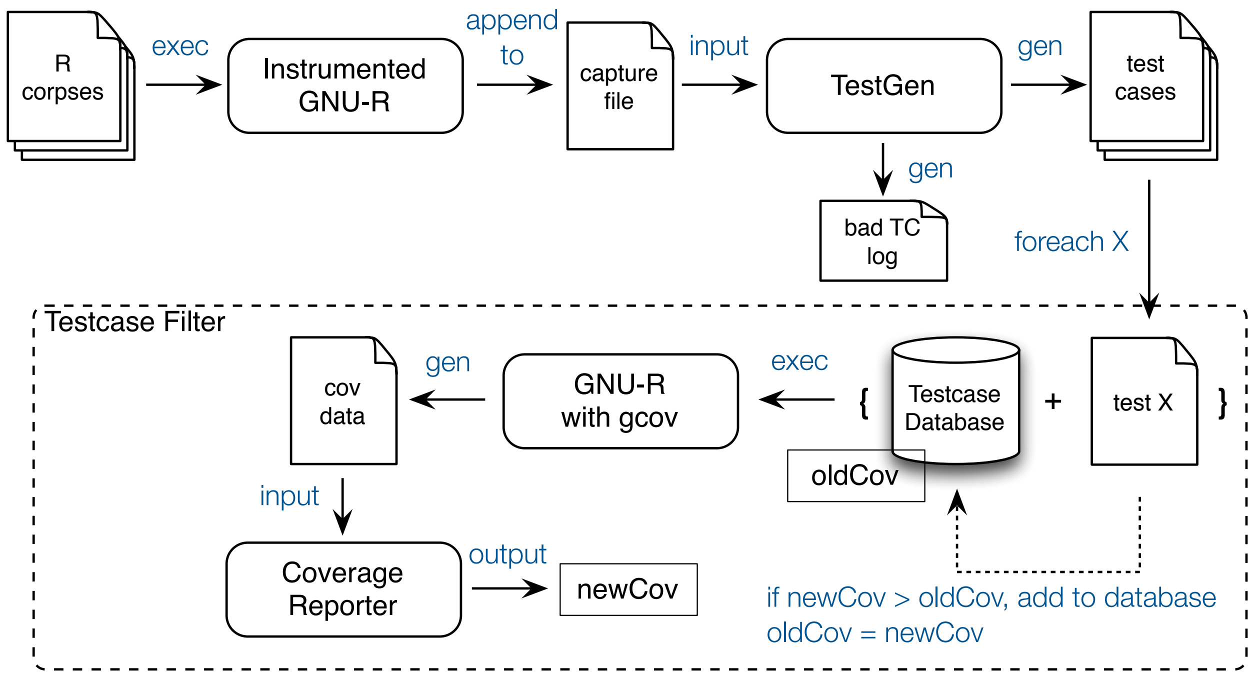
TestR/Capturer Overview

The Capturer is part of TestR system, focusing on automatically generating testcases by extracting and utilizing the use cases of functions in legacy R programs.

Infrastructure

The architecture of Capturer is presented in the figure below:



Instrumented GNU-R

The VM captures all builtin function call events by logging the function name, the argument list, and the return value. At the end of the execution, the logged information will be appended to a file named “capture” located under a user specified directory. The arguments and return value will be deparsed into strings. A single deparsed string will be replaced with an invalid argument indicator if it 1) contains more characters than allowed or 2) spans more lines than allowed. Currently, we set the limit of characters to be 50K and limit of lines to be 5. If the original call terminated with an error, the error will be logged as well.

The entry of the capture file has the format:

```
func: function_name
type: P | I  // P: primitive call; I: internal call
args: list(string1, string2, ..., stringN) | <arguments too long, ignored>
retv: string | <return value too long, ignored> | <error>
```

* Repo: allr/r-instrumented/branch:capture

TestGen

TestGen processes the capture file, generates all valid testcases, and logs all invalid testcases. TestGen runs on a trusted VM; it testruns each testcase being generated and compares the return value with the captured one. The testcase is flagged invalid if there is a mismatching.

The valid testcases will be stored in the file system as:

```
output.dir
  |- tc_function1/tc_1, tc_2, ...
  |- tc_function2/tc_1, tc_2, ...
  |- ...
  |- tc_functionN/tc_1, tc_2, ...
```

Following are the formats one would expect to see in testcases, one for primitive function, one for internal:

```
# primitive
argv <- eval(parse(text='list(string1, string2, ..., stringN)'));
do.call("function_name", argv);

# internal
argv <- eval(parse(text='list(string1, string2, ..., stringN)'));
.Internal(function_name(argv[[1]], argv[[2]], ..., argv[[N]]));
```

* Repo: allr//testr/testgen.r

Testcase Filter

The responsibility of Filter is to make sure adding new testcases into the Database only if there will be a coverage increasement. It functions by selecting a testcase from a set, which could be produced by either TestGen or Testr.R, runs on GNU-R with gcov a combination of the selected testcase and the Database, invokes the Coverage Reporter for a new coverage rate, and adds the testcase into the Database if there is a raise. Filter is going to a glue of a couple of components including testR harness, GNU-R with gcov, and Coverage Reporter.

* Repo: to be implemented.

GNU-R with gcov

GNU-R built with gcov support. Please refer to allr/testr/README.md for a how-to.

Coverage Reporter

The Coverage Reporter takes the output from GNU-R with gcov and summarizes the result. It reports the coverage on a variety of granularities, including file, function, and line. For detailed explanation, more function instroduction, and sample report, please look at the comments in the source file.

* Repo: allr/testr/coverage.r