ON SUBDOMAINS: TESTING, PROFILES, AND COMPONENTS

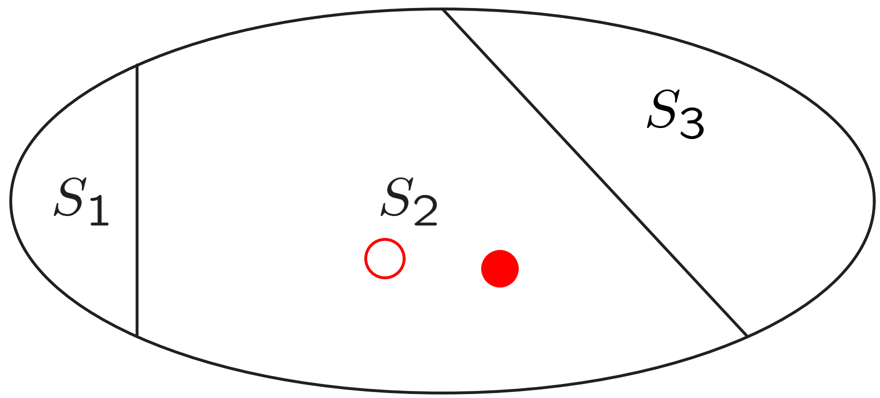
Dick Hamlet
Portland State University
Portland, OR, USA

Components work joint with

Dave Mason **Denise Woit**
Ryerson Polytechnic University
Toronto, Ontario, CANADA

Subdomains and Testing

A program input space can be decomposed into subdomains:



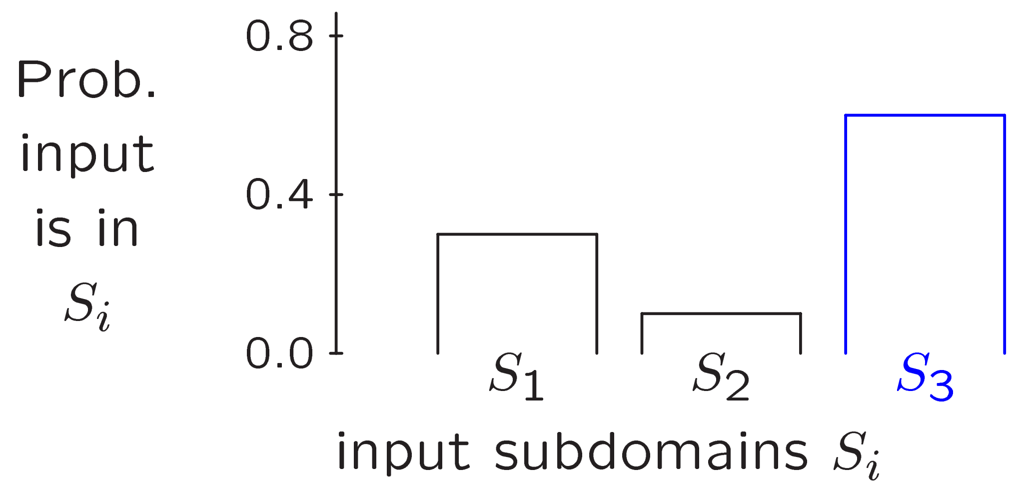
All non-random testing methods are subdomain methods:

- Choose subdomains whose points are "the same."
- Select one test point from each subdomain.
- For analyzing X, "the same" *must be relative to* X.

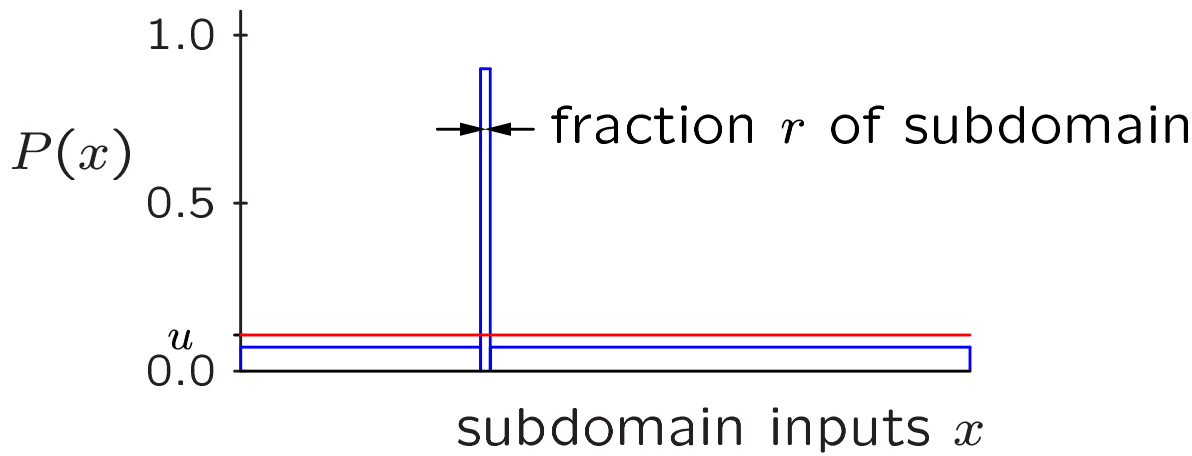
Operational Profile

Usage profiles are histograms described by a weighting vector:

profile $< .3, .1, .6 >$:

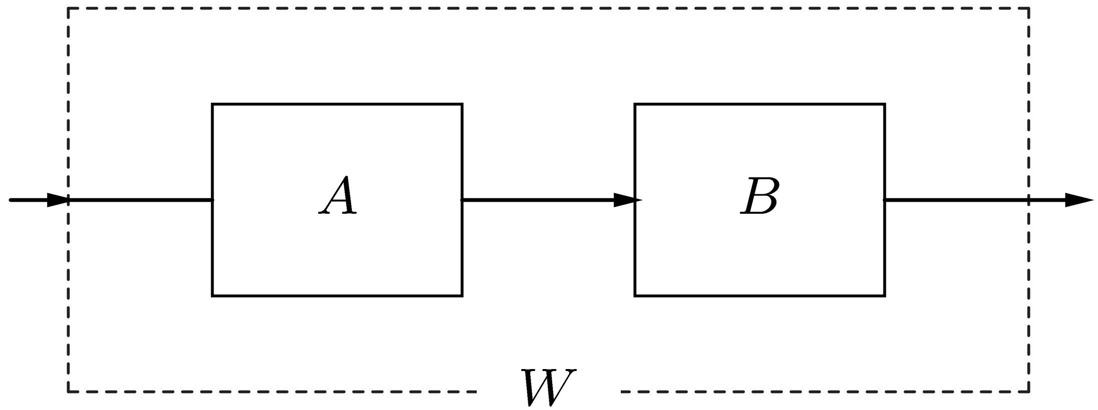


The Dreaded "Spiky" Profile



Failure rate:	$r \approx 0$	measured
	$1 - u \approx 1$	observed

Component Reliability Theory



Measurements on A and B at component development time allow calculation for W at system design time.

Component Datasheet Mappings

For n subdomains input profile $P =< h_1, h_2, ..., h_n >$.

Reliability Mapping Measure failure rates f_i in each subdomain. The component reliability is:

$$R = \sum_{i=1}^{n} h_i (1 - f_i).$$

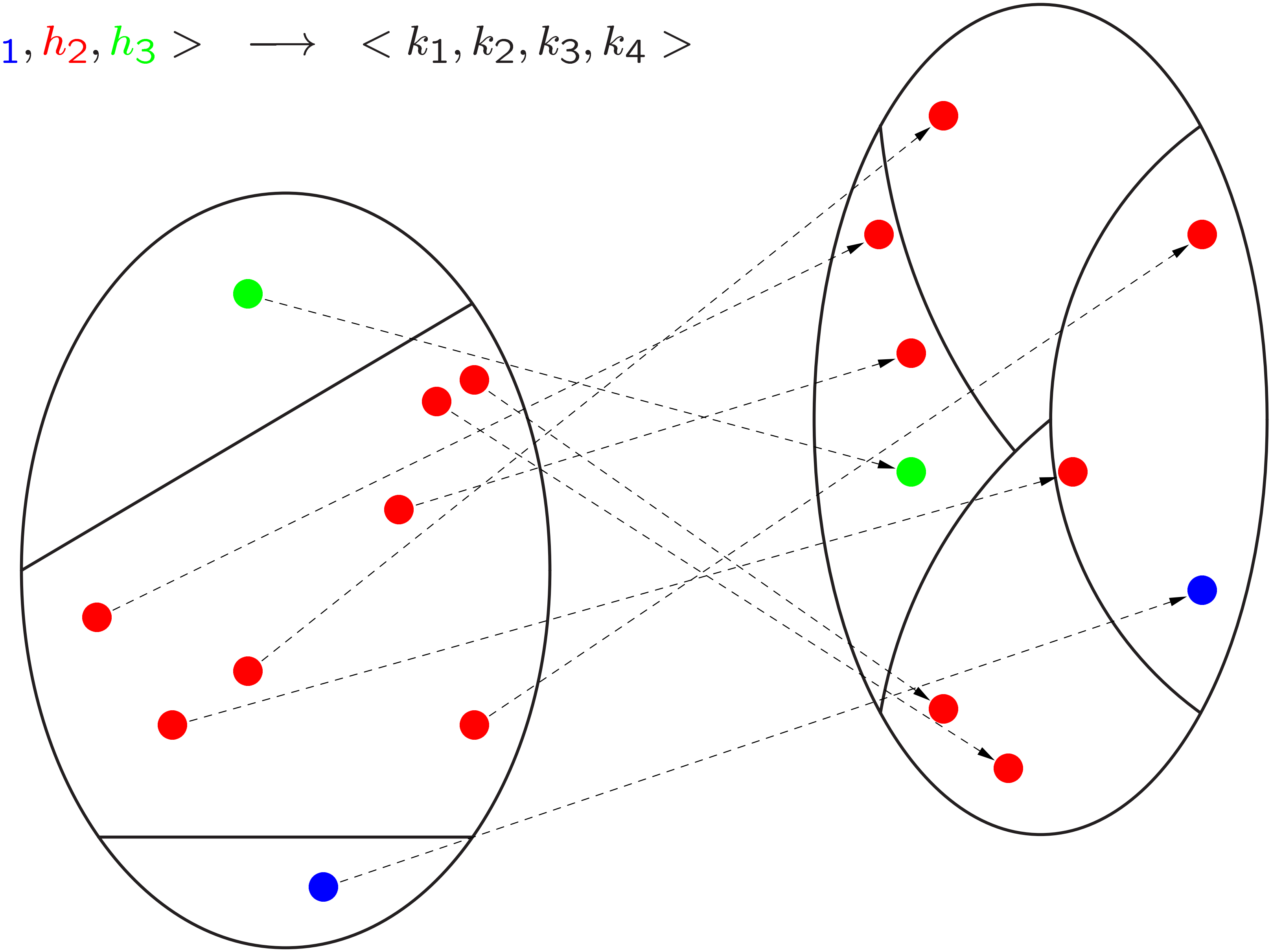
Profile-transformation Mapping For *arbitrary* output subdomains $U_1, U_2, ..., U_m$, the output profile $Q =< k_1, k_2, ..., k_m >$ is:

$$k_j = \sum_{i=1}^{n} h_i \frac{\left|\{z \in S_i | c(z) \in U_j\}\right|}{|S_i|},$$

where c is the component function.

Profile Transformation

$< h_1, h_2, h_3 > \quad \longrightarrow \quad < k_1, k_2, k_3, k_4 >$



Example System Calculation

Key: parameter measured calculated

Integers limited to $2^{16} - 1$.
A's function $c(x) = \sqrt{|x - 13|}$.
A's subdomains:

$$S_1 = \{n|n < 0\}, S_2 = \{0\}, S_3 = \{n|n > 0\}.$$

failure rates $f_1 = .01$, $f_2 = 0$, $f_3 = .001$.
Input profile to A: $< .3, .1, .6 >$.
Reliability of A alone:

$$.3(1 - .01) + .1(1 - 0) + .6(1 - .001) = .996$$

B's subdomains:

$$U_1 = \{n | n \leq 0\}, U_2 = \{n | 1 \leq n \leq 10\},$$

$$U_3 = \{n | 11 \leq n \leq 100\}, B_4 = \{n | n > 100\}.$$

failure rates .1, 0, 0, and .02 respectively.

Fraction of A outputs in B's subdomains:

Subdomain	from S_1	from S_2	from S_3
U_1	0	0	0
U_2	.003	1.0	.002
U_3	.147	0	.162
U_4	.850	0	.836

B input profile $< 0, .102, .141, .757 >$:

$$\begin{aligned} k_1 &= .3(0) + .1(0) + .6(0) = 0 \\ k_2 &= .3(.003) + .1(1.0) + .6(.002) = .102 \\ k_3 &= .3(.147) + .1(0) + .6(.162) = .141 \\ k_4 &= .3(.850) + .1(0) + .6(.836) = .757 \end{aligned}$$

Reliability of B alone:

$$0(1 - .1) + .102(1 - 0) + .141(1 - 0) + .757(1 - .02) = .986$$

System reliability (.996)(.986) = .982

Discussion: “The Same”

Functional (specification-based) subdomains *are* composed of “the same” points, but only if the program is correct.

Structural (program-based) subdomains are composed of “the same” points only in arcane program terms.

Intersecting subdomains is a good idea, but what subdomains should we start with? A failure model of real defects is needed.

Discussion: Continuity

The 'other' engineers can rely on the continuous properties of matter to interpolate between test points.

How can we use *specification* continuity?

How can we improve *program* continuity?