Experimental Results Report

1 Introduction

This report presents the experimental results of test suite generation based on different coverage criteria. The objective is to evaluate the fault-exposing potential of each test suite for various benchmark programs.

2 Experimental Results

The following table summarizes the size of each test suite and the number of faults exposed by it:

Program	Test Suite	Size	Faults Exposed
printtokens2	branch_add	4	2
printtokens2	statement_random	14	2
printtokens2	branch_random	17	3
printtokens2	statement_add	4	1
printtokens2	branch_total	6	6
printtokens2	statement_total	19	1
printtokens2	universe	4057	9
totinfo	branch_add	5	13
totinfo	statement_random	1	1
totinfo	branch_random	7	13
totinfo	statement_add	1	1
totinfo	branch_total	5	13
totinfo	statement_total	1	1
totinfo	universe	1026	21
schedule	branch_add	7	5
schedule	statement_random	11	1
schedule	branch_random	14	3
schedule	statement_add	4	1
schedule	branch_total	8	3
schedule	statement_total	7	3
schedule	universe	2634	9

tcas	branch_add	11	13
tcas	statement_random	6	9
tcas	branch_random	11	10
tcas	statement_add	4	10
tcas	branch_total	13	13
tcas	statement_total	7	10
tcas	universe	1590	41
printtokens	branch_add	6	4
printtokens	statement_random	17	1
printtokens	branch_random	21	3
printtokens	statement_add	6	3
printtokens	branch_total	7	5
printtokens	statement_total	23	2
printtokens	universe	4072	7
schedule2	branch_add	5	2
schedule2	statement_random	12	3
schedule2	branch_random	11	7
schedule2	statement_add	3	3
schedule2	branch_total	7	5
schedule2	statement_total	7	3
schedule2	universe	2679	9
replace	branch_add	11	4
replace	statement_random	18	4
replace	branch_random	24	4
replace	statement_add	8	1
replace	branch_total	21	8
replace	statement_total	21	4
replace	universe	5542	16

3 Observations

From the experimental results, we can make the following observations:

- The sizes of the generated test suites vary significantly across different programs and coverage criteria. For example, the `branch_add` test suite for `printtokens2` contains only 4 test cases, while the `universe` test suite for the same program contains 4057 test cases.
- Branch coverage criteria seem to be more effective at exposing faults compared to statement coverage criteria for certain programs. For instance, the `branch_total` test suite for `printtokens2` exposes 6 faults, which is the highest among all test suites for that program.
- Compared to the original `universe` test suite, the generated test suites can expose a similar number of faults with significantly fewer test cases. For example, the `branch_total` test suite for `tcas` exposes 13 faults with

only 13 test cases, while the `universe` test suite exposes 41 faults with 1590 test cases.

- The `statement_random` and `statement_add` criteria are generally the least effective at exposing faults, as they consistently expose fewer faults across different programs.

4 Conclusion

The results suggest that coverage-based test suite generation can effectively reduce the size of test suites while still maintaining a good fault detection capability. Branch coverage criteria seem to be more effective than statement coverage criteria in exposing faults. However, the effectiveness of different coverage criteria can vary depending on the specific program being tested.