TestR

Generating unit tests for R internals

Roman Tsegelskyi, Jan Vitek

Purdue University
https://github.com/allr/testR

Motivation

```
R1 > source('~/GNU-Rs/R1/tests/arith-true.R')
.
[1] TRUE
[1] TRUE
[1] TRUE
[1] TRUE
[1] TRUE
[1] TRUE
[1] TRUE
[1] TRUE
[1] TRUE
Time elapsed:  0.428 0 0.426 0 0
Warning messages:
1: In log(-1) : NaNs produced
2: In gamma(0:-47) : NaNs produced
3: In digamma(x) : NaNs produced
4: In psigamma(x, 0) : NaNs produced
```

Motivation

- Ensuring correctness of builtin functions written in C (More than 600)
- Automating this by generating test cases
- Generalize it to testing any R function

TestR

```
foo <- function (x) {
     x * 2;
}
```

```
R1 > foo(2)
[4]
```

TestR

- A test is a call to a test function with arguments to handle errors, warnings, etc.

```
test(id=0, code={
    foo <- function (x) {
      x * 2
    }
    foo(2)
}, o=4);
```

- Handles not only unit tests but also more complex test types

TestR (continued)

- Test cases can be generated from a template..

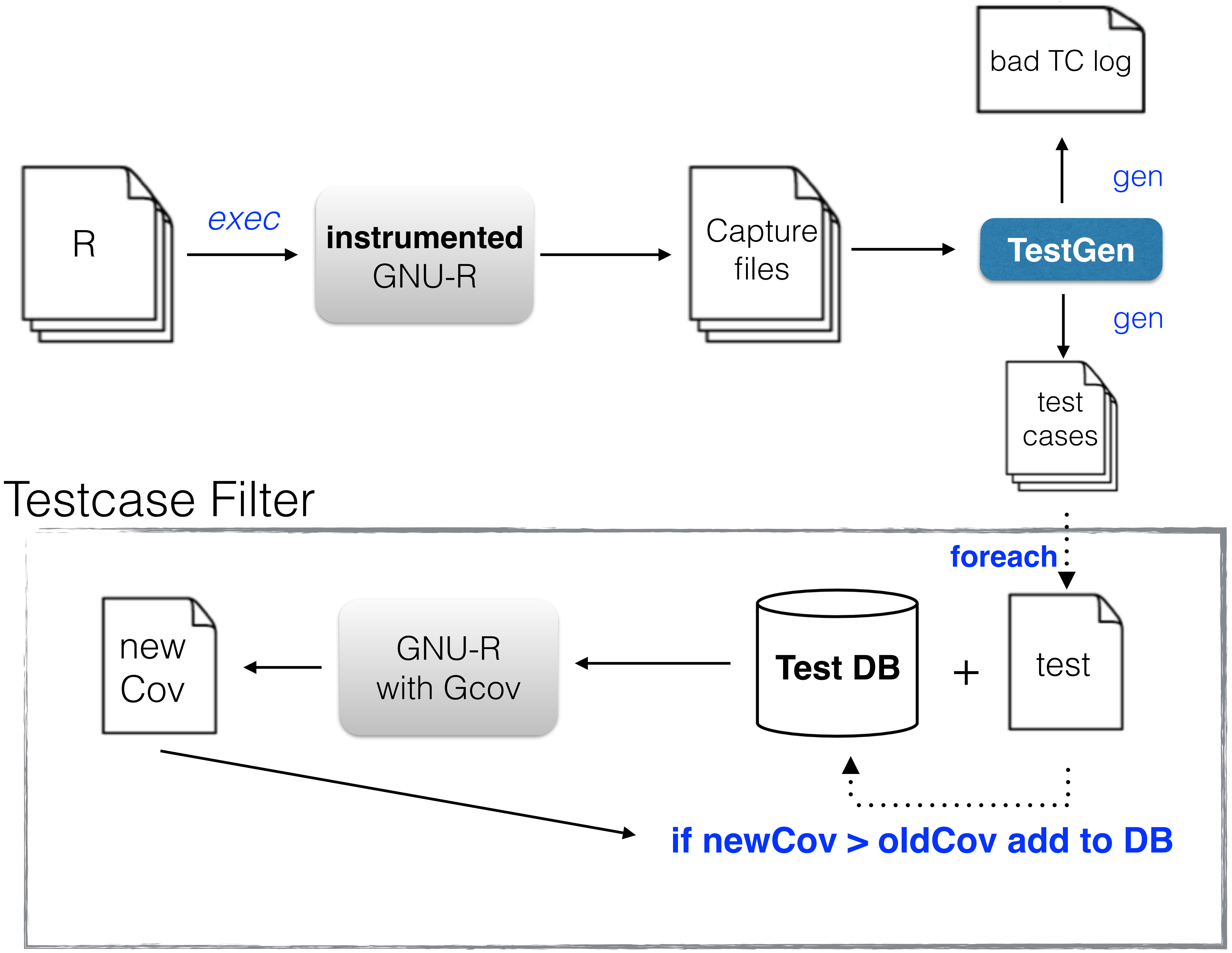
```
test(name = "foo",
  g(a, 1, 2, 3, 4),
  g(b, c(1,2), c(2,3),
    c(3,4)),
  g(c, "+","-"),
  code={a %c% b}
)
```

```
test(id=18,
  1 + c(1, 2),
  name = "foo[a=1,b=c(1,2),
     c = \"+\"]",
  o = c(2, 3)
)
```

Examples

```
expected <- eval(parse(text="TRUE"));
test(id=0, code={
  argv <- eval(parse(text="list(c(-0.9, 1.0))"))
  do.call(`is.atomic`, argv)
}, o=expected);
```

```
expected <- eval(parse(text="1+0i"));
test(id=0, code={
  argv <- eval(parse(text="list(1, 0+0i)"));
  do.call(`+`, argv)
}, o=expected);
```


bad TC log
R
exec
instrumented GNU-R
Capture files
TestGen
gen
gen
test cases
Testcase Filter
foreach
new Cov
GNU-R with Gcov
Test DB
+
test
if newCov > oldCov add to DB

Instrumented GNU-R

```
# identical
func: identical
type: I
args: list("closure", "S4", TRUE,
 TRUE, TRUE, TRUE, FALSE)
retn: FALSE
```

```
#is.na
func: is.na
type: P
args: list(NA_integer_)
retn: TRUE
```

Instrumented GNU-R

```
func: function_name

type: P | I

args: list(s1, s2, … , sn)
    | <arguments too long, ignored>

retv: string
    | <return value too long, ignored>
    | <error>
```

Dependent calls to builtins

```
foo <- function(){
  Tfile <- file("test1", “w+")
  cat("abc\ndef\n", file = Tfile)
  readLines(Tfile)
}

foo <- function(){
   file.create(‘file.1’)
   file.create(‘file.2’)
   file.append(‘file.1’, file.2’)
}
```

```
expected <- eval(parse(text="NULL"));
test(id=0, code={
writeLines<-  function (text, con = stdout(), sep = "\n",
useBytes = FALSE)
  {
      if (is.character(con)) {
          con <- file(con, "w")
          on.exit(close(con))
      }
      .Internal(writeLines(text, con, sep, useBytes))
  }

argv <- eval(parse(text="list(c(\"[476] \\\"1986-02-12\\\"
\\\"1986-02-13\\\"),"file.1");
do.call(`writeLines`, argv);
}, o=expected);
```

Instrumented GNU-R

```
func: function_name
body: closure_code
args: list(string1, string2, ..., stringN) |
    <arguments too long, ignored>
retv: string |
   <return value too long, ignored> |
   <error>
```

TestGen

- Process the capture file, generate all valid tests, log invalid tests
- Run each test on trusted VM and validate the return value
- Generates TestR output

Filtering

- Tests only added to Database if coverage increase
- For builtins only measure coverage of src/main, but can be done for any folder in general
- Use gcov to measure coverage of C code
  (nothing for R coverage yet)

Experimental results

- GNU R test suite gives 73% coverage in src/main
- Capturing builtin calls gave 45% coverage.
- Test suite has 3803 test cases out of 37M candidates.
- Capturing closures that contain primitive calls gives 58% coverage and adds 892 tests

Errors in R VMs

- CXXR (C++ R) (University of Kent)
  - 8 failed test cases compared to R-2.15.1
  - 263 failed test cases compared to R-3.0.1
- Renjin (R on JVM)
  - 621 failed test cases compared to R-3.0.1
  - 12 NULL pointer exceptions and 15 class cast exceptions

Conclusions

- An infrastructure for automatically generating test cases from legacy R code
- Generate test suite covers 80% of GNU R test suite covers, while shrinking size to 4695 tests
- Infrastructure finds bugs in R VM implementations
- Infrastructure can be used for creating test cases for any functions in R packages