Information Theory in Software Testing

The Map

Some Places on the Map

Shannon Entropy

$$\mathcal{H}(X) = -\sum_{x \in X} p(x) \log_2 p(x)$$

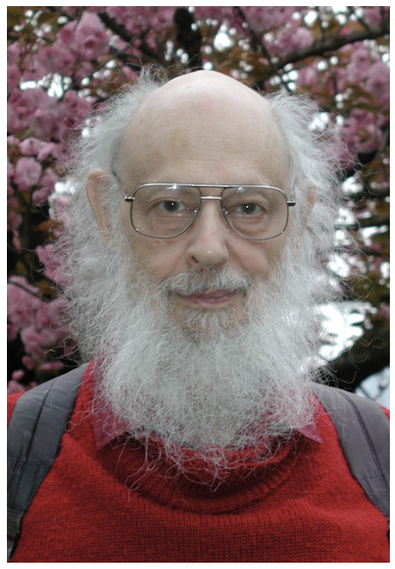
randomness of a
random variable

source

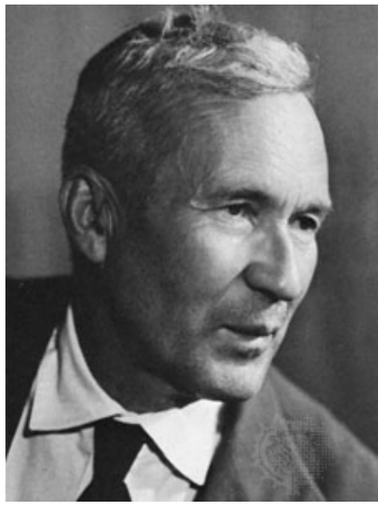
transmitted

expressions
statements
paths
programs

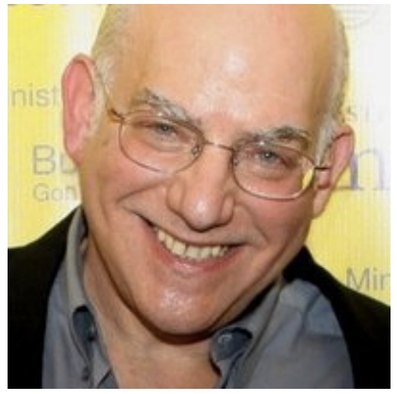
Kolmogorov Complexity



Solomonoff



Kolmogorov



Chaitin

The length of the shortest program that can produce a given string from no inputs

randomness of a string

NID: The Normalised Information Distance

For two strings x and y,

$$\mathrm{NID}(x,y) = \frac{\max\{K(x|y), K(y|x)\}}{\max\{K(x), K(y)\}}$$

Enables comparisons between strings of different lengths

NCD: The Normalised Compression Distance

For two strings x and y,

$$\mathrm{NCD}(x,y) = \frac{C(xy) - \min\{C(x), C(y)\}}{\max\{C(x), C(y)\}}$$

Computable approximation using compressors such as 7zip, Bzip

Input diversity

Idea

Use information theory to measure test set diversity

- Diversity $\longleftrightarrow$ Randomness $\longleftrightarrow$ Information
- Generic
- Universal

Kolmogorov complexity for input diversity

- Don't know probability distribution on inputs
- Test Set $\longleftrightarrow$ Set of Objects
- Minimise the similarity between the Objects
- Normalised Compression Distance for multisets (NCD) applied to test inputs (Input TSDm)

"Select diverse test cases"

- Inputs
- Outputs
- Execution Traces
- Combinations of these.

I-TSDm could be used before implementation with only partial specification or description

Output diversity

Channel Capacity of the program's I/O channel

$$\max_{\sigma \in \Sigma_{\mathcal{I}}} \mathcal{M}(\mathcal{I}; \mathcal{O})$$

$$\max_{\sigma \in \Sigma_{\mathcal{I}}} \mathcal{H}(\mathcal{O}) = \log_2(|\mathcal{O}|)$$

$\mathcal{H}(f^{-1}o)$

$f^{-1}o$

f

o

$p(o)$

Test Suite Selection

Output Diversity

1. Generate random inputs
2. Discard inputs if they lead to discovered outputs
3. Repeat until too hard to find new test inputs

High Pearson correlation with statement, branch and path coverage but 47% improvement in bug finding over these.

Alshawan and Harman, ICSE 2012, ISSTA 2014

Squeeziness causes problems

Loss of information from running program **P**

deterministic case

$$\mathcal{H}(I) - \mathcal{H}(O) \quad = \mathcal{H}(I|O)$$

where $[\![P]\!]I = O$

Conditional entropy of I given O: **Squeeziness**.

$$Sq(f) = \mathcal{H}(I) - \mathcal{H}(O) = \sum_{o \in O} p(o)\ \mathcal{H}(f^{-1}o)$$

via the partition property

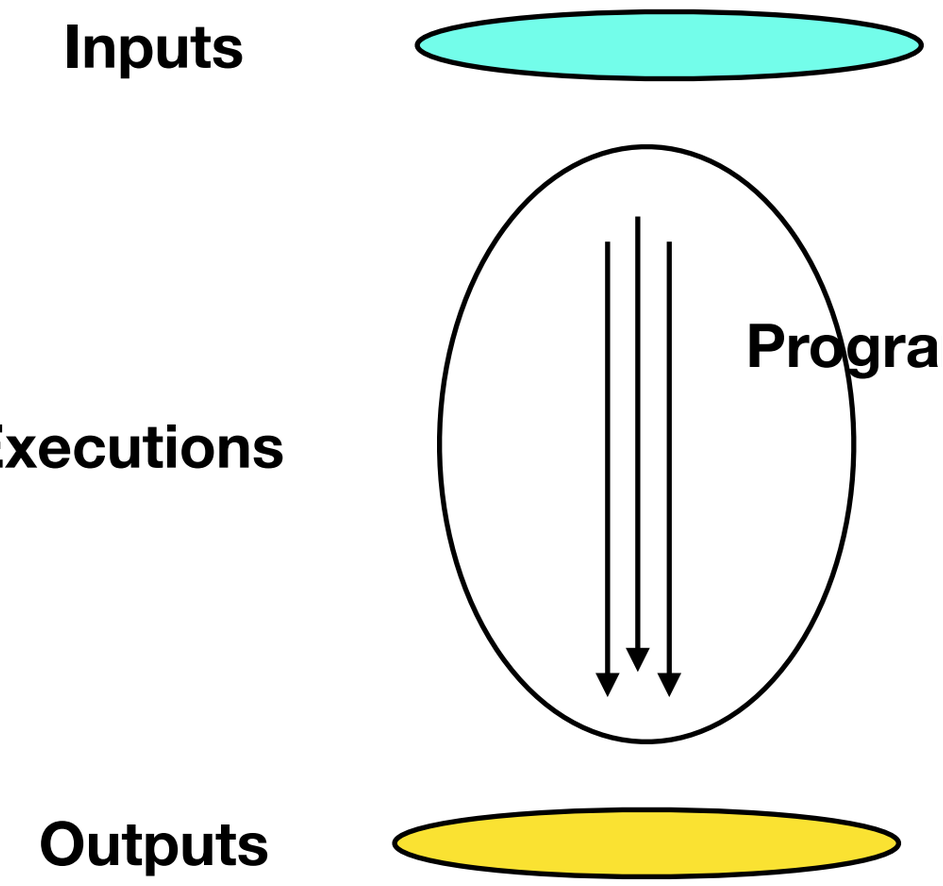
$\mathcal{H}(f^{-1}o)$

$f^{-1}o$

f

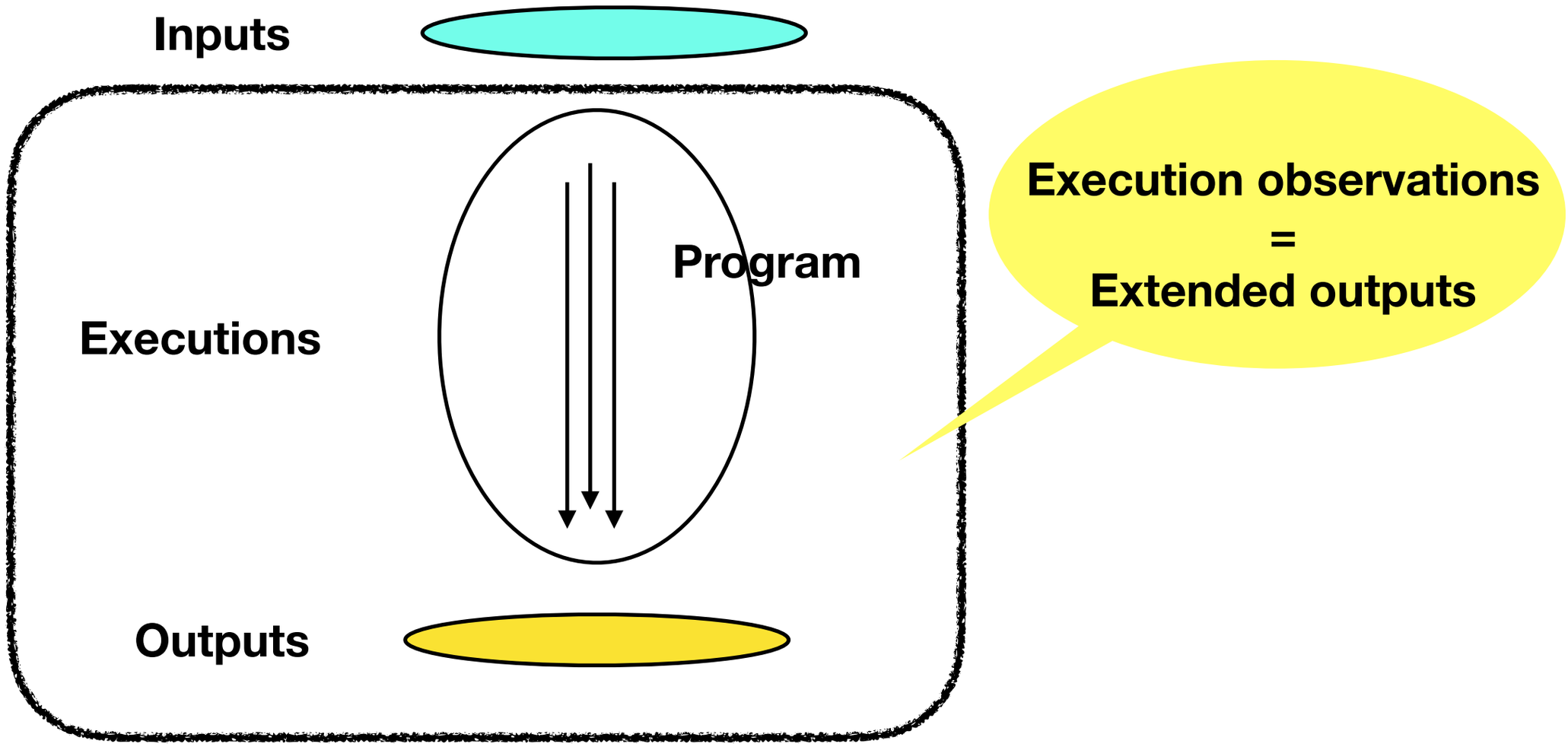
o

$p(o)$



$|O| < |B| < |I|$

Behaviour ~ set of executions

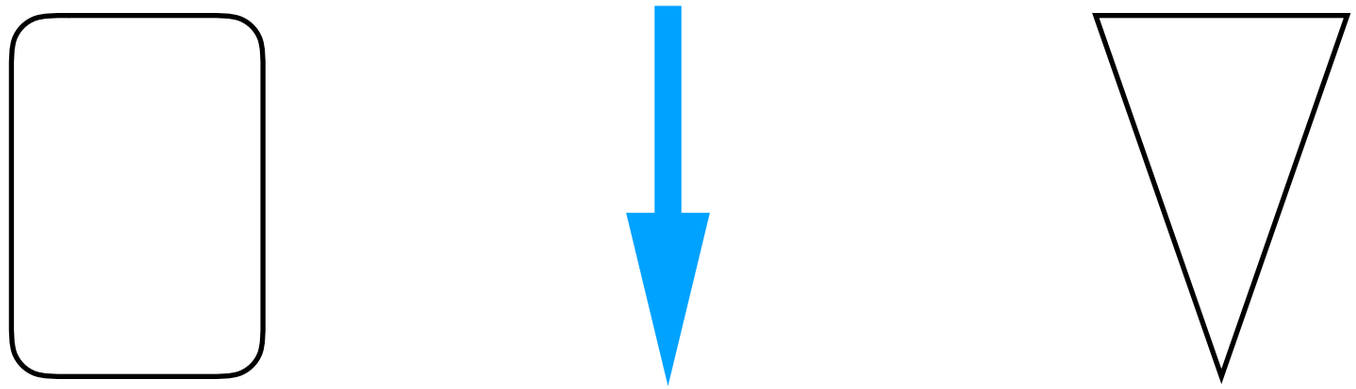

Inputs
Program
Executions
Outputs
Execution observations
=
Extended outputs

40K loc version

Classification Algorithm

ID

ML



not squeezy

direction of execution

squeezy

Failed Error Propagation

Intended

input
t1:x==3
t2:x==-5

Unintended

```
x=x+2;
if(x>0)
   x=x%4;
else x=x;
```

```
x=3*x;
if(x>0)
   x=x%4;
else x=x;
```

Error

output
t1:x==1
t2:x==-3

output
t1:x==1
t2:x==-15

Intended

t

P

A

C

pp

Q

B

o

Unintended

t

P''

A'

$[Q](\llbracket Q \rrbracket \Sigma_{pp'})$

$\Sigma_{pp'}$

C'

pp'

Q

B'

$\llbracket Q \rrbracket \Sigma_{pp'}$

o

Correlation with p(FEP)

Table 4: Spearman's Rank Correlation Coefficient for all programs.

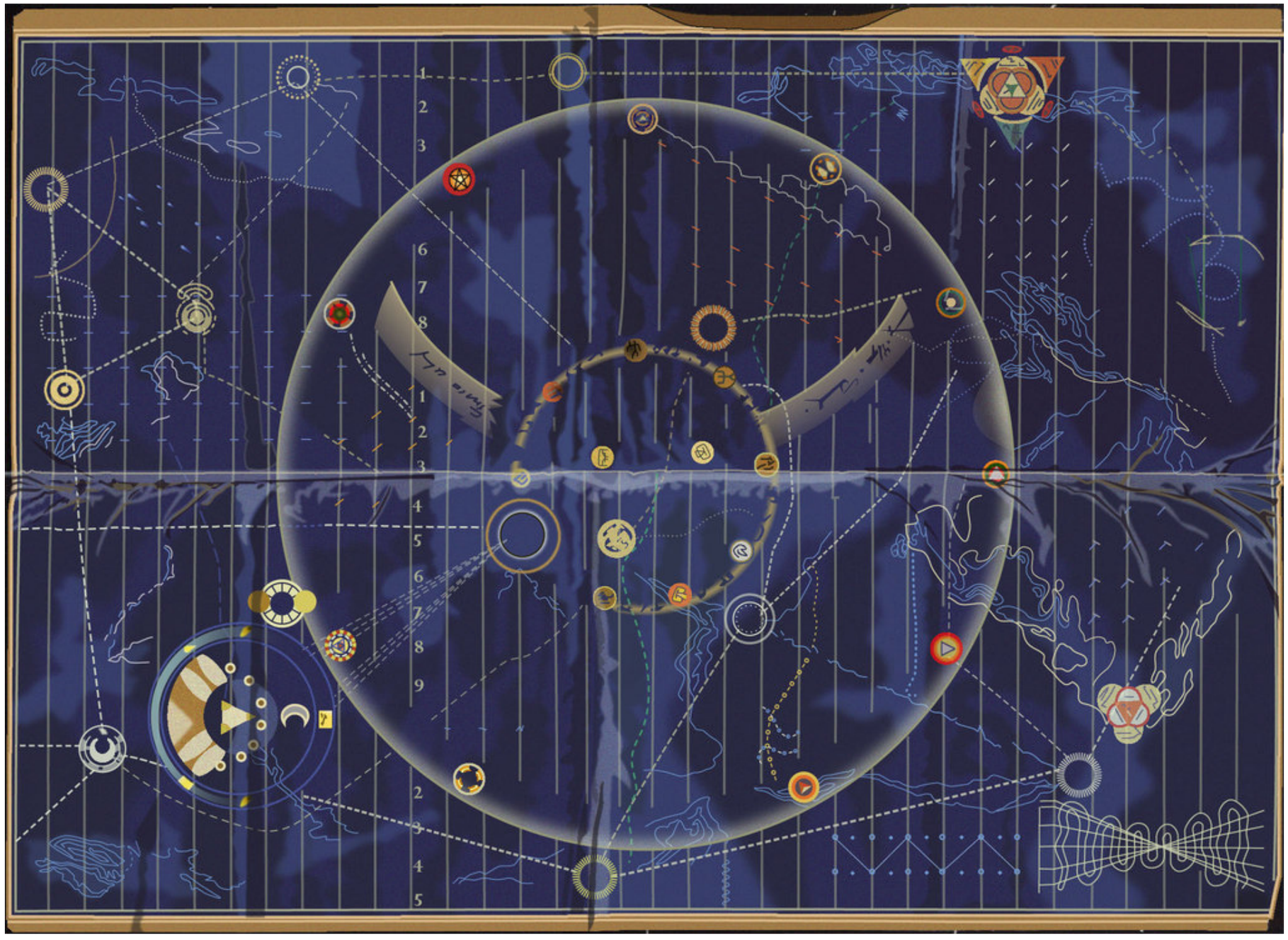
Experiment	Correlation
EXP1	0.715267
EXP2	0.699165
EXP3	0.955647
EXP4	0.948299
EXP5	0.031510

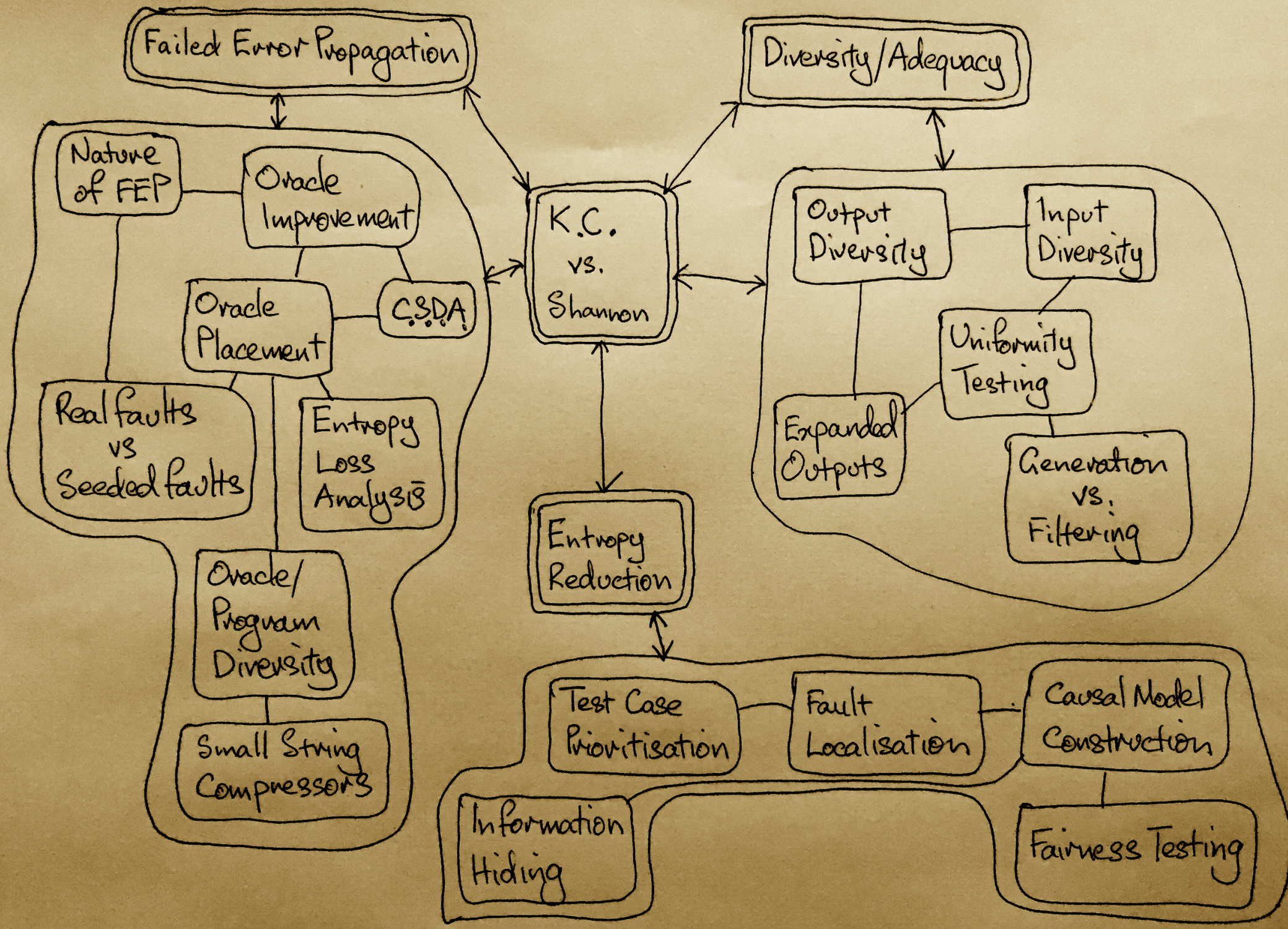
Low squeeziness => low p(FEP)

Table 7: Maximum p(FEP) for all programs

$\mathbf{sq}(Q')$ **Range**	**Max** $\mathbf{sq}(Q)$	**Max p(FEP)**
≤ 0.1	0.090683	0.090683
≤ 0.01	0.001120	0.001120
≤ 0.001	0.000800	0.000200

Back to the Map




Failed Error Propagation
Diversity/Adequacy
Nature of FEP
Oracle Improvement
Oracle Placement
C.S.D.A.
Real faults vs Seeded faults
Entropy Loss Analysis
Oracle/ Program Diversity
Small String Compressors
K.C. vs. Shannon
Output Diversity
Input Diversity
Uniformity Testing
Expanded Outputs
Generation vs. Filtering
Entropy Reduction
Test Case Prioritisation
Fault Localisation
Causal Model Construction
Information Hiding
Fairness Testing