COM4506/6506: Testing and Verification in Safety Critical Systems

Dr Ramsay Taylor

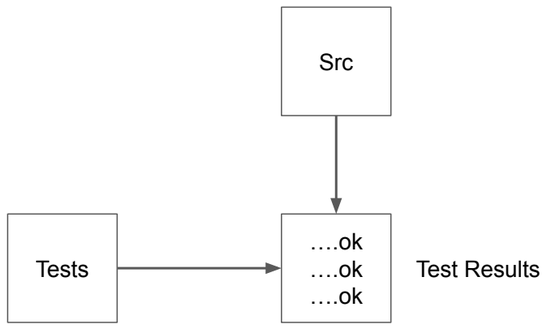
Contents

- Can I stop testing yet?
- Code Coverage as a test set metric
- Code coverage that is better than line/statement coverage

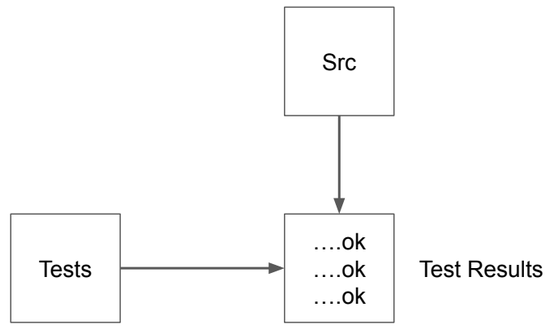
Test Adequacy

Test Adequacy

Test Adequacy

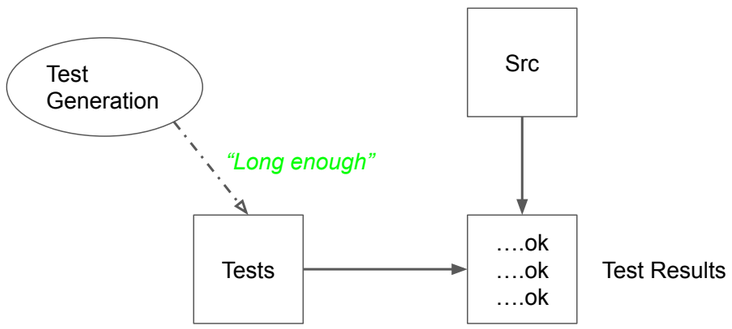


Test Adequacy

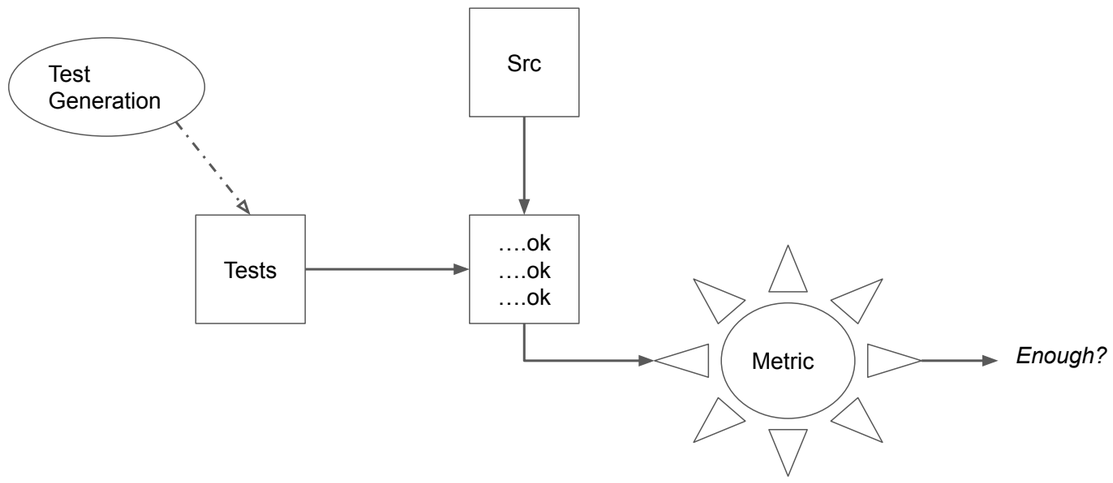


“Everything passes, its done, I’m going to the pub…”

Test Adequacy



Test Adequacy



Line/Statement Coverage

```
    |  -module(abiftest).
    |  -export([dv/2]).
    |
    |  dv(A,B) ->
0..|      if (A == 0) and (B > 4) ->
0..|           B;
    |         true ->
0..|           B / A
    |      end.
```

Line/Statement Coverage

```
    |  -module(abiftest).
    |  -export([dv/2]).
    |
    |  dv(A,B) ->
1..|      if (A == 0) and (B > 4) ->
1..|           B;
    |         true ->
0..|           B / A
    |      end.
```

dv(0,5)

Line/Statement Coverage

```
    |  -module(abiftest).
    |  -export([dv/2]).
    |
    |  dv(A,B) ->
2..|      if (A == 0) and (B > 4) ->
1..|           B;
    |         true ->
1..|           B / A
    |      end.
```

dv(0,5)
dv(5,5)

Line/Statement Coverage

```
    |  -module(abiftest).
    |  -export([dv/2]).
    |
    |  dv(A,B) ->
2..|      if (A == 0) and (B > 4) ->
1..|           B;
    |         true ->
1..|           B / A
    |      end.
```

dv(0,5)
dv(5,5)
dv(0,2)

** exception error: an error occurred when evaluating an arithmetic expression
in function abiftest:dv/2
(abiftest.erl, line 8)

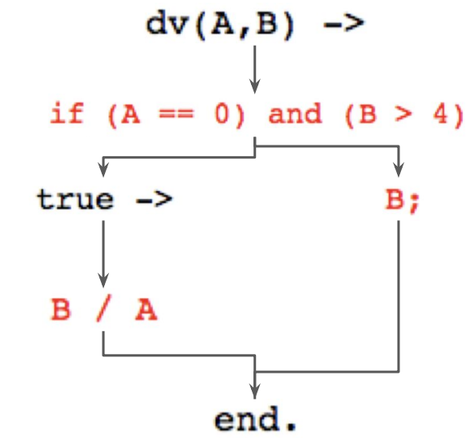
Line/Statement Coverage

- **Line coverage (or statement coverage) is a terrible metric for tests**
  - It only shows that you ran something *at all*, it tells you nothing about the circumstances

- **You should still achieve line coverage!** *Not even trying* bits of code would be bad
  - Line coverage is really easy to measure in most language environments

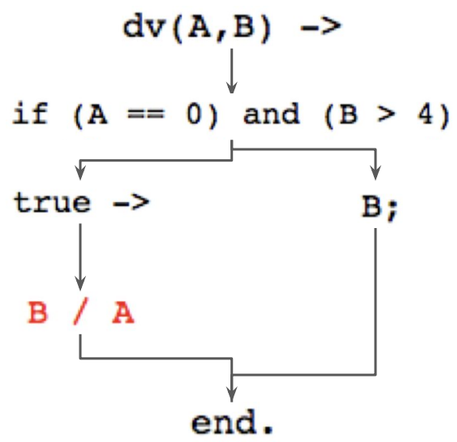
Branch Coverage

We can do better...



Branch Coverage

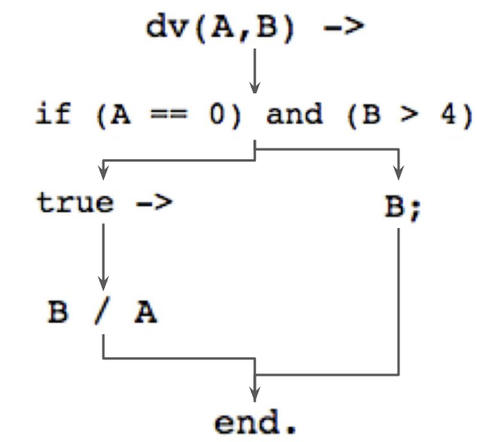
We can do better...



dv(0,5)

Branch Coverage

We can do better...

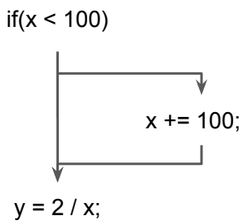


dv(0,5)
dv(5,5)

Branch Coverage

- Branch coverage doesn't help in that example (partly because its in Erlang!) but it can be useful in other cases.
- If statements without else statements are an obvious example.

```
if(x < 100) {
    x += 100;
}
y = 2 / x;
```



Condition Coverage

It still doesn't discuss *how* we got to this branch...

```
(A == 0) and (B > 4)
```

*This **decision** has two **conditions***

Condition Coverage

Condition coverage requires we try all evaluations for all *conditions*

```
(A == 0) and (B > 4)
```

dv(0,5)
dv(5,5)
dv(5,0)

A != 0	dv(5,5), dv(5,0)
A == 0	dv(0,5)
B =< 4	dv(5,0)
B > 4	dv(0,5), dv(5,5)

Condition Coverage

Condition coverage requires we try all evaluations for all *conditions*

```
(A == 0) and (B > 4)
```

dv(0,5)
dv(5,5)
dv(5,0)

We have both Branch and Condition Coverage, but we've still not hit the fault!

*Although we have made the Decision True and False, we haven't looked at all the **ways** to make it True and False using its conditions*

Modified Condition/Decision Coverage

Modified Condition/Decision Coverage

[...], every decision in the program has taken all possible outcomes at least once, and each condition in a decision has been shown to independently affect that decision's outcome. A condition is shown to independently affect a decision's outcome by varying just that condition while holding fixed all other possible conditions.

DO-178B [precursor to DO-178C]

Modified Condition/Decision Coverage

(A == 0) and (B > 4)

- matched: 1
- non-matched: 2

When false:

	matched	non-matched
A == 0	0	2
B > 4	1	1

dv(0,5)
dv(5,5)
dv(5,0)

Modified Condition/Decision Coverage

```
    |  -module(abiftest).
    |  -export([dv/2]).
    |
    |  dv(A,B) ->
2..|      if (A == 0) and (B > 4) ->
1..|           B;
    |         true ->
1..|           B / A
    |      end.
```

dv(0,5)
dv(5,5)
dv(5,0)
dv(0,2)

Summary

- We need some kind of metric for our test sets
  - To tell us whether they are any good
  - To tell us whether we can stop expanding them!
- Simply asking "does it actually run all the lines of code?" is a dreadful metric, but we should do at least that much!
- Looking at *how* we reached the code sections is important.
- MC/DC coverage is required by many Safety Critical Standards.