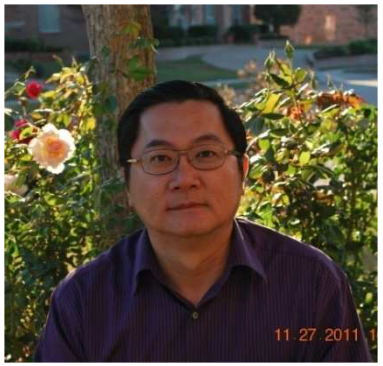
STAR Laboratory of Advanced Research on Software Technology

Adaptive Random Testing

W. Eric Wong
Department of Computer Science
The University of Texas at Dallas
ewong@utdallas.edu
http://www.utdallas.edu/~ewong

Speaker Biographical Sketch



- Professor & Director of International Outreach
  Department of Computer Science
  University of Texas at Dallas
- Guest Researcher
  Computer Security Division
  National Institute of Standards and Technology (NIST)
- Vice President, IEEE Reliability Society
- Secretary, ACM SIGAPP (Special Interest Group on Applied Computing)
- Principal Investigator, NSF TUES (Transforming Undergraduate Education in Science, Technology, Engineering and Mathematics) Project
  - *Incorporating Software Testing into Multiple Computer Science and Software Engineering Undergraduate Courses*
- Founder & Steering Committee co-Chair for the SERE conference (*IEEE International Conference on Software Security and Reliability*) (http://paris.utdallas.edu/sere13)

Basic Concepts

- Input domain: Set of all possible inputs
- Exhaustive testing:
  - Test the program with the entire input domain
  - Practically infeasible
- Failure-causing inputs: Inputs that exhibit failures

Random Testing (1)

- Random Testing
  - Selects test cases from the entire input domain randomly and independently
- Advantages:
  - Intuitively simple
  - Allows statistical quantitative estimation of the software's reliability

Random Testing (2)

- Two approaches
  - Uniform distributions
  - Operational distributions (profiles)

How to Improve Random Testing

- Any common information or characteristics to all faulty programs?

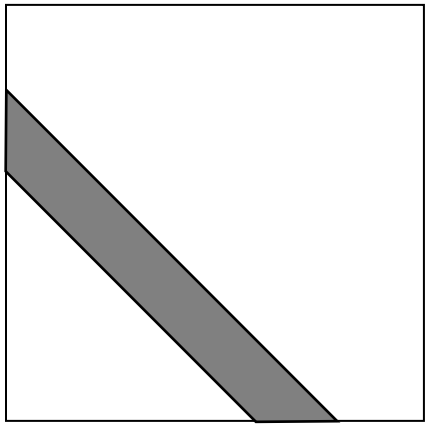
Failure-causing inputs

Patterns of Failure-Causing Inputs

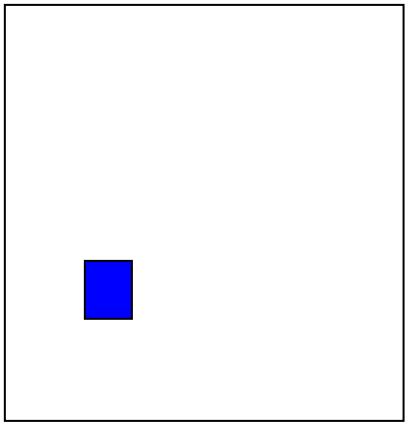
- Strip Pattern
- Block Pattern
- Point Pattern

Types of Failure Patterns

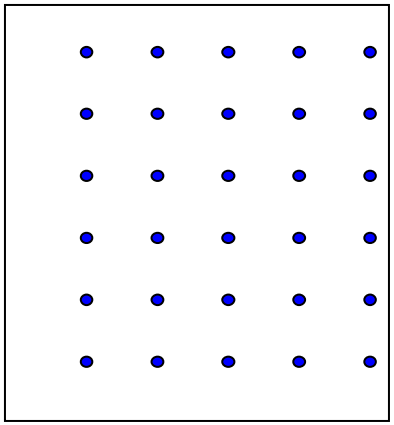
Strip Pattern



Block Pattern

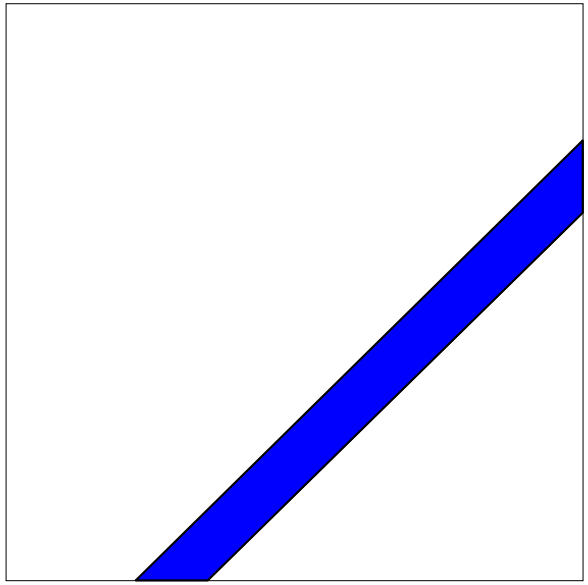


Point Pattern



Strip Pattern

Two Dimensional
Input Domain

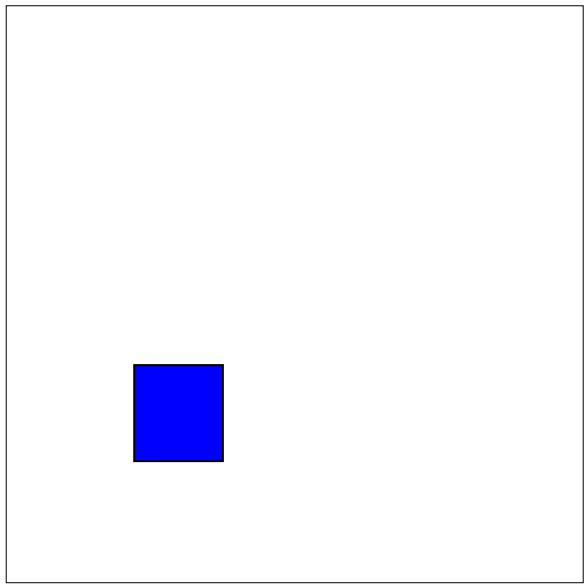


if ($2x - y > 10$)
/* The correct statement is if ($2x - y > 20$) */
then
    $z = x/2y$
else
    $z = xy$

A different type of error "if ($2x - y >= 10$)"

Block Pattern

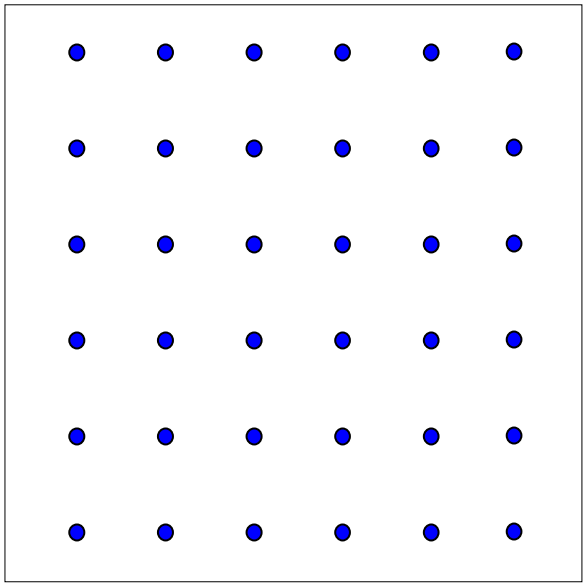
Two Dimensional
Input Domain



```
if ((x >= 4 and x <=6) and
    (y >= 4 and y <= 6))
then
    z = x + y
    /* The correct statement is z = x − y */
else
    z = 100
```

Point Pattern

Two Dimensional
Input Domain

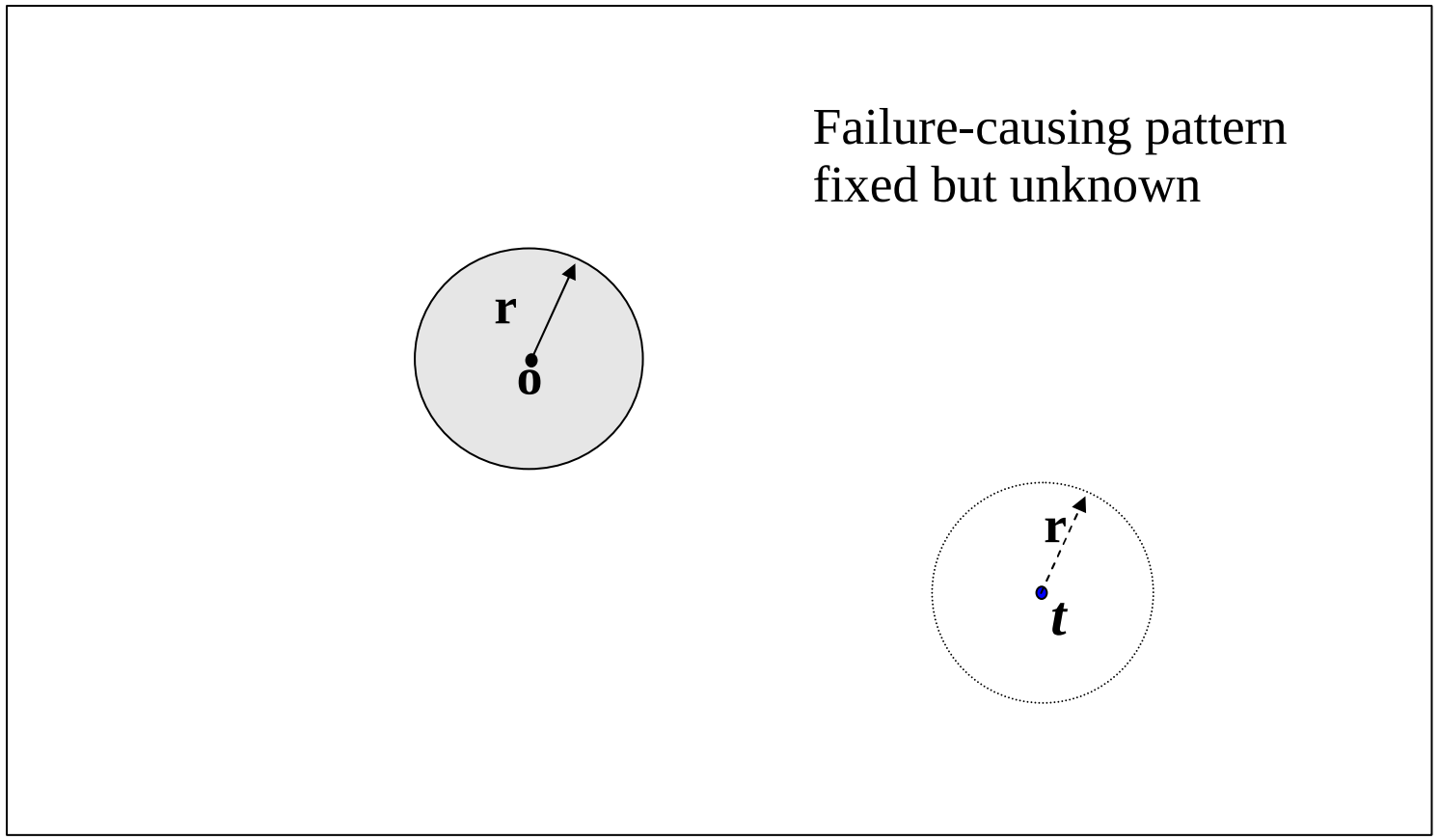


if (((*x* mod 10) = 0) and
((*y* mod 10) = 0))
then
$z = f(x,y)$
/* should be $z = g(x,y)$ */
else
$z = f(x,y)$

Which Pattern Occurs More Frequently?

Block and strip patterns

Intuition of ART



Adaptive Random Testing (1)

- For non-point failure patterns
  - An even spread of random test cases will enhance the fault detection capabilities

Adaptive Random Testing (2)

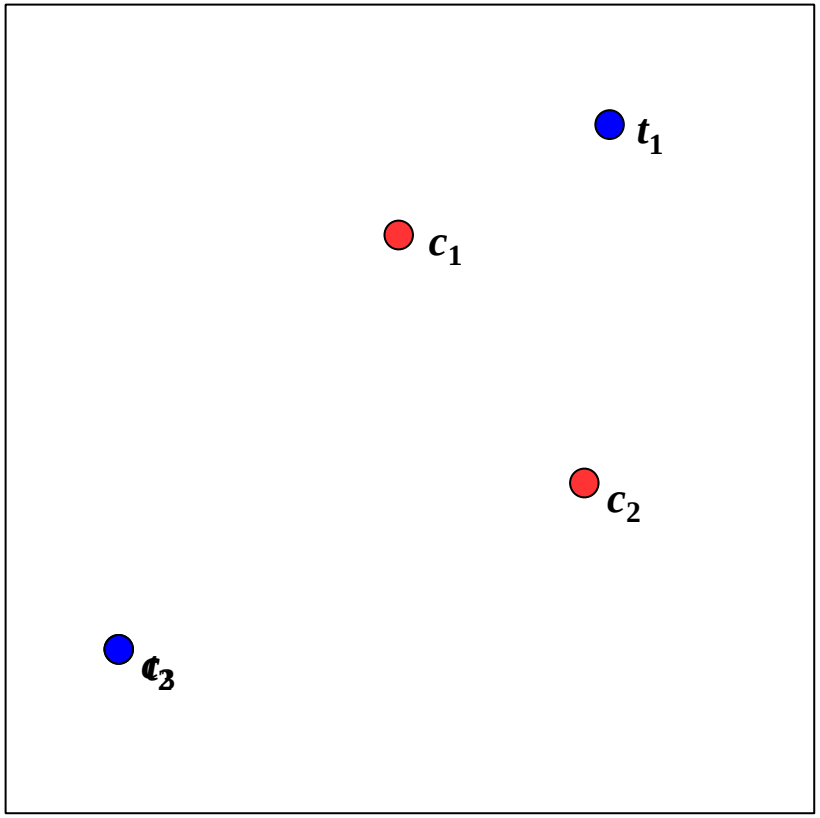
- Simulation and empirical results showed that as compared with random testing, fewer test cases required to detect *the first failure* (smaller *F-measure*)

- F-measure of ART ≅ 50-60% of that of RT with replacement

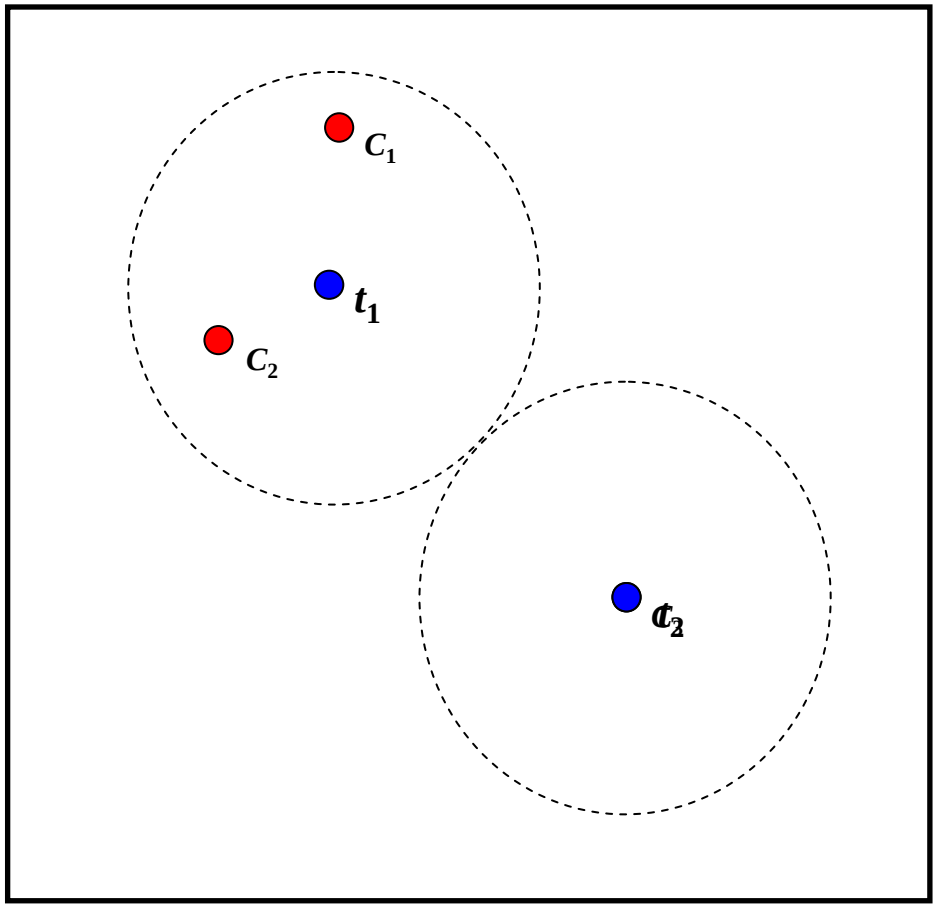
How to Achieve "Even Spread" ?

- Notion of distance
- Notion of exclusion
- . . . . . .

ART by Distance



ART by Exclusion



Even Spread Approaches

- Distance
  - Distance measures
  - Size of candidate set
  - . . . . . .
- Exclusion
  - Exclusion amount
  - Shape of exclusion region
  - . . . . . .

Possible topic for your term paper

ART versus RT