}
- Time-{3, 4, 5, 7, 8, 10, 11, 13, 20, 23}

For each selected subject, the difference between the buggy and the fixed program version is only the bug fix that applies to a single defective class—the class under test (CUT). Note that all tools generated tests for the fixed version of the CUT as all participating tools generate regression test suites.

3. BASELINE AND PARTICIPANTS

Similar to the 3rd tool competition [10], we used the random test generation tool Randoop as a baseline. We contacted the Randoop developers to discuss the best configuration for the competition. Table 1 lists the selected options and their values. Note that Randoop requires the user to provide, in addition to the target class, all dependency classes that Randoop needs to exercise in order to create tests for the target class. We therefore configured the corresponding option (`classlist`) on a per-CUT basis. Defects4J provides, as one of the available artifacts, the list of fault-related classes for each fault. We used this list to configure the `classlist` option of Randoop. The rationale behind this decision is twofold. First, Randoop is not a participating tool but rather a baseline. Second, according to the Randoop developers, some tuning is expected each time Randoop is executed on a CUT—providing the list of dependencies simulates such tuning. Note that we did not use two features of Randoop (package awareness and observer methods), which would have increased the effectiveness [6] of the generated tests. Randoop failed to generate package-aware tests for the benchmark subjects due to a bug—a corresponding bug fix became available after the competition had started. We did not provide observer methods as this would have required us to run an external purity analysis.

In addition to Randoop, we employed and evaluated the developer-written test suites that are provided by Defects4J

[1]Closure-{25, 29, 35, 53}, Lang-{30, 62}, and Time-{6, 14, 15, 16, 21, 24}

Table 2: Summary of participating tools.

Tool	Technique	Static analysis
EvoSuite [2]	evolutionary algorithm	yes
jTexPert [11]	guided random testing	yes
T3 [8, 9]	random testing	no
Randoop (baseline) [7]	random testing	no

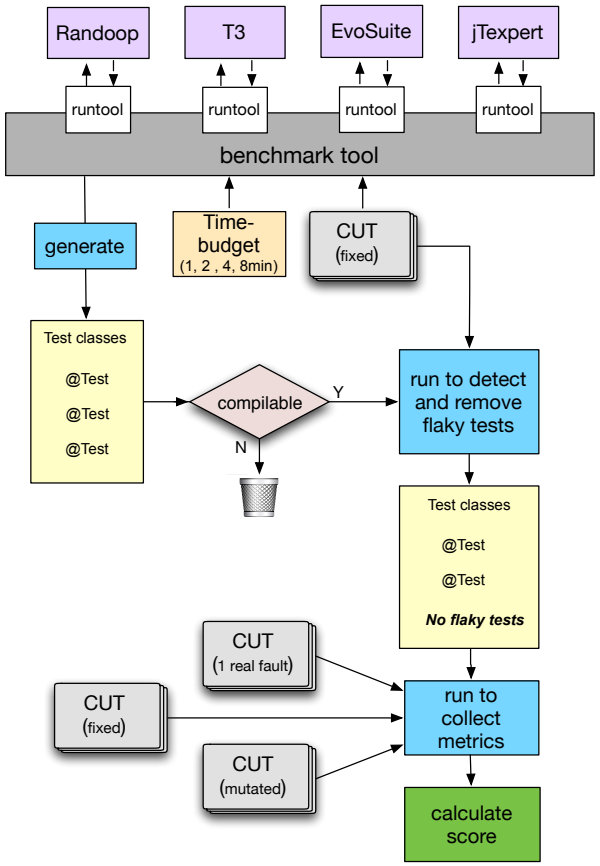

Figure 1: Overview of the competition methodology.

for each real fault. While these test suites provide realistic scores for the effectiveness of developer-written test suites, they are not comparable with the test suites generated by the participating tools: the developer-written test suites have evolved over years, and it is not possible for us to estimate the amount of human effort that was required to produce them. We were lacking enough resources to write additional test suites (and time ourselves) with enough diversity (programmer skills, backgrounds, knowledge of the selected CUTs, etc.).

In contrast to Randoop, EvoSuite, T3 and jTexPert are participating tools, which means that their developers installed and configured the tool for the competition. Table 2 provides a summary of the participating tools.

4. METHODOLOGY

Figure 1 visualizes the overall methodology that we applied in this tool competition. As in previous years, each participant had to implement a wrapper, named `runtool`, for

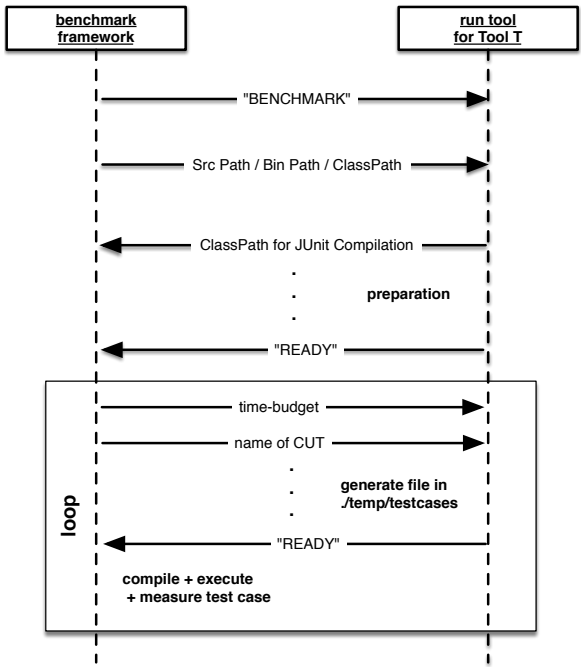

Figure 2: Protocol implemented by `runtool`.

their test generation tool. This wrapper represents the interface between the benchmark infrastructure and the test generation tool, and implements the simple communication protocol described in Figure 2. In contrast to previous years [1], this years' protocol implements a time budget for the test generation. This allows the evaluation of the test generation tools when controlling for generation time. We selected the following time budgets: 1, 2, 4, and 8 minutes.

Due to the large number of necessary tool and test suite executions, we parallelized the infrastructure and distributed the work over 32 virtual machines, running on two HP Z820 workstations with 20 cores and 256Gb of memory each. Additionally, we decoupled the test generation from the test execution and metric computation.

4.1 Installation and parameter tuning

Prior to running the competition, we provided a set of test subjects, also selected from the Defects4J data set. The participants were able to use these test subjects to test the correct implementation of the protocol and to tune their tool's parameters. Note that the test subjects and the benchmark subjects are non-overlapping sets.

4.2 Test generation

For each tool and time budget, we generated 6 test suites to account for the randomness in the tools. Each tool was executed on a virtual machine with 1 CPU and 8Gb or memory. Note that the virtual machines were configured with a single CPU to control for implementation differences of the tools (e.g., multi-threading support), which could affect the results. For each time budget, a tool was allowed to run at most twice as long as the given time budget. The infrastructure terminated a tool if it didn't finish within this time frame. The scoring formula (Section 4.6) applies a penalty, which is inversely proportional to the excess ratio of the time budget—e.g., a tool that takes twice as long to generate a test suite only gets half the points for the test effectiveness.

4.3 Test execution

We executed each test suites on a virtual machine with 2 CPUs and 16Gb of memory to compute the test effectiveness metrics detailed in Section 4.5. Compared to the test generation, the test execution and metric computation is tool-independent. We therefore doubled the specifications of the virtual machines to efficiently compute the metrics. The execution of the entire competition (test generation and execution) took over a week on the 32 virtual machines, which roughly corresponds to 8 CPU months. After we obtained and aggregated the data from the 32 virtual machines, we made the results available to all participants.

4.4 Flaky tests

A test generation tool may generate flaky tests. A flaky test is a test that doesn't reliably pass when executed on the same program version. For example, a test that asserts on the system time only passes during generation and fails on every later execution. Moreover, an order-dependent test might fail if the dependencies aren't met. Flaky tests are harmful because they might lead to false-positive warnings. Recall that all tools generated regression test suites for the fixed program version, which means that every test suite should pass on the program version for which it was generated. We validated this assumption and removed flaky tests if necessary to obtain reliable fault-detection and mutation analysis results.

We employed Defects4J's infrastructure to remove flaky tests, which is a fully automated process that works as follows. It first removes all non-compiling test classes. Then, it executes each compilable test class five times. If a test fails during any of these five executions, it is considered flaky and removed, and the test suite is recompiled and reexecuted. This loop is repeated until all tests pass five times in a row.

The overall outcome of this process was 1) a test suite that reliably passes on the version it was generated for, used to compute the effectiveness metrics (see Section 4.5), and 2) the numbers of uncompilable test classes and flaky tests, used as a penalty in the scoring formula (see Section 4.6).

4.5 Computed metrics

We integrated Defects4J's test execution and analysis component into the benchmark infrastructure to compute three test effectiveness metrics (code coverage ratio, mutation score, and real fault detection score) for each test suite.

Code coverage.

For each test suite, the benchmark infrastructure computes two code coverage ratios (statement coverage and condition coverage) on the fixed program version. It uses the Defects4J's code coverage component, which is based on Cobertura[2].

Mutation score.

For each test suite, the benchmark infrastructure computes the mutation score on the fixed program version. It uses the Defects4J's mutation analysis component, which is based on the Major mutation framework [3, 5]. The benchmark infrastructure uses all mutation operators available in Major.

[2]https://github.com/cobertura/cobertura

Real-fault detection.

For each test suite, the benchmark infrastructure computes the real fault detection score, using the buggy program version. Given that each buggy program version contains exactly one real fault, the real fault detection score is either 0 or 1. Note that the score is independent of how many generated tests expose the real fault.

4.6 Scoring formula

To determine the overall score we augmented the scoring formula used in the 2015 competition [10]. In this edition, we executed each tool several times using different time budgets (i.e., 1, 2, 4, and 8 minutes). Given a tool T, a time limit L, a class under test C, we provide the following test effectiveness metrics for each execution r:

$$covScore_{\langle T,L,C,r\rangle} := w_i \cdot cov_i + w_b \cdot cov_b + w_m \cdot cov_m + (\text{real fault found ? } w_f : 0)$$

Similar to previous editions, we considered the achieved instruction coverage (cov_i), branch coverage (cov_b), and mutation kill score (i.e., the ratio of killed mutants cov_m). w_i, w_b and w_m are the weights, for which we chose the values $w_i = 1$, $w_b = 2$, and $w_m = 4$. As DEFECTS4J provides a real fault for each CUT, we also give an extra score (w_f) to a generated test suite if it is able to detect the real fault. A test suite detects a fault if it does not fail on the fixed version and at least one of its test cases fails on the buggy version. The fixed score for a test suite that detects the real fault is $w_f = 4$.

As in previous editions, we penalize a tool if the tool did not finish the test generation in time. For each time limit L, we allowed each tool to execute up to $2 \times L$ time before killing the tool generation process. The time score ($tScore$) is then computed following this formula:

$$tScore_{\langle T,L,C,r\rangle} := covScore_{\langle T,L,C,r\rangle} \cdot min\left(1, \frac{L}{genTime}\right)$$

where T is the time budget for the execution r, and $genTime$ is the total generation time for the execution r of the tool T (observe this is a value between 0 and $2 \times L$). We chose to give less score to a tool if it does not meet the specified time limit. For example, if the tool needs to be externally killed if the time threshold of $2 \times L$ is exhausted, then we it will score at most half of the coverage score (as $L/(2 \times L) = 1/2$).

Each generated test suite might also include uncompilable classes and flaky (i.e. unstable) test cases. The penalty for uncompilable classes and flaky tests is defined as follows:

$$penalty_{\langle T,L,C,r\rangle} := \begin{cases} 2 & \text{if no compilable test classes} \\ \frac{\#uClasses}{\#Classes} + \frac{\#fTests}{\#Tests} & \text{otherwise} \end{cases}$$

where $\#uClasses$ and $\#Classes$ are the number of uncompilable generated test classes and total generated test classes respectively, and $\#fTests$ and $\#Tests$ are the number of flaky test cases and total test cases respectively. We chose to penalise the generation of uncompilable test classes since they might decrease the usability of the tool. Also, as flaky tests are not reliable and need to be removed from the test suite, we also penalize a tool that generates a test case that might fail when it is executed again.

Finally, the scoring formula for a tool T at a given execution r with time limit L and a class under test C is:

$$score_{\langle T,L,C,r\rangle} := tScore_{\langle T,L,CUT,r\rangle} - penalty_{\langle T,L,CUT,r\rangle}$$

Given the non-determinism in all the considered test generation tools (including RANDOOP), we executed each tool six times. The score for a given time budget L and class under test C is the average of all the executions for the same tool T, budget L and class under test C:

$$score_{\langle T,L,C\rangle} := avg\big(Score_{\langle T,L,C,r\rangle}\big) \text{ for all } r \text{ executions}$$

The *final* score for a tool T is the sum of all scores for all classes under test and time budgets used in the competition:

$$score_T := \sum_{L,C} Score_{\langle T,L,C\rangle}$$

4.7 Threats to Validity

Conclusion validity.

Reliability of treatment implementation: In order to reduce the threat of treating the tools differently, a clear protocol was stablished. We gave the same instructions to all participants (developers of testing tools that will be evaluated) of the unit testing tool competition.

Reliability of measures: As unreliable measures can invalidate our competition, we tried to be as objective as possible for measuring the test efficiency and effectiveness. For example, all timing information was measured in wall clock time using Java's `System.currentTimeMillis()` method. Effectiveness measures were automatically measured by DEFECTS4J. Finally, as the participants' tools are based on non-deterministic algorithms, it was necessary to run the benchmark several times in order to obtain an average value for the measured variables. However, due to time and resource restrictions we could only run each tool a maximum of six times.

Internal validity.

Our study could be affected negatively if CUTs were not selected appropriately. To mitigate this, CUTs were arbitrarily selected out of DEFECTS4J's real faults, uniformly distributed across the 5 open source projects. Other artifacts used for this study were the benchmark infrastructure and the `runtool` wrappers that were specifically developed for this competition. To improve confidence in the benchmark infrastructure, it was extensively tested by its developers prior to the competition. Regarding the `runtool` wrappers, implemented by each participant, these were also tested while tuning the tools' parameters. The test benchmark used for parameter tuning was not part of the competition benchmark.

Construct validity.

In our study, the final score has been calculated based on a scoring formula whose weights were assigned in accordance with those quality indicators that are considered most important. A good test suite detects real faults but the set of all faults in a program is unknowable. Consequently, proxy metrics such as code coverage ratio or mutation score are commonly used. The scoring formula considers for each test suite its real fault detection rate for one known real fault and additionally its code coverage ratio and mutation score. This combined score accounts for the fact that a test suite might detect the one known real fault by chance, even if its overall code coverage ratio and mutation score is very low. We chose the weight for each test effectiveness metric based on experience gained through empirical studies that investi-

Table 3: Overall scores for all tools.

Tool	Budget	Score	Std.dev
EvoSuite	*	1127	136.96
T3	*	978	86.17
jTexPert	*	931	137.03
Randoop	*	747	40.31

Table 4: Scores for all time budgets.
(Optimal gives the maximum score and Developer the score achieved by the developer-written test suites).

Tool	Budget	Score	Std.dev
T3	1min	220	27.836
EvoSuite	1min	209	33.564
jTexPert	1min	179	38.811
Randoop	1min	155	11.500
EvoSuite	2min	259	45.679
T3	2min	241	30.649
jTexPert	2min	231	41.199
Randoop	2min	179	12.553
EvoSuite	4min	318	58.304
T3	4min	253	27.687
jTexPert	4min	251	48.000
Randoop	4min	197	16.254
EvoSuite	8min	341	57.720
jTexPert	8min	270	47.830
T3	8min	263	27.687
Randoop	8min	216	20.708
Optimal	–	748	–
Developer	–	611	–

gated the correlation of proxy metrics for test effectiveness and real fault detection [5].

5. RESULTS

Table 3 gives the overall scores for all participating tools and the baseline tool Randoop. Additionally, Table 4 shows the scores for each time budget. For comparison, this table also shows the optimal score (Optimal) and the score for the developer-written tests (Developer), for which no time budget applies and which were only executed once due to their deterministic nature. Table 5 details the test effectiveness metrics for the developer-written tests for each benchmark subject, where CUT gives the class under test, gen_t the generation time in seconds, cov_i the instruction coverage ratio, cov_b the branch coverage ratio, cov_m the mutation kill score, and F the number of detected real faults. The detailed results for each participating tool and time budget are listed in the appendix.

Acknowledgement

We would like to thank Giuliano Antoniol and the Randoop developers for supporting us with this tool competition. This work was partly funded by the PERTEST project (TIN2013-46928-C3-1-R).

Table 5: Results for the developer-written tests.

CUT	gen_t	cov_i	cov_b	cov_m	F
Chart-11	-	.41	.56	.17	1
Chart-12	-	.36	.27	.07	1
Chart-16	-	.75	.57	.51	1
Chart-17	-	.75	.76	.50	1
Chart-1	-	.53	.45	.19	1
Chart-20	-	.86	.87	.54	1
Chart-23	-	.52	.32	.06	1
Chart-24	-	.96	.90	.78	1
Chart-25	-	.70	.41	.02	1
Chart-26	-	.89	.69	.64	1
Chart-2	-	.74	.67	.53	1
Chart-3	-	.79	.73	.52	1
Chart-4	-	.71	.58	.28	1
Chart-6	-	.94	.93	.53	1
Chart-7	-	.88	.85	.42	1
Chart-9	-	.75	.76	.50	1
Closure-100	-	.97	.94	.70	1
Closure-124	-	1.00	.95	.86	1
Closure-130	-	.98	.92	.70	1
Closure-132	-	.98	.91	.79	1
Closure-14	-	.98	.93	.82	1
Closure-16	-	.98	.91	.71	1
Closure-20	-	.98	.91	.77	1
Closure-46	-	.97	.91	.70	1
Closure-68	-	.94	.88	.66	1
Closure-74	-	.91	.83	.63	1
Closure-98	-	.97	.93	.76	1
Closure-99	-	.97	.95	.68	1
Lang-28	-	1.00	1.00	.92	1
Lang-33	-	.95	.93	.73	1
Lang-36	-	.97	.87	.62	1
Lang-37	-	.99	.99	.84	1
Lang-41	-	.95	.93	.74	1
Lang-43	-	.82	.61	.42	1
Lang-47	-	.99	.98	.66	1
Lang-50	-	.71	.59	.55	1
Lang-57	-	1.00	.92	.78	1
Lang-58	-	.98	.90	.67	1
Lang-59	-	.99	.98	.66	1
Lang-60	-	.99	.98	.66	1
Lang-63	-	.93	.90	.61	1
Lang-65	-	.97	.90	.73	1
Math-103	-	.95	.88	.54	1
Math-106	-	.85	.78	.65	1
Math-18	-	.89	.82	.54	1
Math-20	-	.89	.83	.51	1
Math-21	-	.96	.92	.69	1
Math-2	-	.96	.96	.92	1
Math-39	-	1.00	1.00	.86	1
Math-44	-	.96	.95	.76	1
Math-52	-	1.00	.95	.89	1
Math-56	-	.88	.90	.77	1
Math-64	-	.98	.91	.79	1
Math-67	-	.75	.82	.61	1
Math-7	-	.96	.96	.71	1
Math-88	-	.92	.87	.77	1
Math-91	-	.96	.92	.70	1
Math-93	-	.92	.86	.73	1
Time-10	-	.97	.97	.73	1
Time-11	-	.81	.64	.47	1
Time-13	-	.87	.78	.79	1
Time-20	-	.77	.69	.54	1
Time-23	-	.93	.87	.66	1
Time-3	-	.99	.93	.64	1
Time-4	-	1.00	.98	.83	1
Time-5	-	1.00	1.00	.86	1
Time-7	-	.95	.81	.52	1
Time-8	-	.93	.87	.71	1

6. REFERENCES

[1] S. Bauersfeld, T. Vos, and K. Lakhotia. Unit testing tool competition – lessons learned. *Workshop on Future Internet Testing (FITTEST)*, 2013.

[2] G. Fraser and A. Arcuri. Evosuite: automatic test suite generation for object-oriented software. In *SIGSOFT FSE*, pages 416–419, 2011.

[3] R. Just. The Major mutation framework: Efficient and scalable mutation analysis for Java. In *Proceedings of the International Symposium on Software Testing and Analysis (ISSTA)*, pages 433–436, San Jose, CA, USA, July 23–25 2014.

[4] R. Just, D. Jalali, and M. D. Ernst. Defects4J: A database of existing faults to enable controlled testing studies for Java programs. In *Proceedings of the International Symposium on Software Testing and Analysis (ISSTA)*, pages 437–440, San Jose, CA, USA, July 23–25 2014.

[5] R. Just, D. Jalali, L. Inozemtseva, M. D. Ernst, R. Holmes, and G. Fraser. Are mutants a valid substitute for real faults in software testing? In *Proceedings of the Symposium on the Foundations of Software Engineering (FSE)*, pages 654–665, Hong Kong, November 18–20 2014.

[6] L. Ma, C. Artho, C. Zhang, H. Sato, M. Hagiya, Y. Tanabe, and M. Yamamoto. GRT at the SBST 2015 tool competition. In *Proceedings of the Eighth International Workshop on Search-Based Software Testing*, pages 48–51, 2015.

[7] C. Pacheco and M. D. Ernst. Randoop: feedback-directed random testing for java. In *Companion to the 22nd ACM SIGPLAN conference on Object-oriented programming systems and applications companion*, OOPSLA '07, pages 815–816, New York, NY, USA, 2007. ACM.

[8] I. Prasetya. T3, a combinator-based random testing tool for Java: Benchmarking. *Int. Workshop Future Internet Testing, Lecture Notes in Computer Science*, 8432, 2014.

[9] I. S. W. B. Prasetya, T. E. J. Vos, and A. Baars. Trace-based reflexive testing of OO programs with T2. In *1st Int. Conf. on Software Testing, Verification, and Validation (ICST)*, pages 151–160. IEEE, 2008.

[10] U. Rueda, T. E. J. Vos, and I. S. W. B. Prasetya. Unit Testing Tool Competition - Round Three. In G. Gay and G. Antoniol, editors, *8th IEEE/ACM International Workshop on Search-Based Software Testing, SBST 2015, Florence, Italy, May 18-19, 2015*, pages 19–24, 2015.

[11] A. Sakti, G. Pesant, and Y. Gueheneuc. Instance generator and problem representation to improve object oriented code coverage. To appear in IEEE Transactions on Software Engineering, 2015.

APPENDIX

Tables 6, 7, 8, and 9 give the detailed results for each tool and time budget. All numbers are averages across 6 runs for each triple <CUT, tool, time-budget>. The columns for each table are as follows:

- CUT: class under test (i.e., the benchmark subject).
- gen_t: generation time in seconds.
- cov_i: instruction coverage ratio.
- cov_b: branch coverage ratio.
- cov_m: mutation kill score.
- F: number of detected real faults.
- U: ratio of uncompilable test classes.
- B: ratio of broken (i.e., flaky) tests.

Table 6: Averaged results for 6 runs on 60 seconds time budget

	Randoop							T3							Evosuite							jTexPert						
CUT	gen_t	cov_i	cov_b	cov_m	F	U	B	gen_t	cov_i	cov_b	cov_m	F	U	B	gen_t	cov_i	cov_b	cov_m	F	U	B	gen_t	cov_i	cov_b	cov_m	F	U	B
Chart-11	63.2	56.4	34.4	7.5	1.0	0	0	48.8	71.0	57.6	12.1	1.0	0	0	63.6	81.0	58.0	13.3	.5	0	0	71.1	90.0	73.8	40.1	1.0	0	0
Chart-12	65.2	36.8	20.2	5.4	0	0	0	59.5	48.1	43.0	15.1	1.0	0	0	63.5	54.1	48.0	10.7	.1	0	0	111.2	44.8	32.3	7.9	0	0	0
Chart-16	65.2	56.3	52.9	51.0	1.0	0	0	59.3	40.8	32.4	22.7	1.0	0	0	62.5	81.8	74.6	47.3	1.0	0	0	82.1	72.8	62.2	25.5	1.0	0	0
Chart-17	64.2	0	0	0	0	.9	0	59.7	89.8	83.0	45.5	1.0	0	.1	63.4	69.2	58.9	27.9	1.0	0	0	90.9	54.6	47.4	22.3	.8	0	.4
Chart-1	65.5	34.2	22.7	13.8	0	0	0	10.1	0	0	0	0	0	0	67.7	34.5	22.5	11.3	0	0	0	106.4	27.1	20.1	5.5	.6	.3	0
Chart-20	64.9	68.1	25.0	27.2	0	0	0	59.6	77.2	72.9	51.5	0	0	0	59.6	95.4	87.5	81.8	0	0	0	63.3	90.9	75.0	45.4	0	0	0
Chart-23	65.6	40.6	27.6	2.4	0	0	0	59.7	28.7	16.6	7.1	1.0	0	0	61.7	41.6	30.6	3.3	0	0	0	116.2	30.0	18.1	5.4	.3	0	.2
Chart-24	63.6	92.0	80.0	75.6	1.0	0	0	13.8	100.0	100.0	91.4	1.0	0	0	59.4	100.0	100.0	91.8	1.0	0	0	73.2	79.3	81.6	35.5	.1	0	0
Chart-25	66.3	9.3	2.2	.5	0	0	0	59.7	10.9	7.0	1.0	0	0	0	62.0	13.0	8.3	.2	0	0	0	93.8	8.3	3.7	.6	0	0	0
Chart-26	67.6	46.2	22.7	8.7	0	0	0	9.7	0	0	0	0	0	0	65.1	54.5	37.5	5.7	0	0	0	104.9	0	0	0	0	.5	0
Chart-2	65.7	24.6	19.5	11.3	0	0	0	62.3	50.9	40.8	32.9	.1	0	0	69.0	0	0	0	0	1.0	0	77.8	44.9	38.5	18.1	0	0	0
Chart-3	64.9	0	0	0	0	1.0	0	59.8	88.6	73.3	55.2	0	0	.1	65.2	77.5	60.4	27.9	0	0	0	89.1	72.8	62.8	38.1	0	0	.3
Chart-4	66.5	45.2	30.4	11.9	.1	0	0	62.6	39.3	26.5	12.9	.3	0	0	77.9	45.4	29.4	17.2	.1	0	0	101.6	1.9	.8	.6	0	.4	.5
Chart-6	63.6	44.1	43.7	17.8	1.0	0	0	24.1	50.0	50.0	25.0	1.0	0	0	60.0	50.9	37.5	15.4	.8	0	.2	55.0	50.0	50.0	18.4	0	0	0
Chart-7	64.8	89.9	74.0	53.1	0	0	0	55.6	97.3	94.4	59.9	0	0	.1	61.7	58.6	38.2	21.7	0	0	0	65.5	85.4	76.8	65.7	0	0	.1
Chart-9	64.6	25.4	14.1	5.2	0	.9	0	59.7	90.7	82.7	47.9	0	0	.1	63.3	65.7	53.3	22.9	0	0	0	88.3	51.5	40.1	18.3	0	0	.4
Closure-100	65.4	37.7	14.2	8.1	0	0	0	69.0	8.8	3.5	4.7	0	0	0	60.1	24.4	3.5	8.1	0	0	0	70.5	41.4	16.3	7.6	0	0	0
Closure-124	65.4	1.3	0	0	0	0	0	72.7	9.4	4.4	2.7	0	0	0	60.4	9.4	0	1.3	0	0	0	60.0	4.0	1.4	1.3	0	0	0
Closure-130	65.7	6.9	2.2	0	0	.1	0	79.6	1.2	.5	0	0	0	0	61.8	9.1	2.5	0	0	0	0	62.1	7.2	3.0	0	0	0	0
Closure-132	65.1	5.0	1.4	.7	0	0	0	24.7	0	0	0	0	0	0	64.4	9.8	3.1	3.4	0	0	0	82.3	13.6	9.7	5.9	0	0	0
Closure-14	69.5	22.8	14.1	7.7	0	.1	0	73.5	0	0	0	0	0	0	62.1	9.4	1.0	2.6	0	0	0	63.9	22.8	12.0	8.9	0	0	0
Closure-16	69.6	3.6	0	0	0	0	0	73.3	4.6	2.1	2.6	.1	0	0	60.2	12.3	0	0	0	0	0	59.0	24.5	12.0	8.0	0	0	0
Closure-20	66.3	2.4	.5	.6	0	.1	0	24.2	0	0	0	0	0	0	63.9	11.0	3.9	3.9	0	0	0	83.2	12.7	9.3	4.6	0	0	0
Closure-46	64.5	6.8	0	0	0	0	0	37.5	6.8	0	0	0	0	0	61.2	2.2	0	0	0	0	0	56.9	0	0	0	0	0	1.0
Closure-68	64.7	21.5	13.9	6.2	0	0	0	68.5	15.1	8.8	4.7	0	0	0	75.0	16.6	11.2	3.6	0	0	0	121.0	0	0	0	0	0	0
Closure-74	66.0	2.3	.4	.1	0	0	0	63.5	6.6	3.6	1.2	0	0	0	64.2	5.6	.2	1.8	0	0	0	68.8	5.2	2.1	.7	0	0	0
Closure-98	68.9	22.7	8.3	3.0	0	0	0	70.6	0	0	0	0	0	0	61.4	29.2	5.8	8.2	0	0	0	81.4	51.0	21.4	15.7	0	0	0
Closure-99	66.7	13.3	0	0	0	0	0	70.3	0	0	0	0	0	0	60.2	24.4	3.2	6.9	0	0	0	69.6	37.7	11.0	4.9	0	0	0
Lang-28	64.4	12.0	6.2	10.2	0	0	0	13.9	12.0	18.7	15.8	0	0	0	58.7	12.0	6.2	8.3	0	0	0	54.5	24.0	18.7	5.0	0	0	0
Lang-33	64.1	80.2	58.2	44.6	0	0	0	7.4	0	0	0	0	0	0	63.0	80.7	60.8	36.1	1.0	0	0	80.2	89.4	83.4	50.5	1.0	0	0
Lang-36	77.1	84.9	68.1	36.2	0	0	0	34.6	89.6	73.8	38.2	0	0	0	63.3	83.9	67.1	27.1	.1	0	0	78.2	93.9	79.3	36.7	0	0	0
Lang-37	64.2	97.0	91.1	73.0	1.0	0	0	59.5	98.5	97.0	96.8	0	0	0	75.0	78.6	61.4	24.4	.1	0	0	100.7	93.8	80.1	56.5	0	0	0
Lang-41	63.3	69.0	52.1	48.3	1.0	0	0	6.6	0	0	0	0	0	0	62.1	81.4	61.2	39.0	1.0	0	0	74.5	90.0	84.3	54.4	1.0	0	0
Lang-43	64.8	12.2	1.0	0	0	0	0	38.8	12.2	1.0	0	0	0	0	60.2	49.5	31.9	12.4	1.0	0	0	59.7	54.6	43.2	19.8	1.0	0	0
Lang-47	63.4	80.6	71.5	40.3	1.0	0	0	58.9	93.5	88.9	75.9	.5	0	0	72.5	75.6	63.8	23.1	.6	0	0	121.0	91.5	84.8	51.1	.8	0	.1
Lang-50	65.7	77.9	71.9	44.9	0	0	0	49.9	93.6	88.4	61.2	0	0	.1	70.6	80.4	63.1	19.1	0	0	0	79.7	92.6	83.3	61.1	0	0	0
Lang-57	63.0	81.5	59.3	20.9	0	0	0	15.8	86.8	70.3	41.9	0	0	0	60.6	86.1	75.0	40.4	.8	0	0	61.8	95.6	82.2	54.9	.3	0	0
Lang-58	68.7	81.9	66.9	40.9	0	0	0	36.3	91.1	78.5	46.2	0	0	0	63.9	79.4	64.1	21.0	0	0	0	77.2	92.5	79.8	44.7	0	0	0
Lang-59	63.5	83.9	76.7	44.3	1.0	0	0	58.5	95.2	91.2	77.0	.8	0	0	72.2	78.5	68.7	25.2	.1	0	0	121.0	93.7	88.6	53.6	.1	0	0
Lang-60	63.6	82.5	75.1	44.8	1.0	0	0	58.8	95.7	91.7	78.9	1.0	0	0	70.7	77.3	67.2	23.8	.1	0	0	121.0	88.7	82.6	48.5	0	0	.1
Lang-63	78.6	68.1	65.4	44.1	0	0	0	67.5	75.9	72.1	64.8	.8	0	0	65.6	81.0	78.8	13.5	0	0	0	65.9	98.8	95.8	72.8	1.0	0	0
Lang-65	63.3	84.6	67.4	33.7	0	0	0	33.6	95.0	90.2	53.3	1.0	0	.1	61.8	72.5	56.5	28.1	.3	0	0	73.2	88.2	75.8	38.1	.3	0	0
Math-103	65.5	97.6	94.4	80.5	1.0	0	0	27.1	97.6	94.4	82.0	1.0	0	0	59.8	85.7	77.7	62.9	0	0	0	61.7	76.9	68.5	46.6	.8	0	0
Math-106	63.3	55.5	36.8	4.3	0	0	0	60.7	68.2	57.8	36.9	0	0	0	59.6	63.4	46.4	24.9	0	0	0	59.2	73.0	57.8	24.6	0	0	0
Math-18	65.4	6.6	.7	0	0	0	0	70.7	7.0	.7	.3	0	0	0	70.9	85.7	70.9	29.9	0	0	0	95.7	46.8	32.8	8.8	0	0	0
Math-20	65.5	6.7	.7	0	0	0	0	70.3	7.0	.7	.3	0	0	0	69.4	85.3	67.7	33.0	0	0	0	92.5	56.6	39.6	9.8	0	0	0
Math-21	65.9	78.1	75.0	28.4	0	0	0	69.1	0	0	0	0	0	0	61.0	58.4	58.3	23.6	0	0	0	79.1	71.8	71.4	21.4	0	0	0
Math-2	68.2	89.3	84.6	76.2	0	0	0	36.6	100.0	100.0	79.5	0	0	0	65.5	98.4	96.7	39.3	0	0	0	60.1	98.2	92.3	81.3	.1	0	0
Math-39	66.4	29.6	0	4.3	0	0	0	9.8	0	0	0	0	0	0	62.8	74.3	39.1	20.7	0	0	0	58.8	29.6	0	.4	0	0	0
Math-44	64.0	32.5	8.6	8.3	0	0	0	10.4	0	0	0	0	0	0	63.9	63.0	49.6	34.9	0	0	0	77.7	25.4	6.1	7.6	0	.1	0
Math-52	66.3	60.2	29.5	5.7	0	0	0	56.5	75.5	48.2	60.1	.5	0	0	64.6	72.8	45.4	52.8	0	0	0	70.5	68.6	68.3	37.2	0	0	0
Math-56	65.6	21.4	15.6	10.1	0	0	0	28.1	96.9	97.3	77.1	1.0	0	0	64.0	97.8	94.2	48.7	.1	0	0	76.3	94.9	82.2	46.9	.8	0	0
Math-64	63.7	5.4	0	.1	0	0	0	35.8	25.6	13.2	2.7	0	0	0	64.2	35.8	22.7	5.6	0	0	0	68.6	5.4	0	0	0	0	0
Math-67	63.9	0	0	0	0	0	0	68.3	92.4	84.8	53.1	1.0	0	0	68.4	87.5	81.8	36.2	1.0	0	0	91.9	34.0	7.8	1.4	0	0	0
Math-7	67.7	29.1	6.8	10.5	0	0	0	11.3	0	0	0	0	0	0	61.7	50.0	29.5	26.2	0	0	0	81.3	26.5	5.7	7.3	0	0	0
Math-88	65.8	10.6	4.4	1.0	0	0	0	62.8	10.6	4.4	1.0	0	0	0	60.1	8.3	2.2	.5	0	0	0	61.3	62.2	53.8	29.3	.1	0	0
Math-91	78.0	61.0	48.8	33.3	0	.5	0	33.0	93.2	83.3	61.8	0	0	0	63.6	96.5	94.0	58.6	0	0	0	61.5	92.9	82.5	66.6	0	0	0
Math-93	109.1	62.7	56.8	49.0	0	0	0	85.8	11.3	9.4	6.2	0	0	0	67.1	81.5	78.9	24.6	0	0	0	94.0	64.3	59.1	43.1	0	0	0
Time-10	64.8	67.1	63.0	32.4	0	0	0	7.6	0	0	0	0	0	0	61.7	91.4	87.3	46.3	0	0	0	61.2	82.3	74.2	68.5	0	0	0
Time-11	65.1	22.3	17.0	4.6	0	0	0	46.0	54.9	42.6	17.2	0	0	.1	63.3	30.6	25.9	11.3	1.0	0	0	81.6	30.2	26.0	6.4	0	0	.1
Time-13	64.6	72.5	56.6	27.8	.3	0	0	68.5	37.9	21.1	13.8	0	0	0	65.7	49.7	29.0	11.9	0	0	0	110.9	89.1	75.6	41.3	1.0	0	0
Time-20	64.6	66.0	51.1	20.4	.1	0	.1	61.4	75.8	63.2	28.9	0	0	0	70.0	35.3	25.2	9.3	0	0	0	97.6	72.9	60.1	34.4	0	0	0
Time-23	65.7	78.1	65.4	41.5	0	0	0	8.5	0	0	0	0	0	0	64.4	83.5	71.5	40.3	0	0	0	82.8	73.8	62.9	38.8	0	0	0
Time-3	64.8	75.9	67.4	14.2	.6	0	.1	65.1	70.0	51.8	19.4	.5	0	0	64.4	87.3	79.6	35.3	1.0	0	0	71.2	84.0	86.2	23.5	.5	0	0
Time-4	64.8	50.6	34.9	11.1	.1	0	.1	53.7	96.7	93.8	78.4	.8	0	.1	63.0	72.4	55.5	29.6	.1	0	0	79.0	93.2	90.4	57.3	.5	0	.1
Time-5	65.5	74.0	52.0	2.6	0	0	0	68.5	95.6	88.2	63.4	.6	0	0	69.2	96.7	80.9	25.1	.3	0	0	68.4	96.7	91.1	67.0	.3	0	0
Time-7	64.8	82.7	67.6	20.0	0	0	0	67.9	33.7	19.3	10.8	0	0	0	64.7	87.2	84.0	39.5	.3	0	0	79.8	86.2	67.9	38.2	0	0	.2
Time-8	64.5	75.0	60.2	38.6	0	0	0	7.7	0	0	0	0	0	0	63.8	81.3	68.3	33.7	.6	0	0	87.3	75.4	62.0	37.2	0	0	0

Table 7: Time budget 120 randoop t3 evosuite jtexpert

	Randoop							T3							Evosuite							jTexPert						
CUT	gen_t	cov_i	cov_b	cov_m	F	U	B	gen_t	cov_i	cov_b	cov_m	F	U	B	gen_t	cov_i	cov_b	cov_m	F	U	B	gen_t	cov_i	cov_b	cov_m	F	U	B
Chart-11	123.9	56.4	34.4	7.6	1.0	0	0	47.3	71.0	57.9	12.2	1.0	0	0	121.4	92.0	82.7	27.9	1.0	0	0	118.5	91.4	77.8	45.3	1.0	0	0
Chart-12	127.0	44.0	30.1	6.8	0	0	0	117.7	49.5	45.4	15.8	1.0	0	0	120.9	62.0	53.3	19.6	.1	0	0	174.5	52.2	39.6	9.2	.1	0	0
Chart-16	127.2	57.5	54.6	53.6	1.0	0	0	59.1	41.2	32.0	22.2	1.0	0	0	118.7	82.7	78.1	48.5	1.0	0	0	132.3	80.4	74.4	28.5	1.0	0	0
Chart-17	125.3	0	0	0	0	.9	0	120.5	92.7	88.2	54.5	1.0	0	0	120.2	83.0	74.2	32.7	1.0	0	0	141.1	57.7	51.4	21.2	.5	0	.4
Chart-1	128.4	34.2	22.7	13.8	0	0	0	10.3	0	0	0	0	0	0	124.3	36.8	26.3	12.1	0	0	0	207.2	30.7	23.9	6.1	.6	.1	0
Chart-20	126.2	68.1	25.0	27.2	0	0	0	73.9	86.3	77.0	54.5	0	0	0	104.1	95.4	87.5	81.8	0	0	0	113.3	90.9	75.0	45.4	0	0	0
Chart-23	127.3	41.6	30.3	3.5	0	0	0	119.6	31.0	20.2	8.3	1.0	0	0	117.8	44.1	36.6	8.4	.8	0	0	223.0	45.3	32.4	8.7	1.0	0	0
Chart-24	123.9	92.0	80.0	75.6	1.0	0	0	14.2	100.0	100.0	92.7	1.0	0	0	100.2	100.0	100.0	65.3	.3	0	0	103.2	82.0	88.3	34.6	.1	0	0
Chart-25	127.1	9.3	2.2	.5	0	0	0	120.0	12.8	8.7	1.3	0	0	0	112.9	13.4	8.8	1.8	0	0	0	140.4	11.1	6.4	.9	0	0	0
Chart-26	128.1	48.7	27.5	9.6	0	0	0	10.4	0	0	0	0	0	0	126.7	63.7	56.8	17.0	0	0	0	208.5	0	0	0	0	.5	0
Chart-2	128.1	27.9	21.8	13.2	1.0	0	0	120.2	51.9	42.0	33.1	.5	0	0	128.1	18.0	15.9	7.3	0	.6	0	132.6	55.1	47.2	26.5	0	0	0
Chart-3	126.8	15.7	9.4	3.6	0	.9	0	121.3	94.0	86.4	52.8	0	0	.1	122.2	78.7	64.6	33.8	.1	0	0	152.5	74.5	62.8	38.5	0	0	.3
Chart-4	128.1	51.9	36.3	23.5	0	0	0	125.9	35.4	25.0	13.5	.5	0	0	132.5	48.1	32.5	17.2	0	0	0	233.5	9.2	5.1	2.4	.1	.1	.4
Chart-6	124.4	44.1	43.7	17.8	1.0	0	0	25.4	50.0	50.0	25.0	1.0	0	0	113.8	54.9	48.9	23.8	1.0	0	0	107.0	50.0	50.0	17.8	0	0	0
Chart-7	126.8	95.3	85.1	61.0	0	0	0	57.8	96.9	94.4	60.4	0	0	.1	121.3	95.3	89.1	63.9	.1	0	0	122.4	84.6	76.5	60.2	0	0	.1
Chart-9	126.4	37.3	23.3	10.0	0	.9	0	121.0	91.3	86.1	47.8	0	0	.1	120.7	79.8	69.7	31.5	.1	0	0	146.2	59.4	50.9	24.9	0	0	.4
Closure-100	130.1	37.7	14.2	8.1	0	0	0	130.5	13.3	5.3	7.3	0	0	0	120.8	42.5	21.7	18.1	0	0	0	120.5	40.0	15.7	6.8	0	0	0
Closure-124	126.3	9.4	4.4	2.7	0	0	0	119.5	9.4	4.4	2.7	0	0	0	105.5	12.8	2.4	2.5	0	0	0	111.2	4.0	1.4	1.3	0	0	0
Closure-130	126.9	7.2	2.8	0	0	.1	0	141.9	5.0	2.0	0	0	0	.1	118.0	11.2	3.6	.4	0	0	0	111.8	7.2	3.0	0	0	0	0
Closure-132	126.2	5.6	2.5	1.2	0	0	0	23.0	0	0	0	0	0	0	119.4	13.3	4.7	8.8	0	0	0	134.6	14.0	10.3	7.9	0	0	0
Closure-14	127.6	23.1	14.3	9.7	0	.1	0	131.9	28.3	21.5	19.9	0	0	0	115.9	10.5	1.4	3.5	0	0	0	114.2	24.9	14.3	10.4	0	0	0
Closure-16	127.5	6.1	0	0	0	0	0	116.7	17.3	5.6	6.9	.6	0	0	109.3	18.2	5.6	4.0	0	0	0	108.9	25.0	13.3	8.7	0	0	0
Closure-20	128.2	3.6	2.4	2.8	0	.2	0	23.0	0	0	0	0	0	0	119.0	12.7	4.7	8.5	0	0	0	131.1	13.6	10.1	6.7	0	0	0
Closure-46	125.8	6.8	0	0	0	0	0	41.4	6.8	0	0	0	0	0	97.5	2.2	0	0	0	0	0	107.4	0	0	0	0	0	1.0
Closure-68	126.3	22.5	14.6	6.6	0	0	0	78.1	15.5	8.7	4.5	0	0	0	126.5	17.4	12.3	4.4	0	0	0	241.0	0	0	0	0	0	0
Closure-74	128.4	3.0	1.0	.1	0	0	0	73.7	6.5	3.6	1.2	0	0	0	118.9	5.7	.2	2.0	0	0	0	121.9	6.7	2.9	1.3	0	0	0
Closure-98	128.6	22.7	8.3	4.0	0	0	0	127.7	22.4	10.6	12.5	.5	0	0	114.5	35.3	8.9	9.1	0	0	0	127.3	56.8	26.6	16.1	0	0	0
Closure-99	126.2	20.0	3.2	4.4	0	0	0	125.9	18.5	7.7	8.3	0	0	0	118.2	43.3	20.1	16.9	0	0	0	118.4	40.0	14.2	6.5	0	0	0
Lang-28	126.1	12.0	6.2	10.2	0	0	0	13.5	12.0	18.7	15.4	0	0	0	94.8	23.3	27.0	17.7	0	0	0	104.4	48.6	40.6	8.1	0	0	0
Lang-33	125.0	80.9	59.4	48.6	.6	0	0	7.9	0	0	0	0	0	0	118.8	81.9	61.0	39.3	1.0	0	0	129.2	90.9	85.6	54.9	1.0	0	0
Lang-36	241.0	60.0	37.1	13.1	0	0	0	33.4	89.4	73.9	38.3	0	0	0	119.8	86.1	70.7	36.1	0	0	0	126.9	95.8	85.4	39.2	.6	0	0
Lang-37	125.0	98.5	94.8	78.7	1.0	0	0	92.9	98.5	98.0	96.3	0	0	0	124.1	83.5	66.4	35.0	0	0	0	149.7	94.4	81.6	70.5	0	0	0
Lang-41	123.8	69.0	52.7	48.5	1.0	0	0	7.2	0	0	0	0	0	0	118.3	84.7	65.5	44.2	1.0	0	0	121.9	91.3	85.1	52.6	1.0	0	0
Lang-43	126.5	12.2	1.0	0	0	0	0	42.4	12.2	1.0	0	0	0	0	114.4	54.0	37.8	16.5	1.0	0	0	109.1	54.4	41.7	19.9	1.0	0	0
Lang-47	124.1	85.7	78.9	58.9	1.0	0	0	119.4	98.5	96.5	88.7	1.0	0	0	127.2	81.9	72.2	32.1	.6	0	0	206.9	96.4	91.3	56.2	1.0	0	0
Lang-50	128.6	78.2	72.3	46.4	0	0	0	84.5	94.3	89.7	66.5	0	0	0	126.9	53.5	43.4	22.9	0	.3	0	128.6	93.3	83.7	69.9	0	0	0
Lang-57	123.3	81.5	59.3	20.9	0	0	0	15.8	87.7	71.0	42.5	0	0	0	112.2	86.4	70.8	39.1	1.0	0	0	112.4	94.2	80.4	53.2	.6	0	0
Lang-58	132.7	82.9	68.5	44.5	0	0	0	40.1	90.8	78.1	45.4	0	0	0	119.4	82.2	65.8	33.2	0	0	0	127.3	94.6	84.6	53.7	.6	0	0
Lang-59	124.2	84.7	79.4	50.8	1.0	0	0	119.0	98.9	97.4	88.0	1.0	0	0	125.4	79.9	71.3	31.1	.3	0	0	194.2	96.8	91.9	56.3	.3	0	0
Lang-60	124.2	87.5	80.9	56.7	1.0	0	0	118.8	98.9	97.5	86.4	1.0	0	0	126.5	80.9	72.8	34.0	.1	0	0	194.9	97.4	92.6	58.1	.1	0	0
Lang-63	238.4	68.1	65.4	44.2	0	0	0	128.0	76.2	72.4	67.0	.8	0	0	121.5	84.7	82.5	20.3	0	0	0	113.9	98.7	95.7	73.2	1.0	0	0
Lang-65	123.8	85.5	69.1	33.9	0	0	0	34.7	95.1	89.9	55.4	.8	0	0	118.9	89.2	81.3	35.6	.1	0	0	123.0	90.7	77.4	41.9	.8	0	0
Math-103	126.1	97.6	94.4	82.4	1.0	0	0	26.6	97.6	94.4	81.3	1.0	0	0	102.2	85.7	77.7	62.8	0	0	0	111.7	76.9	68.5	46.1	.8	0	0
Math-106	124.2	55.5	36.8	4.3	0	0	0	77.0	68.2	57.8	36.9	0	0	0	101.7	80.1	70.1	32.2	0	0	0	109.8	75.6	63.1	31.1	.6	0	0
Math-18	128.7	6.9	.7	0	0	0	0	131.4	4.7	.4	.1	0	0	0	130.7	86.1	72.2	29.0	0	0	0	146.8	62.2	46.0	11.9	0	0	0
Math-20	126.7	6.7	.7	0	0	0	0	131.3	4.7	.4	.2	0	0	0	126.3	87.3	71.8	35.9	0	0	0	143.0	50.2	34.2	6.7	0	0	0
Math-21	134.8	78.1	75.0	28.4	0	0	0	129.5	0	0	0	0	0	0	114.2	69.6	70.2	32.5	0	0	0	131.4	87.5	85.1	24.6	0	0	0
Math-2	131.6	100.0	100.0	97.2	0	0	0	57.8	100.0	100.0	79.2	0	0	0	117.9	98.2	96.1	84.7	.1	0	0	109.2	99.2	94.8	76.7	0	0	0
Math-39	130.8	29.6	0	4.3	0	0	0	9.3	0	0	0	0	0	0	119.4	67.1	36.1	17.2	0	0	0	108.1	29.6	0	.4	0	0	0
Math-44	127.9	32.5	8.6	9.7	0	0	0	10.5	0	0	0	0	0	0	121.1	62.6	50.7	32.6	0	0	0	126.6	19.7	4.3	6.2	0	.3	0
Math-52	123.8	60.5	29.5	5.8	0	0	0	66.9	73.5	47.6	57.2	1.0	0	0	121.8	81.2	59.0	58.8	0	0	0	119.7	78.6	72.2	49.1	0	0	0
Math-56	126.1	21.4	15.6	10.1	0	0	0	51.0	95.2	95.3	76.9	.8	0	0	121.0	99.2	95.8	56.1	.1	0	0	144.1	95.4	84.8	49.5	1.0	0	0
Math-64	124.4	10.2	0	.1	0	0	0	38.5	25.8	13.2	2.6	0	0	0	117.9	40.9	27.6	5.8	.6	0	0	119.9	5.4	0	0	0	0	0
Math-67	124.8	0	0	0	0	0	0	128.6	87.5	76.4	47.7	1.0	0	0	122.1	88.7	90.6	52.2	1.0	0	0	142.4	73.1	60.7	29.4	.1	0	0
Math-7	127.0	29.1	6.8	10.9	0	0	0	11.6	0	0	0	0	0	0	120.2	58.9	39.3	32.5	0	0	0	130.1	26.3	5.1	8.1	0	0	0
Math-88	125.3	10.6	4.4	1.0	0	0	0	74.3	10.6	4.4	1.0	0	0	0	119.3	83.9	74.8	41.5	0	0	.2	114.3	68.1	59.0	32.7	0	0	0
Math-91	141.0	87.9	73.3	53.9	0	.2	0	35.3	93.4	83.5	62.1	0	0	0	122.2	97.2	95.9	69.1	0	0	0	112.1	93.6	83.3	67.8	0	0	0
Math-93	125.7	76.8	71.3	60.9	0	0	0	139.6	24.5	22.3	16.5	0	0	0	127.8	84.0	80.4	46.0	.1	0	0	168.9	77.7	72.7	53.1	.1	0	0
Time-10	126.3	75.0	72.1	36.8	0	0	0	7.5	0	0	0	0	0	0	119.3	91.4	90.5	63.2	0	0	0	112.3	83.8	76.4	69.2	0	0	0
Time-11	126.6	18.9	13.4	3.6	0	0	0	70.0	56.7	44.8	17.3	0	0	.1	119.3	32.5	27.8	15.6	1.0	0	0	131.6	35.0	31.0	13.4	0	0	0
Time-13	125.7	74.5	59.7	27.8	.6	0	.1	129.0	57.5	41.2	21.5	0	0	0	123.7	52.7	31.0	16.7	0	0	0	168.7	92.4	81.8	49.5	1.0	0	0
Time-20	126.2	65.6	51.7	20.7	.1	0	.2	119.0	81.7	70.7	36.1	0	0	0	128.5	36.4	25.5	11.1	0	0	0	162.0	80.0	69.6	44.6	0	0	0
Time-23	127.1	78.1	65.4	44.0	0	0	0	8.5	0	0	0	0	0	0	120.5	83.9	72.7	49.2	.1	0	0	132.2	77.9	66.4	41.5	0	0	0
Time-3	126.1	77.2	70.8	30.9	.6	0	.1	124.9	80.6	68.2	38.5	.6	0	.1	123.2	91.0	84.3	47.5	.5	0	0	120.2	86.3	87.5	22.0	0	0	0
Time-4	126.5	54.6	37.2	10.9	.3	0	.2	91.1	99.1	98.5	81.0	1.0	0	.1	121.6	76.9	61.0	38.7	0	0	0	139.6	93.9	92.1	57.9	0	0	.1
Time-5	127.3	75.4	56.7	3.2	0	0	0	128.8	93.2	79.9	55.0	.5	0	0	124.9	97.3	84.3	45.1	.5	0	0	117.7	97.5	92.9	66.3	.6	0	0
Time-7	126.8	85.2	75.3	20.8	0	0	0	116.7	34.6	21.5	11.7	0	0	0	124.5	86.7	80.3	46.1	.1	0	0	141.9	87.6	71.0	36.0	0	0	.2
Time-8	125.7	75.9	61.4	40.1	0	0	0	8.7	0	0	0	0	0	0	120.6	81.3	70.4	47.6	.8	0	0	130.4	75.8	62.2	39.0	.1	0	0

Table 8: Time budget 240 randoop t3 evosuite jtexpert

	Randoop							T3							Evosuite							jTexPert						
CUT	gen_t	cov_i	cov_b	cov_m	F	U	B	gen_t	cov_i	cov_b	cov_m	F	U	B	gen_t	cov_i	cov_b	cov_m	F	U	B	gen_t	cov_i	cov_b	cov_m	F	U	B
Chart-11	245.1	56.4	34.4	7.6	1.0	0	0	51.8	71.1	58.0	12.2	1.0	0	0	315.5	62.0	57.3	18.8	.6	0	0	223.5	76.5	64.9	38.6	.8	0	0
Chart-12	250.5	48.6	38.0	7.3	0	0	0	161.6	49.1	45.0	15.8	1.0	0	0	230.4	64.8	55.5	20.3	0	0	0	297.3	53.0	43.6	10.4	0	0	0
Chart-16	250.7	60.2	58.5	54.7	1.0	0	0	65.8	42.6	34.1	24.7	1.0	0	0	231.2	85.2	80.0	52.6	1.0	0	0	241.6	83.0	79.1	30.9	1.0	0	0
Chart-17	247.7	40.0	25.1	12.0	.8	.9	0	189.8	93.3	89.6	55.4	1.0	0	.1	234.3	87.2	80.9	39.0	1.0	0	0	250.5	64.4	57.3	25.0	.8	0	.3
Chart-1	250.9	34.2	22.7	13.8	0	0	0	8.8	0	0	0	0	0	0	244.5	51.0	39.9	17.4	.1	0	0	314.8	39.8	31.6	9.8	.8	.1	0
Chart-20	249.5	86.3	62.5	45.4	1.0	0	0	68.6	82.5	68.7	45.4	0	0	0	190.4	95.4	87.5	81.8	0	0	0	220.8	90.9	75.0	45.4	0	0	0
Chart-23	251.2	41.6	30.3	3.5	0	0	0	240.1	35.2	28.2	13.5	1.0	0	0	269.2	43.0	34.8	6.9	.5	0	0	326.0	41.6	32.1	8.4	.8	0	0
Chart-24	244.0	92.0	80.0	75.6	1.0	0	0	13.9	100.0	100.0	90.9	1.0	0	0	183.0	100.0	100.0	73.4	.5	0	0	172.0	88.6	91.6	43.2	.5	0	0
Chart-25	251.0	9.3	2.2	.5	0	0	0	239.6	19.9	11.2	1.4	0	0	0	344.5	18.4	10.0	2.3	.5	0	0	255.4	13.9	9.4	1.2	0	0	0
Chart-26	252.4	53.1	32.8	11.8	0	0	0	10.5	0	0	0	0	0	0	243.4	73.5	63.2	19.7	.1	0	0	317.5	0	0	0	0	.8	0
Chart-2	251.4	29.3	22.9	14.6	1.0	0	0	133.3	53.2	42.5	33.2	.6	0	0	244.0	67.2	62.1	43.3	0	0	0	240.1	63.4	56.2	32.9	0	0	0
Chart-3	249.8	0	0	0	0	.9	0	227.0	94.5	87.6	49.7	0	0	.1	235.9	90.3	81.3	44.9	.3	0	0	258.7	76.8	68.0	40.1	0	0	.3
Chart-4	251.5	53.3	38.3	19.6	0	0	0	246.5	45.9	35.1	27.8	.6	0	0	251.1	55.8	41.6	19.7	1.0	0	0	427.2	0	0	0	0	.7	.1
Chart-6	246.0	44.1	43.7	17.8	1.0	0	0	22.8	50.0	50.0	25.0	1.0	0	0	186.6	55.3	48.9	19.0	.8	0	0	209.1	50.0	50.0	17.8	0	0	0
Chart-7	249.6	95.9	87.0	64.1	0	0	0	52.0	97.3	94.4	60.8	0	0	.1	232.2	96.9	95.3	67.3	.5	0	0	224.3	89.5	87.3	71.5	.1	0	.1
Chart-9	248.8	46.7	33.6	15.7	0	.8	0	163.4	94.5	92.2	57.6	0	0	.1	233.5	83.6	75.5	36.1	0	0	0	251.4	63.0	57.0	28.8	0	0	.3
Closure-100	250.1	37.7	14.2	8.1	0	0	0	249.7	26.2	11.0	12.7	0	0	0	223.1	40.7	21.1	20.5	0	0	0	222.5	43.7	18.1	7.6	0	0	0
Closure-124	249.1	9.4	4.4	2.7	0	0	0	147.7	9.4	4.4	2.7	0	0	0	223.5	36.7	22.3	15.0	0	0	0	213.3	4.0	1.4	1.3	0	0	0
Closure-130	251.9	7.2	3.0	0	0	.3	0	262.5	7.5	3.0	.1	0	0	.1	228.5	13.1	5.3	.7	0	0	0	214.8	7.2	3.0	0	0	0	0
Closure-132	251.6	6.2	3.5	2.0	0	0	0	22.4	0	0	0	0	0	0	241.8	11.3	6.2	7.4	0	0	0	236.7	14.3	10.6	7.5	0	0	0
Closure-14	250.6	25.3	16.4	12.3	0	.1	0	253.8	28.4	21.5	20.1	0	0	0	238.5	31.8	20.4	19.9	0	0	0	218.7	25.4	14.9	9.9	0	0	0
Closure-16	250.9	24.2	12.0	9.0	0	0	0	131.9	28.9	11.4	15.7	1.0	0	0	209.9	24.2	11.2	5.6	0	0	0	211.9	25.0	13.3	11.9	0	0	0
Closure-20	251.6	3.6	2.4	2.8	0	.1	0	25.7	0	0	0	0	0	0	236.6	12.6	7.2	10.0	0	0	0	239.2	14.2	10.7	8.9	0	0	0
Closure-46	247.4	6.8	0	0	0	0	.1	39.8	6.8	0	0	0	0	0	232.5	75.5	63.6	37.3	.1	.1	0	209.6	0	0	0	0	0	1.0
Closure-68	248.7	26.1	16.4	9.2	0	0	0	84.1	14.8	7.9	3.7	0	0	0	234.2	22.8	16.5	9.3	0	0	0	406.5	0	0	0	0	0	1.0
Closure-74	252.8	3.1	1.1	.1	0	0	0	71.7	5.7	3.0	.8	0	0	0	238.6	8.9	5.0	2.4	0	0	0	224.5	7.1	3.3	1.7	0	0	0
Closure-98	252.4	22.7	8.3	4.0	0	0	0	233.6	20.0	10.2	12.9	.3	0	0	228.1	57.2	20.6	22.4	0	0	0	232.0	58.0	29.4	17.1	0	0	0
Closure-99	249.0	24.4	5.3	3.2	0	0	0	251.0	22.2	8.0	10.4	0	0	0	220.4	42.5	19.8	18.1	0	0	0	222.3	45.1	19.6	8.0	0	0	0
Lang-28	249.3	12.0	6.2	10.4	0	0	0	15.4	12.0	18.7	13.5	0	0	0	165.3	47.3	51.0	24.1	0	0	0	207.5	46.6	42.7	5.8	0	0	0
Lang-33	246.9	80.9	60.0	49.0	1.0	0	0	7.7	0	0	0	0	0	0	220.0	84.5	66.6	43.6	1.0	0	0	229.5	91.4	86.0	60.4	1.0	0	0
Lang-36	245.2	0	0	0	0	0	0	34.6	90.0	74.8	36.9	0	0	0	231.0	90.3	75.9	38.7	.8	0	0	232.1	96.3	87.4	47.2	.8	0	0
Lang-37	246.7	98.9	96.1	81.2	1.0	0	0	95.2	98.5	97.9	97.3	0	0	0	238.2	88.0	72.1	56.6	0	0	0	255.5	94.6	81.8	61.3	0	0	0
Lang-41	244.6	69.0	52.7	48.5	1.0	0	0	6.5	0	0	0	0	0	0	225.1	86.1	67.7	46.4	1.0	0	0	235.2	91.4	86.4	57.5	1.0	0	0
Lang-43	251.6	12.2	1.0	0	0	0	0	38.9	12.2	1.0	0	0	0	0	198.7	53.2	41.5	19.3	1.0	0	0	213.0	54.6	42.7	20.6	1.0	0	0
Lang-47	245.3	88.8	83.4	67.3	1.0	0	0	169.2	98.7	96.7	86.9	1.0	0	0	244.3	86.2	78.6	43.9	.6	0	0	319.5	98.6	93.7	59.0	1.0	0	0
Lang-50	252.5	78.2	72.3	46.6	0	0	0	104.5	94.4	89.7	60.1	0	0	.1	238.7	85.1	73.8	50.8	0	0	0	231.0	92.3	82.2	75.3	0	0	0
Lang-57	244.1	81.5	59.3	20.9	0	0	0	16.6	86.8	70.3	41.6	0	0	0	178.6	83.9	66.4	37.6	1.0	0	0	215.3	94.2	80.2	54.0	.5	0	0
Lang-58	274.7	83.6	69.8	48.3	0	0	0	39.9	90.4	78.5	46.7	0	0	0	230.1	86.2	73.4	37.6	0	0	0	231.2	95.1	86.5	50.2	.6	0	0
Lang-59	246.0	88.5	83.6	59.5	1.0	0	0	152.4	99.0	97.5	85.6	1.0	0	0	244.2	86.0	79.0	44.7	.8	0	0	306.2	98.5	94.1	61.3	.3	0	0
Lang-60	245.7	89.1	83.7	65.2	1.0	0	0	140.2	99.0	97.5	86.1	1.0	0	0	242.0	85.4	77.0	44.9	.1	0	0	300.2	98.9	94.0	58.8	.1	0	0
Lang-63	248.0	0	0	0	0	0	0	247.5	91.4	86.8	76.4	.8	0	0	230.3	82.6	80.2	20.7	0	0	0	216.0	98.9	95.9	70.8	.8	0	0
Lang-65	244.7	86.0	70.2	33.7	0	0	0	31.7	95.4	90.1	53.2	1.0	0	0	221.0	94.0	87.9	51.9	.8	0	0	224.6	91.5	78.7	45.2	.5	0	0
Math-103	246.0	97.6	94.4	82.4	1.0	0	0	29.6	97.6	94.4	81.9	1.0	0	0	174.1	85.7	77.7	63.7	0	0	0	209.3	60.3	51.8	36.4	.6	0	0
Math-106	245.1	55.5	36.8	4.3	0	0	0	75.6	68.2	57.8	36.9	0	0	0	171.2	85.4	76.3	40.9	.1	0	0	213.1	79.8	68.4	28.2	1.0	0	0
Math-18	249.3	7.0	.7	0	0	0	0	252.8	7.0	.7	.3	0	0	0	239.7	82.2	66.7	26.4	0	0	0	255.5	77.2	62.0	17.6	0	0	0
Math-20	248.1	6.7	.7	0	0	0	0	251.1	7.0	.7	.3	0	0	0	240.1	86.7	71.9	28.7	0	0	0	248.9	72.4	55.4	15.7	0	0	0
Math-21	248.8	78.1	75.0	28.4	0	0	0	281.5	0	0	0	0	0	0	178.9	73.3	73.2	28.1	0	0	0	232.9	87.5	80.3	26.1	0	0	0
Math-2	256.5	100.0	100.0	98.4	.5	0	0	56.5	100.0	100.0	81.5	0	0	0	225.9	99.7	98.7	89.5	0	0	0	211.7	99.7	95.5	78.4	.1	0	0
Math-39	255.0	29.6	0	4.3	0	0	0	10.1	0	0	0	0	0	0	236.0	89.5	69.1	30.3	0	0	0	211.2	29.6	0	1.0	0	0	0
Math-44	249.8	32.5	8.6	10.1	0	0	0	10.8	0	0	0	0	0	0	234.7	68.1	58.3	34.9	0	0	0	237.4	29.8	6.5	8.7	0	0	0
Math-52	245.2	61.4	31.1	7.8	0	0	0	62.7	76.1	48.6	61.3	.8	0	0	237.7	97.0	88.6	79.5	0	0	0	222.9	74.7	69.6	45.3	0	0	0
Math-56	247.0	21.4	15.6	10.1	0	0	0	42.1	97.8	99.4	81.0	.8	0	0	232.7	99.7	98.4	74.2	.8	0	0	262.2	94.2	81.2	48.3	.8	0	0
Math-64	246.1	10.2	0	.1	0	0	0	37.1	25.4	12.9	2.7	.1	0	0	209.4	66.5	54.9	16.4	.6	0	0	224.9	5.4	0	0	0	0	0
Math-67	246.8	0	0	0	0	0	0	258.8	73.8	61.2	36.4	.8	0	0	237.1	86.8	86.7	47.4	1.0	0	0	245.3	66.4	50.4	23.7	.1	0	0
Math-7	247.8	29.1	6.8	10.1	0	0	0	11.1	0	0	0	0	0	0	231.8	74.6	59.1	33.1	0	0	0	231.2	18.3	4.0	5.0	0	.3	0
Math-88	247.8	10.6	4.4	1.0	0	0	0	77.7	10.6	4.4	1.0	0	0	0	227.9	94.2	86.8	52.8	.1	0	0	221.0	67.9	60.0	30.8	0	0	0
Math-91	281.5	87.9	71.1	54.1	0	.3	0	35.4	93.4	83.5	62.1	0	0	0	225.6	97.5	97.4	70.7	0	0	0	216.2	93.7	83.7	67.4	0	0	0
Math-93	419.3	78.1	74.1	63.1	0	0	0	278.0	47.4	39.4	26.8	0	0	0	240.9	85.9	83.4	65.7	0	0	0	381.1	52.4	50.1	35.9	0	0	0
Time-10	249.8	81.4	82.6	42.2	0	0	0	7.7	0	0	0	0	0	0	214.8	88.5	86.5	59.6	.1	0	0	214.9	84.5	77.1	69.4	0	0	0
Time-11	249.4	20.2	15.0	6.4	0	0	0	79.6	58.1	46.3	18.3	0	0	.1	219.0	46.3	40.1	19.4	1.0	0	0	238.5	36.7	32.2	14.9	0	0	0
Time-13	248.2	77.6	63.7	29.7	.8	0	0	248.9	58.7	43.5	23.4	0	0	0	238.4	65.4	44.5	25.1	0	0	0	298.7	94.2	84.2	52.1	1.0	0	0
Time-20	248.4	71.9	57.9	28.2	.3	0	.2	203.4	83.2	74.5	37.9	0	0	0	244.3	68.4	52.6	27.7	0	0	0	273.2	82.5	71.6	43.7	0	0	0
Time-23	250.3	78.1	65.4	44.1	0	0	0	8.5	0	0	0	0	0	0	232.1	83.9	73.8	50.2	0	0	0	237.7	76.7	65.1	39.9	0	0	0
Time-3	249.4	78.0	75.1	32.7	.5	0	.1	226.1	84.8	78.3	65.3	1.0	0	.1	236.2	93.5	90.4	55.9	.3	0	0	224.1	88.3	88.3	23.3	0	0	0
Time-4	249.1	62.5	48.7	14.3	0	0	.1	100.2	98.9	98.5	80.3	1.0	0	.1	234.1	93.5	87.1	67.9	.5	0	0	245.8	94.1	93.3	61.4	.5	0	0
Time-5	250.5	75.6	57.8	3.2	0	0	0	248.4	97.3	92.4	67.1	.8	0	0	240.9	98.4	90.6	73.4	.8	0	0	232.6	97.1	92.9	66.8	.5	0	0
Time-7	249.3	86.4	78.4	22.2	0	0	0	148.5	38.4	24.8	13.0	0	0	0	237.2	90.0	84.0	54.6	.1	0	0	245.9	90.6	75.9	40.3	0	0	.1
Time-8	248.8	76.4	65.1	43.2	.5	0	0	8.2	0	0	0	0	0	0	234.6	82.8	72.8	51.5	.6	0	0	235.4	75.3	61.6	38.3	0	0	0

Table 9: Time budget 480 randoop t3 evosuite jtexpert

	Randoop							T3							Evosuite							jTexPert						
CUT	gen_t	cov_i	cov_b	cov_m	F	U	B	gen_t	cov_i	cov_b	cov_m	F	U	B	gen_t	cov_i	cov_b	cov_m	F	U	B	gen_t	cov_i	cov_b	cov_m	F	U	B
Chart-11	486.7	56.4	34.4	7.6	1.0	0	0	57.1	70.9	57.7	12.2	1.0	0	0	477.9	79.4	76.5	25.3	.8	0	0	428.3	62.0	55.3	31.9	.8	0	0
Chart-12	494.5	48.6	38.0	7.3	0	0	0	152.0	49.3	45.2	16.2	1.0	0	0	596.9	51.3	45.8	17.6	0	0	0	520.6	57.3	49.4	12.1	0	0	0
Chart-16	494.7	61.2	60.5	56.2	1.0	0	0	64.4	43.2	35.1	24.2	1.0	0	0	412.2	88.1	85.0	55.3	1.0	0	0	447.5	85.3	82.4	34.7	1.0	0	0
Chart-17	491.9	50.0	31.8	14.9	1.0	.9	0	202.1	93.7	90.7	49.4	1.0	0	.1	394.9	89.4	84.5	42.3	1.0	0	0	457.4	68.2	63.8	32.2	.6	0	.3
Chart-1	494.2	34.2	22.7	13.8	0	0	0	11.0	0	0	0	0	0	0	468.7	56.1	45.7	22.3	0	0	0	460.1	32.3	25.9	7.2	.6	0	0
Chart-20	492.6	86.3	62.5	56.0	1.0	0	0	71.7	66.6	52.0	34.8	0	0	0	298.1	80.3	75.0	69.6	0	0	0	419.6	90.9	75.0	45.4	0	0	0
Chart-23	493.8	41.6	30.3	3.5	0	0	0	422.6	35.5	28.5	12.6	1.0	0	0	573.5	30.7	26.1	6.2	.6	0	0	572.2	51.6	41.9	12.4	1.0	0	0
Chart-24	483.9	92.0	80.0	75.6	1.0	0	0	14.6	100.0	100.0	91.4	1.0	0	0	323.2	100.0	100.0	82.8	.6	0	0	143.9	84.6	88.3	39.1	.3	0	0
Chart-25	495.0	9.3	2.2	.5	0	0	0	346.0	20.9	12.2	1.6	0	0	0	501.3	26.7	16.4	2.2	.5	0	0	459.4	11.6	8.3	1.3	0	0	0
Chart-26	495.6	56.7	38.8	12.6	.1	0	0	10.5	0	0	0	0	0	0	619.1	65.7	57.1	17.5	0	0	0	563.9	0	0	0	0	.8	0
Chart-2	494.0	29.3	22.9	14.6	1.0	0	0	154.0	52.9	42.3	32.9	.3	0	0	475.3	71.3	66.9	49.8	0	0	0	443.7	66.6	59.8	35.4	0	0	0
Chart-3	494.0	24.2	13.4	7.1	0	.9	0	247.8	95.1	90.5	51.6	0	0	.1	444.7	93.2	87.5	47.7	.5	0	0	461.5	81.2	71.5	42.0	0	0	.2
Chart-4	498.5	59.1	46.0	29.7	.3	0	0	490.2	57.3	46.4	39.3	.8	0	0	489.1	60.7	47.5	36.6	1.0	0	0	811.3	7.4	4.8	2.7	.1	.3	.1
Chart-6	489.7	44.1	43.7	17.8	1.0	0	0	23.9	50.0	50.0	25.0	1.0	0	0	326.7	55.3	50.0	20.2	1.0	0	0	413.9	49.5	48.9	17.2	0	0	0
Chart-7	492.6	96.6	88.8	66.9	.1	0	0	63.1	97.3	94.4	59.9	0	0	.1	375.6	98.0	98.4	61.2	.6	0	0	427.4	89.9	87.6	74.9	.1	0	.1
Chart-9	493.4	52.7	41.5	20.4	0	.8	0	205.2	93.8	90.5	53.7	.1	0	.1	416.2	92.2	87.0	44.0	0	0	0	456.8	71.1	65.0	31.7	.1	0	.3
Closure-100	495.4	48.8	24.7	24.6	.6	0	0	422.4	26.6	10.7	14.2	0	0	0	369.1	51.1	29.1	23.6	.1	0	0	425.8	46.6	23.5	10.5	0	0	0
Closure-124	492.1	9.4	4.4	2.7	0	0	0	137.6	9.4	4.4	2.7	0	0	0	369.5	39.8	25.7	16.8	0	0	0	416.8	4.0	1.4	1.3	0	0	0
Closure-130	495.5	7.2	3.0	0	0	.1	0	504.0	7.5	3.0	0	0	0	.1	424.0	15.3	6.5	1.3	0	0	0	417.6	7.2	3.0	0	0	0	0
Closure-132	493.2	6.2	3.5	2.0	0	0	0	23.5	0	0	0	0	0	0	469.1	14.1	8.3	13.7	0	0	0	441.7	14.2	10.6	7.5	0	0	0
Closure-14	494.5	26.4	18.4	13.0	0	0	0	494.5	27.9	21.1	19.1	0	0	0	451.6	37.6	25.9	26.3	0	0	0	424.2	27.7	17.3	11.6	0	0	0
Closure-16	495.3	24.2	12.0	9.0	0	0	0	210.4	30.4	12.3	14.7	1.0	0	0	354.5	24.2	11.8	4.4	0	0	0	415.9	25.2	13.7	12.4	0	0	0
Closure-20	495.7	3.6	2.4	2.8	0	0	0	26.5	0	0	0	0	0	0	447.1	17.8	10.8	13.4	0	0	0	442.8	13.7	10.4	8.2	0	0	0
Closure-46	490.2	6.8	0	0	0	0	.1	42.0	6.8	0	0	0	0	0	376.6	91.0	75.6	43.1	0	0	0	413.2	0	0	0	0	0	1.0
Closure-68	492.3	28.3	20.0	11.3	0	0	0	89.4	14.8	8.0	4.0	0	0	0	446.8	33.5	24.5	13.4	0	0	0	607.4	0	0	0	0	0	1.0
Closure-74	496.6	3.1	1.1	.1	0	0	0	74.1	5.7	3.1	1.1	0	0	0	444.6	10.9	6.5	3.1	0	0	0	427.8	7.4	3.4	1.7	0	0	0
Closure-98	495.7	22.7	8.3	4.0	0	0	0	375.5	58.9	31.5	41.4	1.0	0	0	389.4	62.9	26.3	27.5	0	0	0	439.5	60.0	31.5	18.1	0	0	0
Closure-99	493.8	42.5	13.9	4.4	0	0	0	376.3	27.4	11.0	12.9	0	0	0	365.1	48.5	23.9	21.1	0	0	0	427.7	46.6	21.2	8.6	0	0	0
Lang-28	494.5	12.0	6.2	10.2	0	0	0	15.8	12.0	14.5	15.4	0	0	0	304.8	75.3	79.1	49.7	0	0	0	411.7	65.3	59.3	13.1	0	0	0
Lang-33	489.7	83.5	65.5	52.0	1.0	0	0	7.7	0	0	0	0	0	0	377.9	76.9	59.8	38.1	.8	0	0	435.1	91.4	86.5	59.4	1.0	0	0
Lang-36	401.3	0	0	0	0	0	0	35.9	88.8	72.2	36.4	0	0	0	399.6	92.0	80.0	41.8	.8	0	0	432.3	96.3	87.3	48.0	1.0	0	0
Lang-37	489.1	99.0	97.3	82.7	1.0	0	0	100.1	98.5	98.1	96.5	0	0	0	460.2	90.1	75.0	58.2	.3	0	0	456.6	94.5	81.4	64.8	0	0	0
Lang-41	486.5	69.0	52.7	48.5	1.0	0	0	7.5	0	0	0	0	0	0	368.3	76.1	59.1	36.8	1.0	0	0	429.2	91.7	86.4	59.3	1.0	0	0
Lang-43	500.3	12.2	1.0	0	0	0	0	41.2	12.2	1.0	0	0	0	0	340.8	62.8	54.3	25.8	1.0	0	0	416.9	54.3	42.5	21.0	1.0	0	0
Lang-47	487.4	90.6	86.7	74.8	1.0	0	0	178.9	98.7	96.8	86.3	1.0	0	0	476.0	90.5	84.6	54.6	.8	0	0	519.2	98.4	93.5	58.8	1.0	0	0
Lang-50	494.1	78.2	72.3	46.7	0	0	0	128.8	94.3	89.7	60.2	0	0	0	445.3	86.2	73.0	56.5	.1	0	0	432.8	93.4	83.4	65.1	0	0	0
Lang-57	485.0	81.5	59.3	20.9	0	0	0	15.8	87.0	70.8	42.4	0	0	0	318.5	85.7	70.0	38.4	1.0	0	0	419.1	94.5	80.7	58.2	.3	0	0
Lang-58	566.2	5.8	2.9	1.5	0	.1	0	38.5	90.9	78.6	46.1	0	0	0	418.4	89.7	77.6	42.4	0	0	0	435.9	95.4	86.8	48.1	.3	0	0
Lang-59	488.7	91.2	86.6	71.0	1.0	0	0	137.5	99.0	97.6	85.3	1.0	0	0	473.7	89.3	82.0	49.9	.6	0	0	511.4	98.8	94.3	59.7	.6	0	0
Lang-60	487.8	90.4	86.8	71.2	1.0	0	0	145.6	99.0	97.6	85.9	1.0	0	0	479.7	88.6	81.2	48.5	.3	0	0	511.1	98.5	94.7	59.0	.3	0	0
Lang-63	387.5	0	0	0	0	0	0	488.0	91.1	86.3	77.8	1.0	0	0	427.3	92.0	90.1	25.5	.1	0	0	421.7	99.0	96.7	76.2	1.0	0	0
Lang-65	486.5	87.0	71.4	36.2	0	0	0	34.4	94.8	90.2	55.2	1.0	0	0	366.9	97.5	93.6	55.3	1.0	0	0	430.2	91.9	79.1	43.6	.1	0	0
Math-103	488.8	97.6	94.4	83.3	1.0	0	0	30.7	97.6	94.4	83.1	1.0	0	0	309.0	85.7	77.7	62.9	0	0	0	419.4	91.2	79.6	55.8	1.0	0	0
Math-106	487.4	72.4	57.8	30.4	0	0	0	71.4	68.2	57.8	36.9	0	0	0	308.2	79.8	70.1	35.1	0	0	0	418.6	84.9	76.3	37.3	1.0	0	0
Math-18	543.4	7.0	.7	0	0	0	0	493.7	6.8	.7	.3	0	0	0	455.4	84.1	69.6	26.5	0	0	0	461.8	75.1	58.0	17.1	0	0	0
Math-20	481.1	0	0	0	0	0	0	493.6	7.2	.7	.3	0	0	0	454.8	87.2	72.9	27.2	0	0	0	473.2	79.6	64.2	20.6	0	0	0
Math-21	492.8	78.1	75.0	29.4	0	0	0	489.4	25.1	23.8	8.3	0	0	0	327.8	91.5	91.0	43.9	0	0	0	465.6	85.4	82.7	25.1	0	0	0
Math-2	664.0	100.0	100.0	98.6	.5	0	0	45.0	100.0	100.0	81.3	0	0	0	374.8	99.4	99.3	90.7	0	0	0	415.9	99.7	94.8	80.1	.3	0	0
Math-39	499.8	29.6	0	7.1	0	0	0	11.0	0	0	0	0	0	0	440.9	90.8	70.9	35.6	0	0	0	414.9	29.6	0	.2	0	0	0
Math-44	491.6	32.5	8.6	11.1	0	0	0	11.3	0	0	0	0	0	0	432.8	73.3	64.4	41.4	0	0	0	430.7	15.1	3.2	4.3	0	.5	0
Math-52	487.6	62.0	32.6	9.5	0	0	0	62.4	75.1	47.9	63.6	.8	0	0	415.0	98.7	91.5	82.7	0	0	0	427.2	86.6	75.0	60.4	0	0	0
Math-56	487.6	21.4	15.6	10.1	0	0	0	40.3	95.7	97.9	80.4	1.0	0	0	428.5	99.7	98.4	74.3	1.0	0	0	495.1	94.5	81.7	48.3	.6	0	0
Math-64	488.6	10.2	0	.1	0	0	0	36.5	28.2	15.1	3.2	0	0	0	364.4	76.9	64.7	23.9	.8	0	0	431.3	5.4	0	0	0	0	0
Math-67	489.8	0	0	0	0	0	0	497.6	76.1	61.2	38.7	1.0	0	0	448.8	90.0	97.0	59.3	1.0	0	0	463.3	68.3	52.9	35.7	.3	0	0
Math-7	490.3	29.1	6.8	10.9	0	0	0	12.1	0	0	0	0	0	0	437.4	77.5	63.7	31.5	0	0	0	433.1	18.3	3.7	5.0	0	.3	0
Math-88	668.7	10.6	4.4	1.0	0	0	0	80.9	10.6	4.4	1.0	0	0	0	400.6	96.0	96.1	57.0	0	0	0	425.9	63.8	56.2	34.3	0	0	0
Math-91	527.2	90.1	78.8	62.3	0	.2	0	40.4	93.2	83.3	61.9	0	0	0	347.7	97.8	98.5	69.9	0	0	0	419.4	93.9	84.0	66.7	0	0	0
Math-93	575.6	78.1	74.5	63.7	0	0	0	515.8	36.0	30.3	21.5	0	0	0	469.3	87.5	85.9	68.7	.1	0	0	538.2	79.0	74.9	54.1	0	0	0
Time-10	492.5	86.9	90.5	46.0	0	0	0	8.3	0	0	0	0	0	0	354.8	91.1	92.3	63.4	.1	0	0	417.8	84.5	76.8	67.6	0	0	0
Time-11	492.4	20.2	15.5	6.5	0	0	0	79.1	57.9	46.2	19.6	0	0	.1	563.7	34.0	30.6	14.5	.6	0	0	441.2	42.7	38.5	20.6	0	0	0
Time-13	491.9	81.2	68.6	35.9	1.0	0	0	489.0	62.1	46.3	24.0	0	0	0	462.6	72.7	54.5	30.9	.1	0	0	505.9	95.8	88.4	56.2	1.0	0	0
Time-20	492.6	75.1	62.4	25.8	.1	0	.2	262.1	83.8	75.4	38.3	0	0	0	468.8	76.9	60.8	39.4	0	0	0	484.4	85.8	76.8	50.3	0	0	0
Time-23	493.8	78.4	66.0	45.0	0	0	0	9.2	0	0	0	0	0	0	401.5	85.3	77.4	51.7	0	0	0	443.5	77.7	66.6	42.4	0	0	0
Time-3	493.7	78.3	79.6	55.3	.6	0	.1	376.4	91.3	87.5	90.5	1.0	0	.1	448.8	95.4	96.8	57.5	.1	0	0	428.6	86.6	87.0	22.2	0	0	0
Time-4	492.6	69.0	57.0	24.0	0	0	.2	121.5	98.8	98.2	78.9	1.0	0	0	435.5	97.7	93.5	73.8	.3	0	0	443.2	93.9	92.2	58.1	.1	0	.1
Time-5	493.1	75.5	58.8	3.2	0	0	0	489.2	99.4	96.8	78.8	1.0	0	0	464.8	98.9	93.2	82.2	1.0	0	0	434.3	97.1	93.4	66.4	.5	0	0
Time-7	493.5	86.9	79.7	25.5	0	0	0	148.9	38.4	25.0	12.9	0	0	0	436.1	91.6	87.5	59.2	0	0	0	446.0	89.3	73.6	40.4	0	0	.1
Time-8	492.6	76.4	65.7	45.1	.6	0	0	8.8	0	0	0	0	0	0	399.4	84.8	77.6	53.8	1.0	0	0	438.6	75.3	62.1	37.8	.1	0	0