An Industry Proof-of Concept Demonstration of Automated Combinatorial Test

Redge Bartholomew

Rockwell Collins

Software Defects Drive Development Cost

…Especially for safety-critical, embedded systems

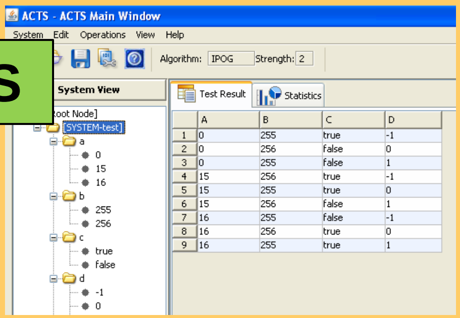
- NIST/NASA study looked at 15 years of defects
  - Avionics; medical, space, security systems; servers, browsers
- 6-way combinatorial testing could detect most of them
- NIST/UT-Arlington created a toolset to automate this
- Industry proof-of-concept experiment used them

NIST/UT-Arlington Approach

User provides test vector generator with input variable definitions & values

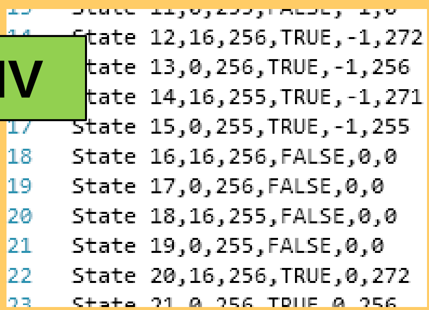
Tool generates minimum set of input test vectors,

ACTS

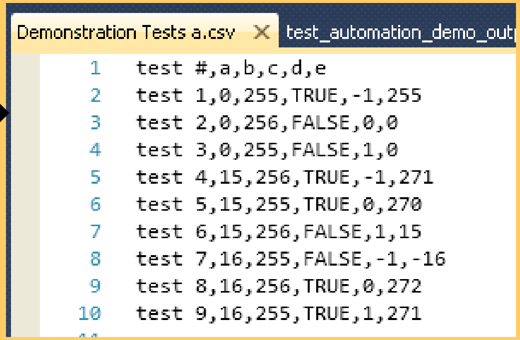


… model checker determines expected outputs for each vector,

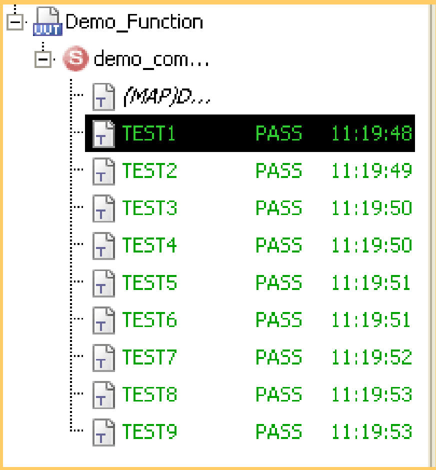
NuSMV



… script merges input vectors with their expected outputs,



… test harness imports, executes, analyzes test cases; identifies failures; measures coverage .



User provides model of inputs, outputs, properties

Proof-of-Concept Experiment

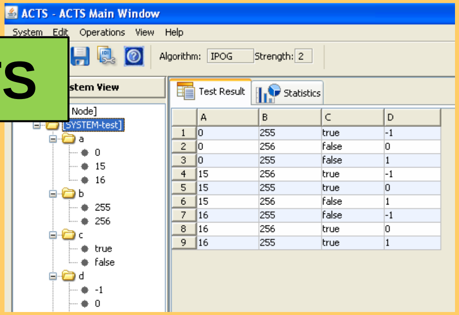
… reduce the number of defects escaping into system test

- Assess potential cost-effectiveness for unit, integration test
  - Accuracy, coverage, scalability, maturity, ease of learning/using
- Verify parts of a software defined radio (C++, 196 KSLOC)
- Inputs/outputs defined from reverse engineering code

Proof-of-Concept Experiment

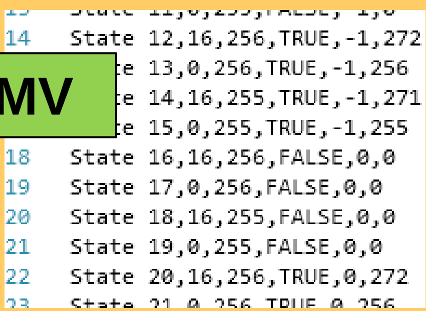
… expected output generation deviated from NIST approach

Tool generates minimum set of input test vectors;



ACTS

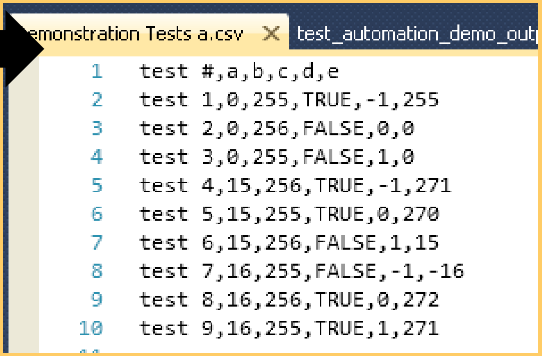
<u>… model checker generates the state space;</u>



NuSMV

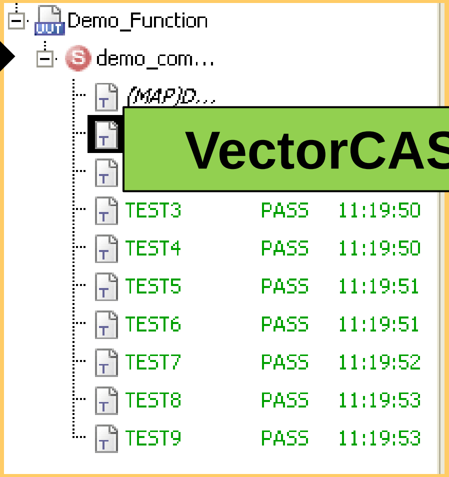
User provides test vector generator with input variable definitions & values

<u>… utility finds states containing input vectors, exports them as test cases;</u>



User provides model of inputs, outputs, <u>properties</u>

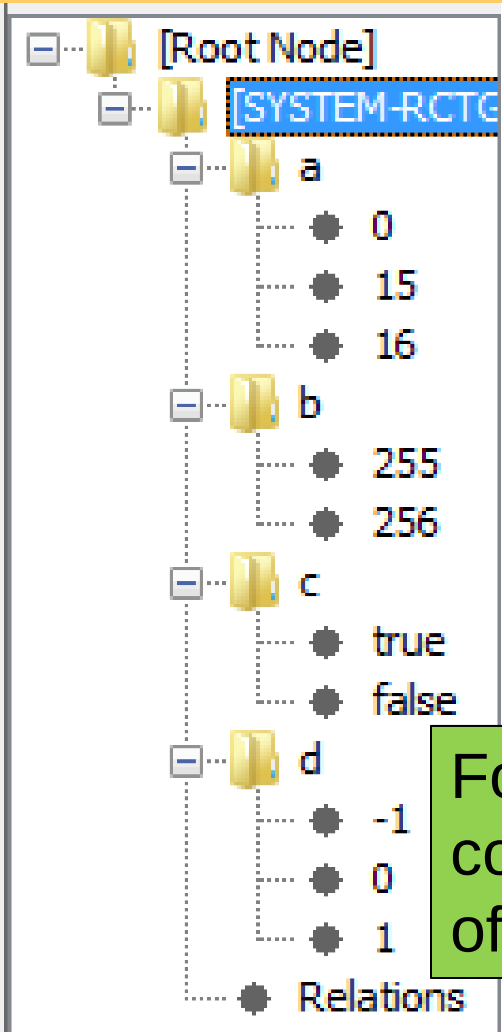
… test harness imports, executes, analyzes test cases; identifies failures; measures coverage.



VectorCAST

Defining the Input Space

… minimizing the vector set



	A	B	C	D
1	0	256	false	-1
2	0	255	true	0
3	0	256	true	1
4	15	255	true	-1
5	15	256	false	0
6	15	255	false	1
7	16	256	true	-1
8	16	255	false	0
9	16	256	true	1

For variables a, b, c, d, ACTS condenses all 37 2-way combinations of input values into 9 test vectors.

Rockwell Collins

Generating Expected Outputs

Developer models input-output relationships,

```
MODULE main
VAR
  a : {0,15,16};
  b : {255,256};
  c : {true,false};
  d : {-1,0,1};

DEFINE
  e :=
    case
      (c = true) : a + b;
      TRUE : a * d;
      esac;
```

For all computations of e,

… when c = true, e = a + b,

… otherwise, e = a * d

… the model checker generates the state space

```
  d = -1
-------- State   30 ------
  e = -15
  a = 15
  b = 255
  c = false
  d = -1
-------- State   31 ------
  e = 271
  a = 15
  b = 256
  c = true
  d = -1
```

Defining the Input Space

… to maximize defect detection & structural coverage

- More input values = better defect detection & coverage
  - … and greater risk of combinatorial explosion
- Equivalence classes can reduce the number of values
- Interaction testing can reduce the number of variables
  - Test only those that interact; others set to default values

Verifying the Expected Outputs

… debugging the model – e.g., coding & requirements errors

- 50 test vectors arbitrarily selected for manual verification.
- Model checker compared output model vs. properties
  - After gaining familiarity with tool, language, notation
- False positive/negative detection of seeded defects
  - Initially, most traced back to bugs in the expected output model

Accuracy & Scalability – Unit Test

… number of Input variables & values

- Radio control loop for normal mode: 11 inputs, 15 outputs
- Seeded 100+ errors in code across several versions
- Required interaction testing (avoid combinatorial explosion)
- 47,040 tests generated, executed, analyzed in ~2.6 hrs.
  - All defects detected; 98% branch coverage (~2 hrs.)

Accuracy & Scalability – Unit Test

… complex logic

- Mode control: 34 inputs, 4 outputs of interest
  - 39 if-tests nested up to 8-levels deep, embedded in a 6-case switch
  - 200+ seeded defects
- Used several sets of interaction tests (largest = 19 inputs)
- 2775 tests generated, executed, analyzed in ~1hour
  - All defects detected; 95% MC/DC required ~16 hours

Modified Condition/Decision Coverage

- Independence of outcome required writing test stubs
  - … to force the sequence of input vectors in multi-conditionals
  - … e.g., in loops that modify the control variable

MC/DC: Every condition in a decision
- … has taken all possible outcomes at least once,
- … and each condition independently affected the outcome

```
while ((current_radio_state != APPLICATIONS_RUNNING) &&
(current_radio_state != NO_WAVEFORM_AVAILABLE))
{
    current_radio_state = Radio_State();
}
```

Radio_State stub must return specific, successive values

Accuracy & Scalability - Integration Test

- Tested code unit via 6 levels of nested procedure calls
  - Intervening variables set to force execution along required path
- 30,905 tests detected seeded defects
  - Finding intervening variables (30+) & default values: several days
- Scalability for large/nested architecture requires mitigation

Ease of Learning & Using – 1st Test

… 1st test (complex logic): ~84 hrs. total vs. 80 hr. target

- Staff had no prior experience with tools or code
- Learning ACTS (~2 hrs.) + defining 34 inputs (~4 hrs.)
- Learning NuSMV (~20 hrs.) + Debugging model (~12 hrs.)
- Structural coverage: branch (~2 hrs.)
  - MC/DC: another 16 hrs. (includes defining/constructing stubs)

Rockwell Collins

Maturity

… adapted from DoD/NASA Technology Readiness Calculator*

- Level 7
  - Prototype software exists & all key functionality is available for demonstration/test
  - It is well integrated with operational systems
  - Operational feasibility has been demonstrated
  - Most software bugs have been eliminated
  - Some documentation is available

Technology Readiness Levels

1. Basic principles observed
2. Concept/application formulated
3. Experimental proof-of-concept
4. Validation in laboratory
5. Validation in a relevant environment
6. Prototype demonstration
7. **<u>Prototype demonstrated in [*operational environment*]</u>**
8. [Operationally] qualified via test
9. Proven via mission operations

* https://acc.dau.mil/CommunityBrowser.aspx?id=320594&lang=en-US

Summary

… effective enough for unit test

- Significant defect detection, coverage from moderate effort
- May play a role in integration test
- Finding equivalence class values requires experience
  - So does using model checker properties to find expected outputs
- Deployment issues: support, packaging, licensing, training

Acronyms

- ACTS – Advanced Combinatorial Testing System
- DoD – Department of Defense
- GAO – Government Accountability Office
- KSLOC – 1000’s of Source Lines of Code
- MC/DC – Modified Condition/Decision Coverage
- NASA – National Aeronautics and Space Administration
- NIST – National Institute of Standards and Technology
- UT-Arlington – University of Texas at Arlington