Software Testing Effectiveness

Nathaniel Crowell
October 27, 2009

Measuring Effectiveness

- Questions to consider before determining effectiveness:
  - What is a test case?
  - What is the output of a test case?
  - What is effectiveness for a set of test cases?
  - What effectiveness can be considered good?
  - Are there other factors besides effectiveness to consider?

Test Cases

- A test case is a set of inputs to the program.
  - To be deterministic, a test case has to represent all inputs including any state information (e.g., environment).
- The output of a test case is either success (the expected output) or failure (differs from the expected output).

Measures of Effectiveness

- P-Measure
  - Probability that at least one failure is detected by the set of test cases.
- E-Measure
  - Expected number of failures detected by the set of test cases.
- F-Measure
  - Expected number of test cases required to detect the first failure.

F-measure

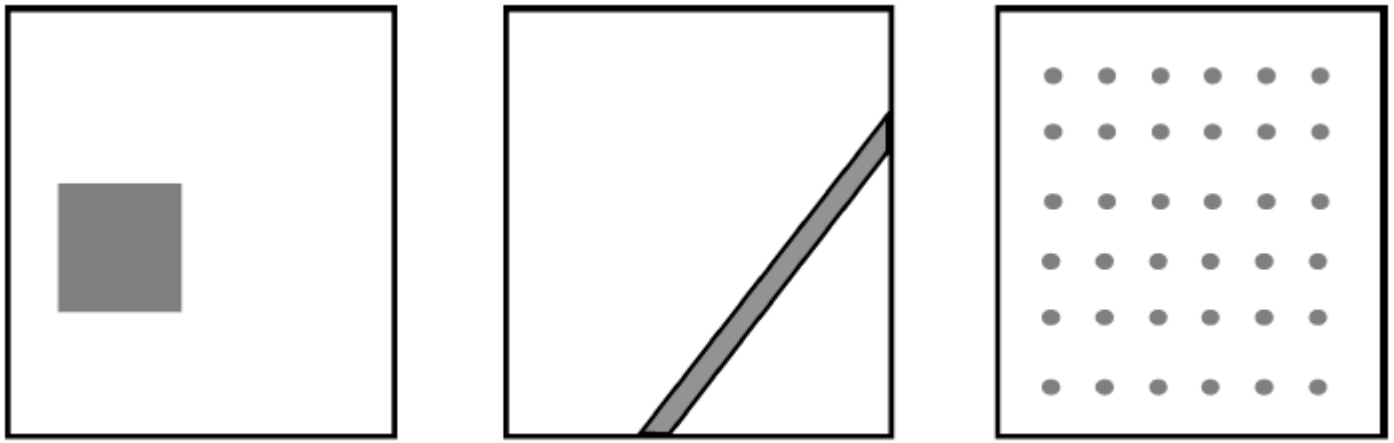
- Usefulness of P and E are straitforward.
- Primary assumption for the F-measure:
  - in debugging, only the first fault found is of immediate concern
  - is this a good assumption?
- Motivates a reduced number of test cases and therefore overall costs of testing.
- Order of test cases matters

Program Failures

- Failures occur for a subset of the possible input domain of the program.
- Generally, inputs that produce failures will cluster in regions of the input domain.
- Contiguous regions form patterns in the input domain.

Failure Patterns

- for two-dimensions: (a) block pattern, (b) strip pattern, (c) point pattern

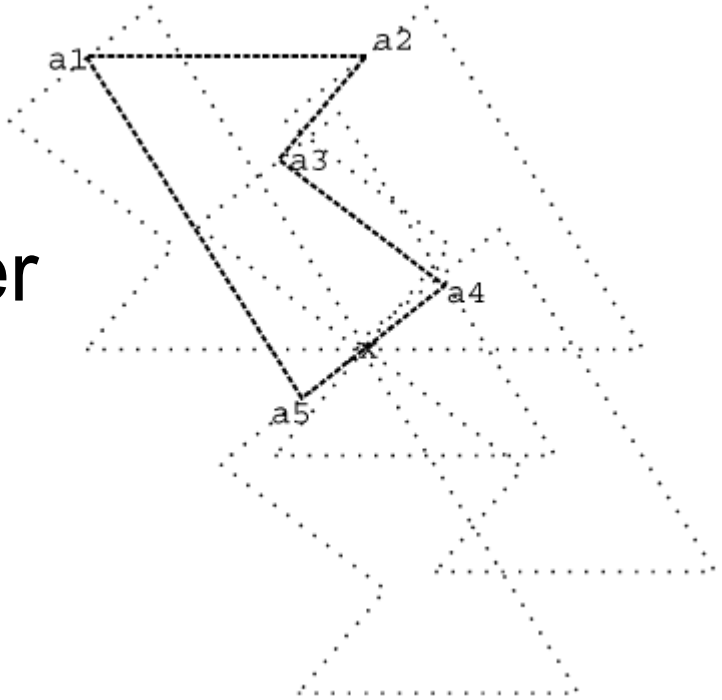


Proof of Effectiveness Upper Bound

- Assumptions:
  - failure region's size, shape, and orientation known.
  - region location in the input domain is not known.
- Do assumptions affect usefulness of the result?
- F-measure of a test set is calculated as the mean count of executed test cases before first failure over all possible locations for the failure region.

Proof (1)

- Each test case has an exclusion region that defines where the failure region may exist if the test case detects a fault.
- Probability of a test case detecting a failure is no greater than the ratio of areas for exclusion region and region of possible failure causing inputs



Proof (2)

- If test cases are generated in the input domain with non-overlapping exclusion regions, probability of detecting at least on failure grows linearly with the number of test cases.
- No other set of test cases can have a lower F-measure than a set generated in this way.
- If the failure region is small, the smallest possible F-measure in this case is one-half that of random testing.

Random Testing (with replacement)

- P-measure (p is failure rate)
  - 1-(1-p)^(1/p)
- E-measure (n is number of test cases)
  - n*p
- F-measure
  - Ratio of size of input domain and size of failure region.

Application of Proof

- Clearly, random black-box sampling of the input domain is a worst case.
- Some randomness may still be important as the size, shape, and orientation of failure regions should not always be known.
- Adaptive Random Testing is an alternative to random testing with replacement.

Adaptive Random Testing

- Technique to distribute test cases evenly within the input space without losing all randomness.
- Fixed-Size-Candidate-Set ART
  - At each iteration of test case generation, a set of candidates are randomly generated.
  - For each, the distance of the candidate from the set of previously executed tests is calculated
  - The candidate with the greatest distance is executed and added to the set of executed tests.

Results

- FSCS-ART has achieved result near the upper bound for effectiveness.
- However, the costs for test case generation are greater than random testing.

References

- Chen, T. Y. and Merkel, R. An upper bound on software testing effectiveness. ACM Trans. Softw. Eng. Methodol. 17, 3 (Jun. 2008), 1-27.
- Chen, T. Y., Kuo, F.-C., and Merkel, R. 2006. On the statistical properties of testing effectiveness measures. J. Syst. Softw. 79, 591–601.
- Chen, T. Y., Kuo, F.-C., and Merkel, R., AND NG, S. 2004. Mirror adaptive random testing. Inform. Softw. Tech. 46, 15, 1001–1010.
- Chen, T. Y., Leung, H., and Mak, I. K. 2004. Adaptive random testing. In Proceedings of the 9th Asian Computing Science Conference. Lecture Notes in Computer Science, vol. 3321. 320–329.