


FACULTEIT INGENIEURSWETENSCHAPPEN

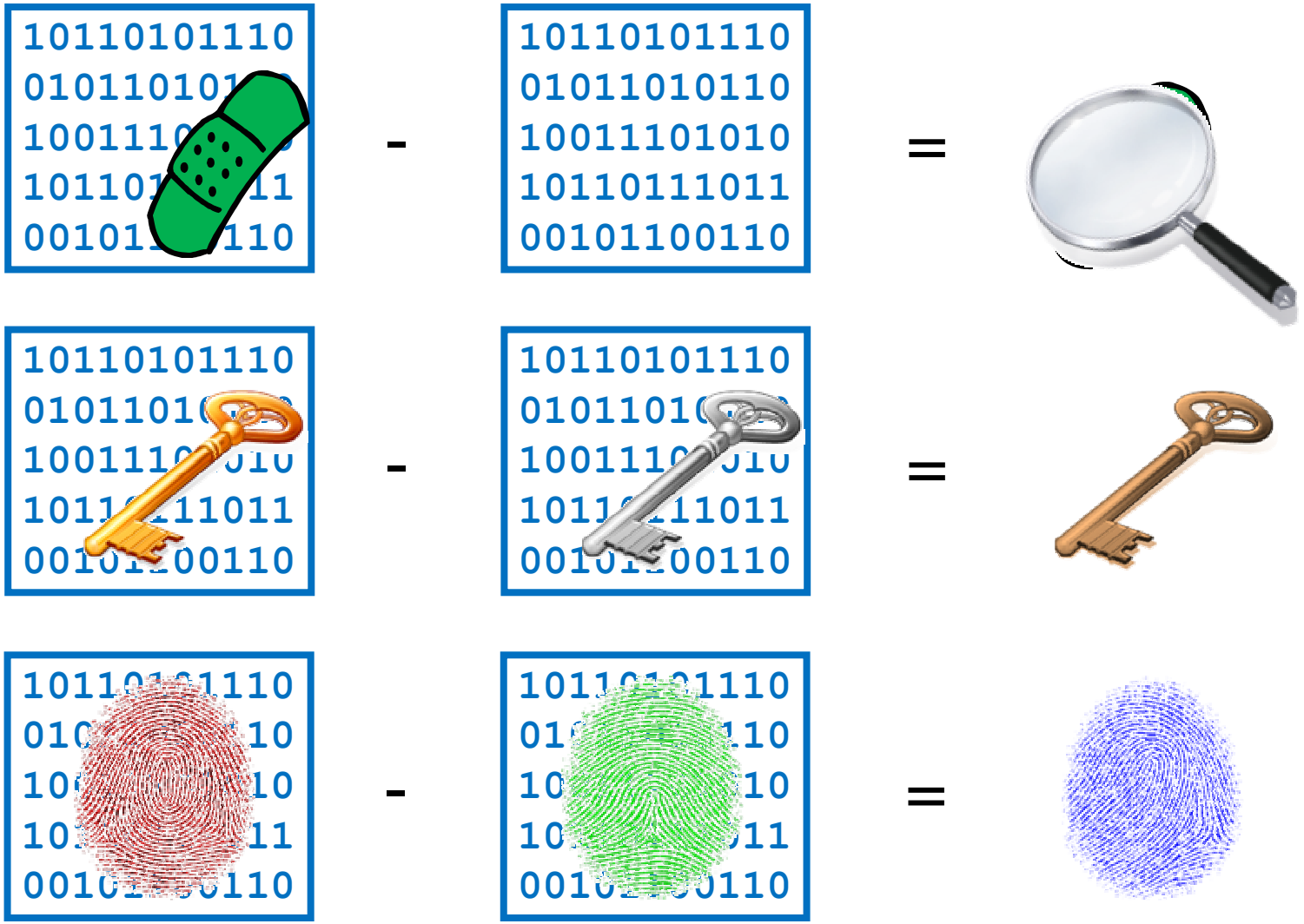
Diversity for Software Protection

Bertrand Anckaert

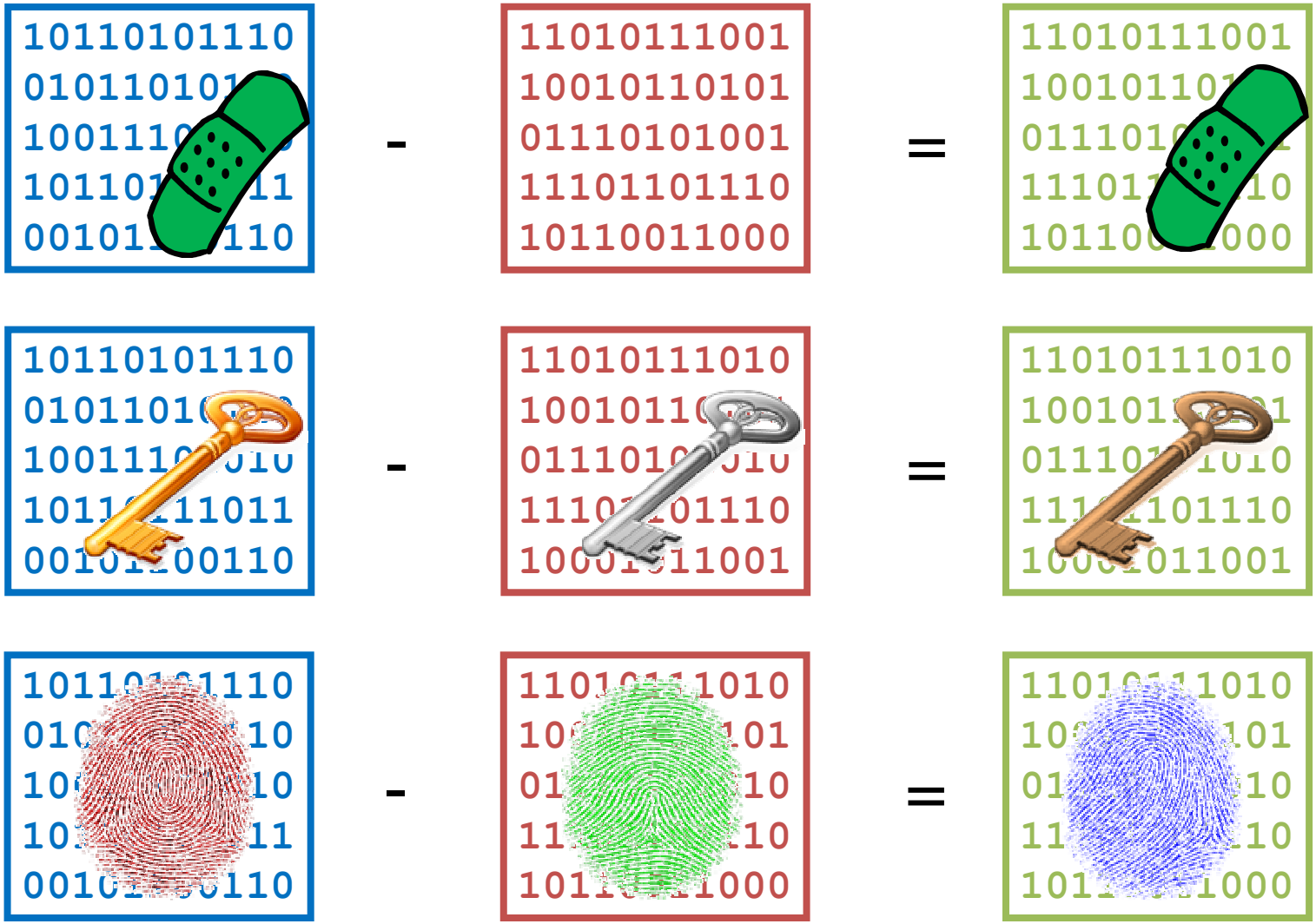
Koen De Bosschere

Plenary workshop on Remote EnTrusting by RUn-time Software auThentication, Sept.

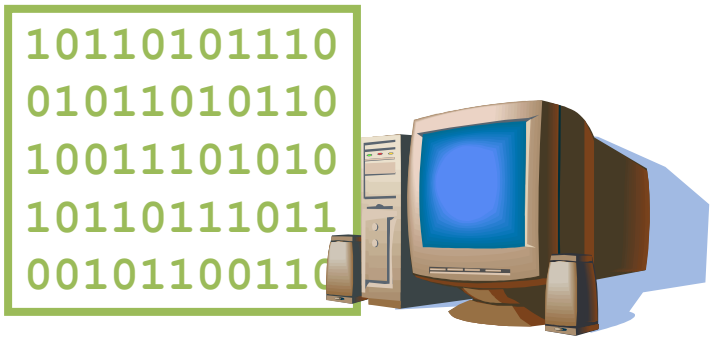
The δ between versions may leak critical information



Artificial diversity can hide the δ in an artificially large Δ



Diversity helps to mitigate the break once, break every time problem



Outline

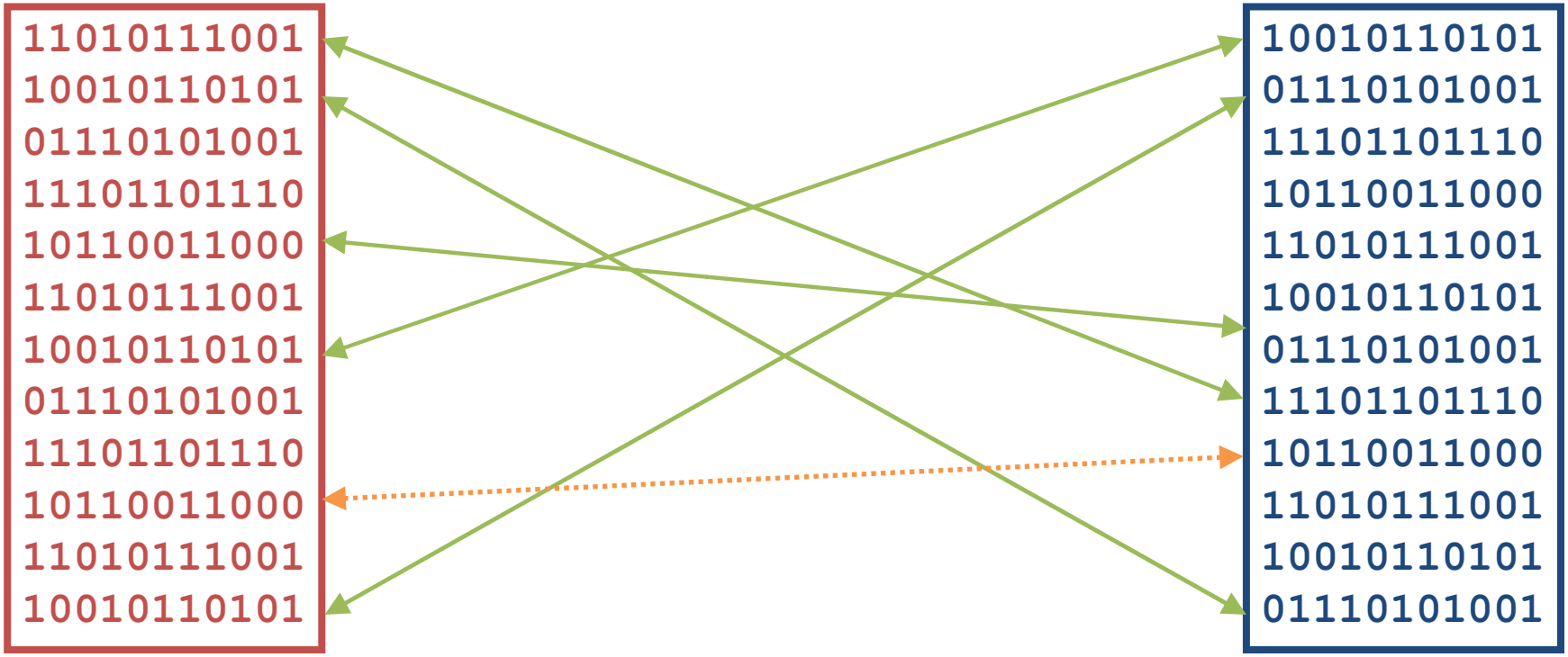
Applications

- Hide δ within Δ
- Mitigate break once break every time problem

Matching system

Diversity system

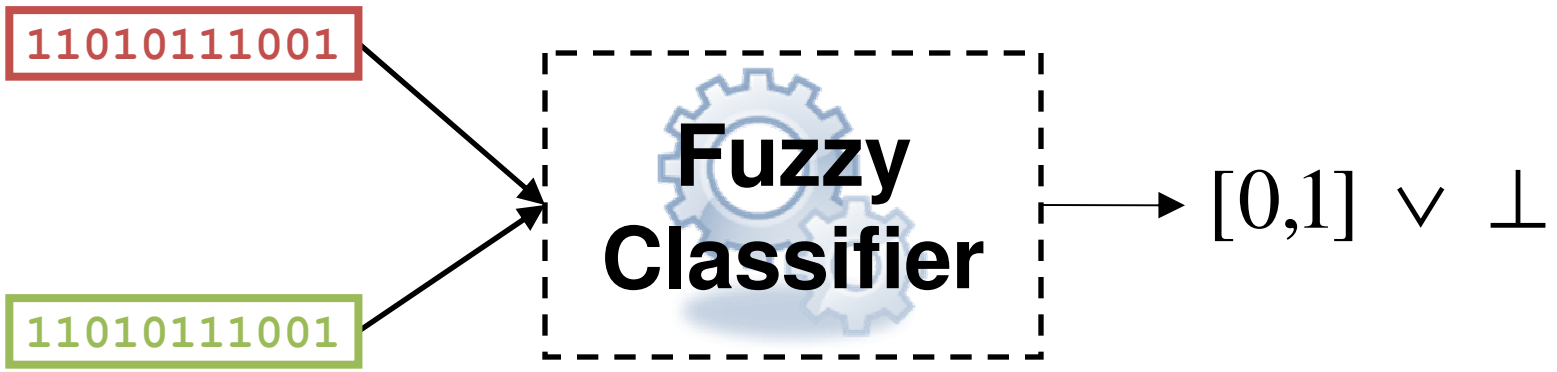
We try to identify related code fragments between versions



Two types of error:

False positives and false negatives

Fuzzy Classifiers



Fuzzy classifiers operate on different types of information

Execution count

Instruction syntax

Data

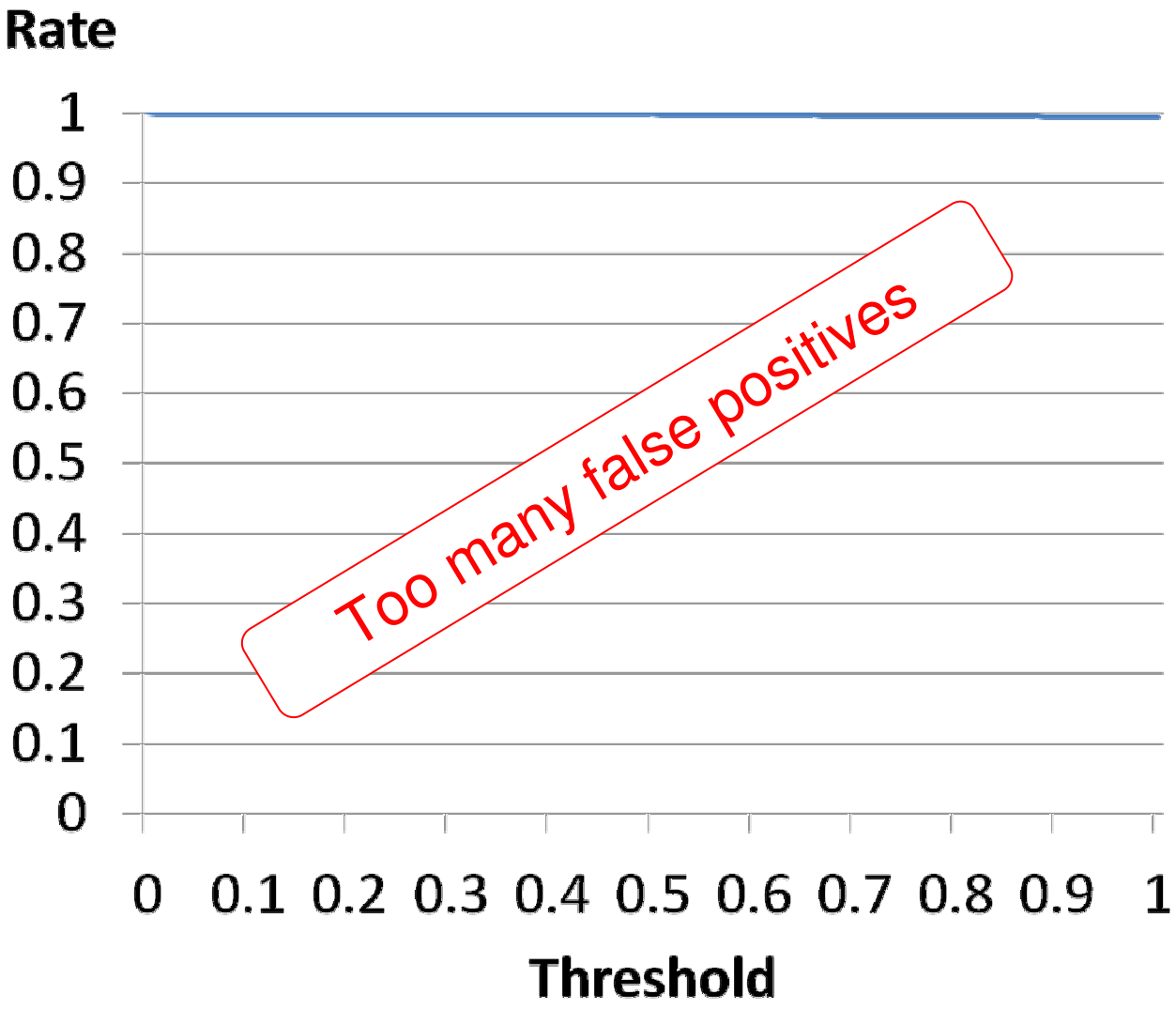
System calls

Data flow

Control flow

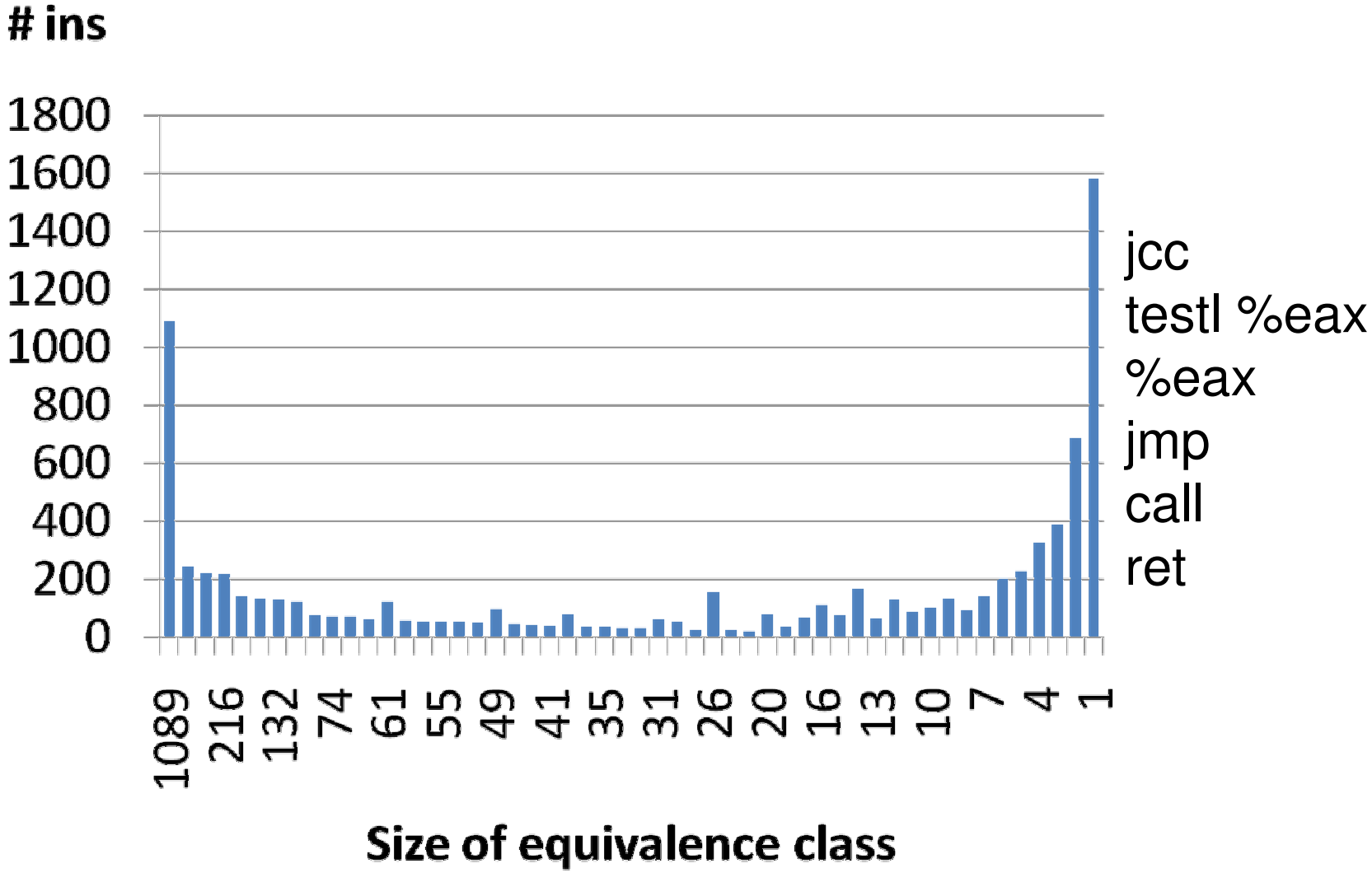
First time executed

Classifier: Instruction Syntax

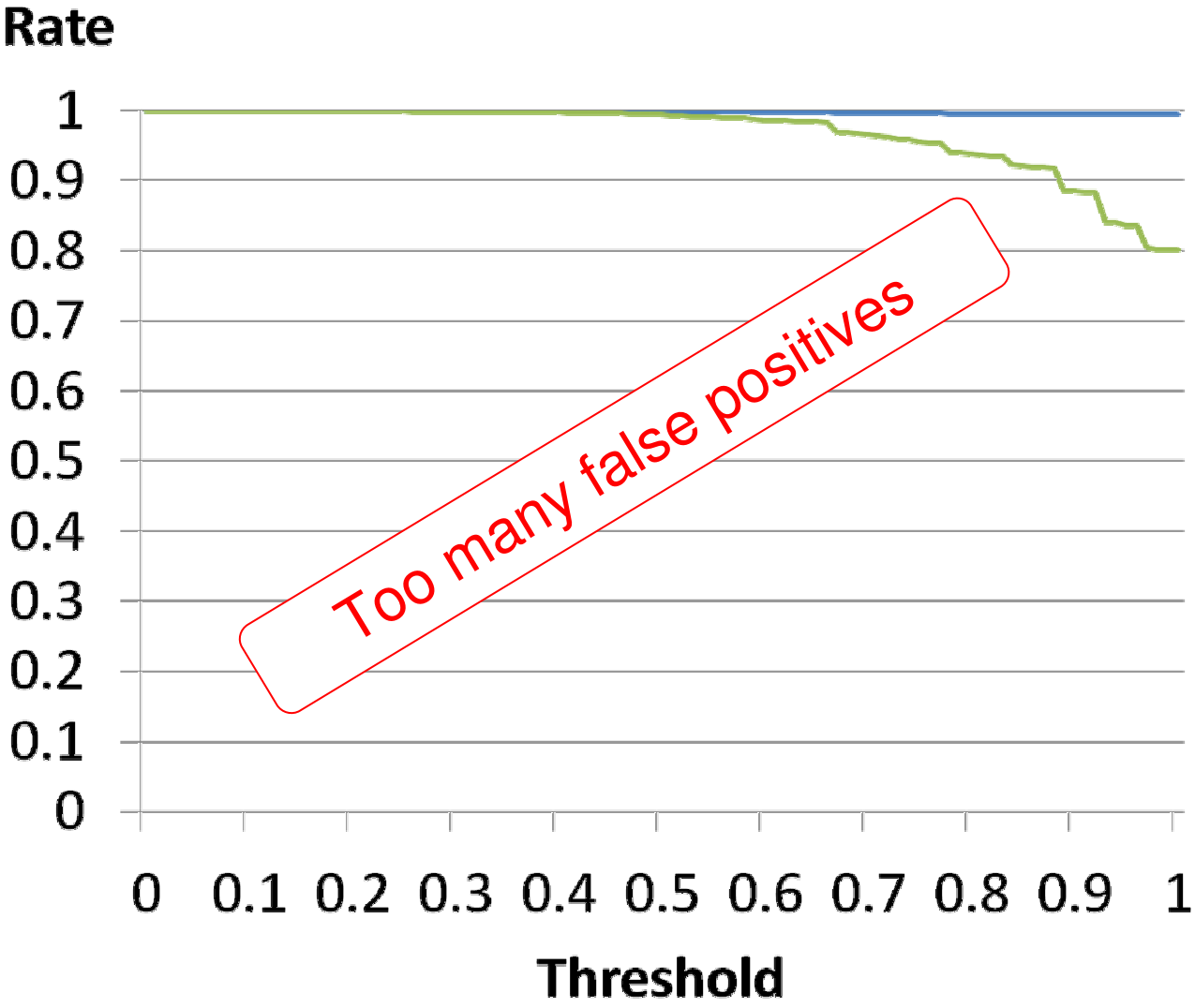


$$\pi = \frac{|\mu_e \setminus \mu_r|}{|\mu_e|}$$

Problem: large equivalence classes

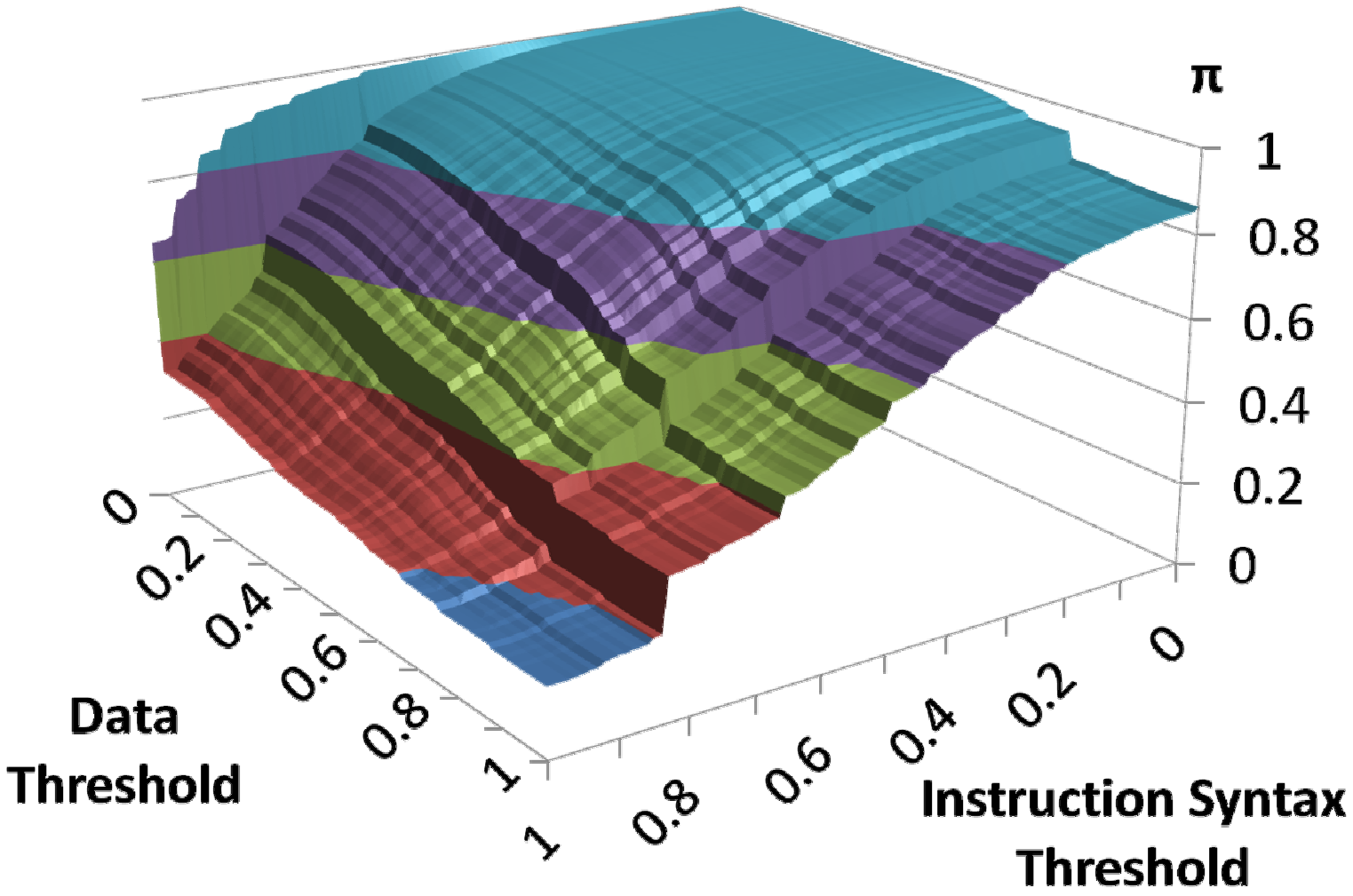


Solution 1: Look at Larger Context



$$\pi = \frac{|\mu_e \setminus \mu_r|}{|\mu_e|}$$

Solution 2: Combination



Fuzzy classifiers operate on different types of information

Execution count

Instruction syntax

Data

System calls

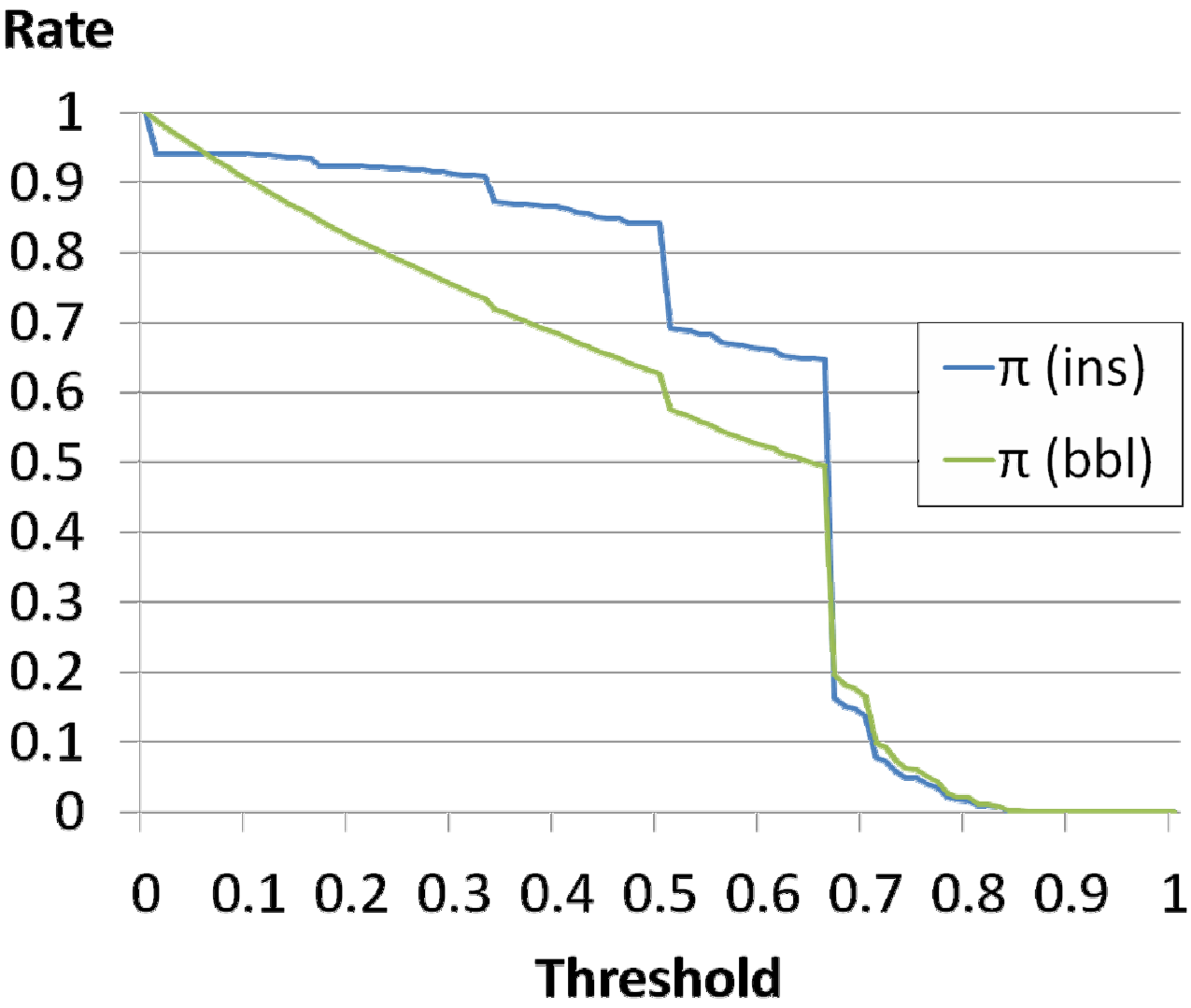
Data flow

Control flow

First time executed

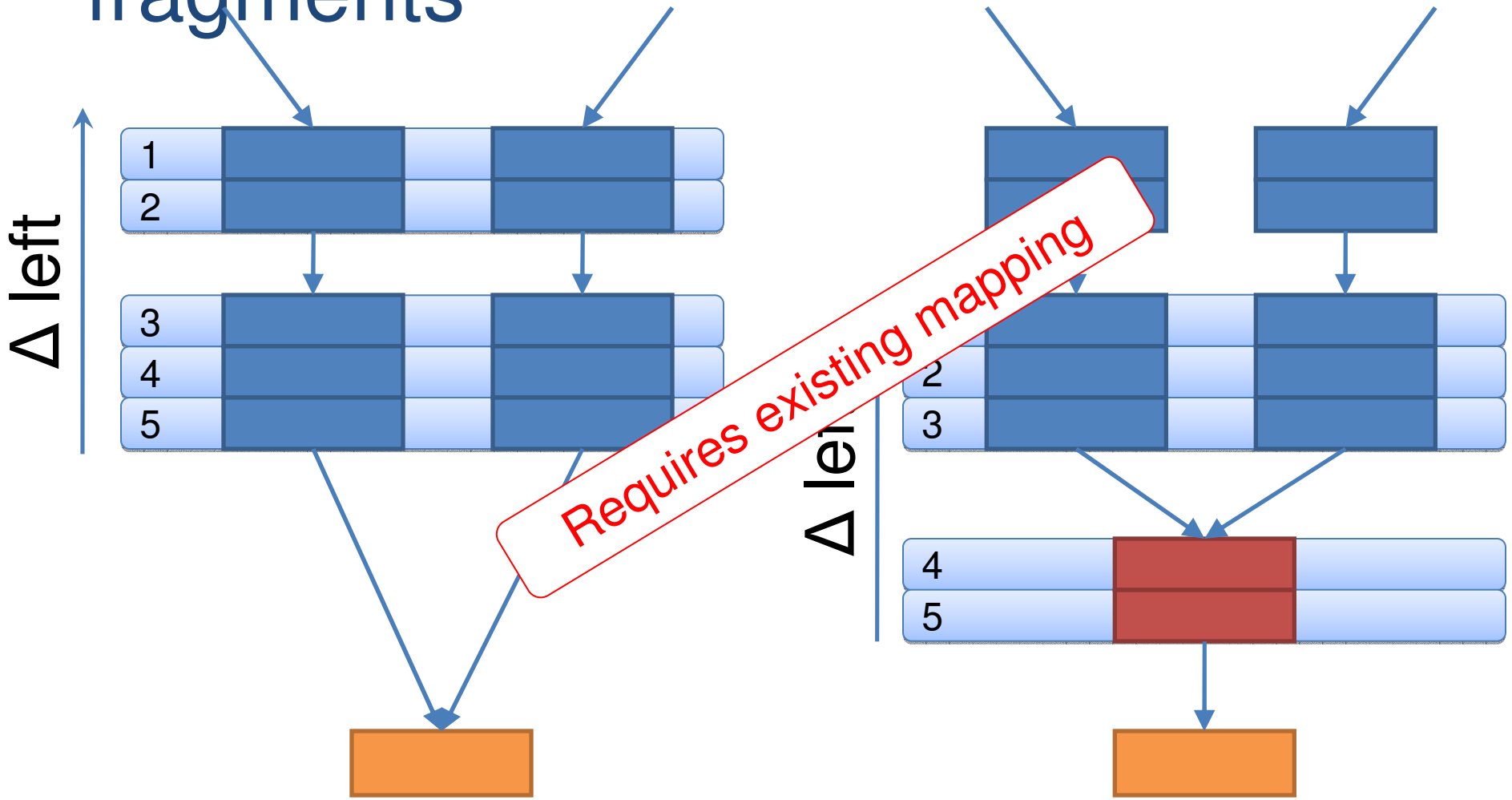
DIOTA

Classifier: Control Flow



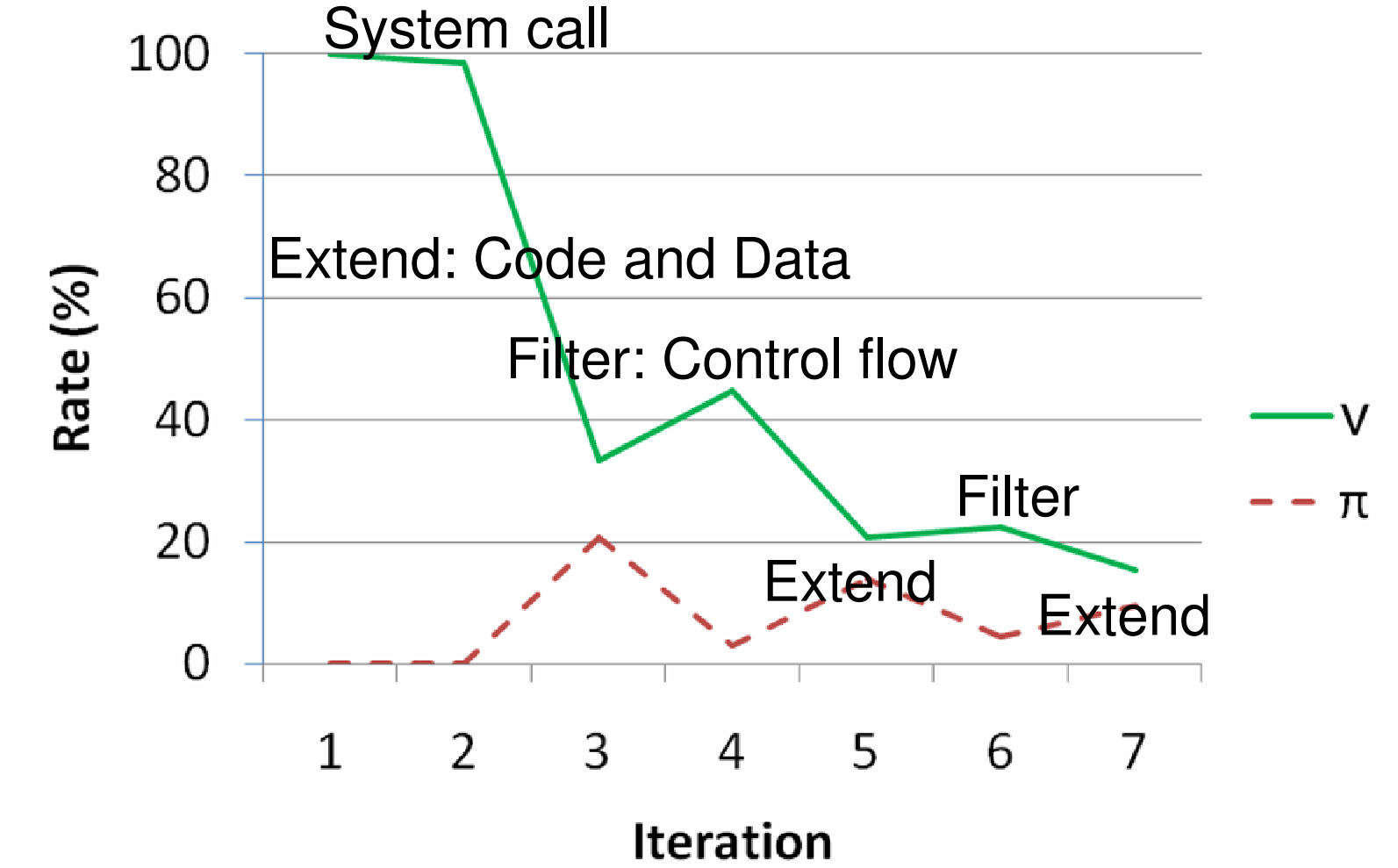
$$\pi = \frac{|\mu_e \setminus \mu_r|}{|\mu_e|}$$

Control flow classifier measures proximity to other matched fragments



Parameters: distance (Δ=5) and direction (=UP)

Solution 3: Iteration – Extend & Filter



We can build a reliable matching system from fuzzy classifiers

Solution 1: Look at larger context

Solution 2: Combination

Solution 3: Iteration

Solution 4: Limit # matches per code fragment

Outline

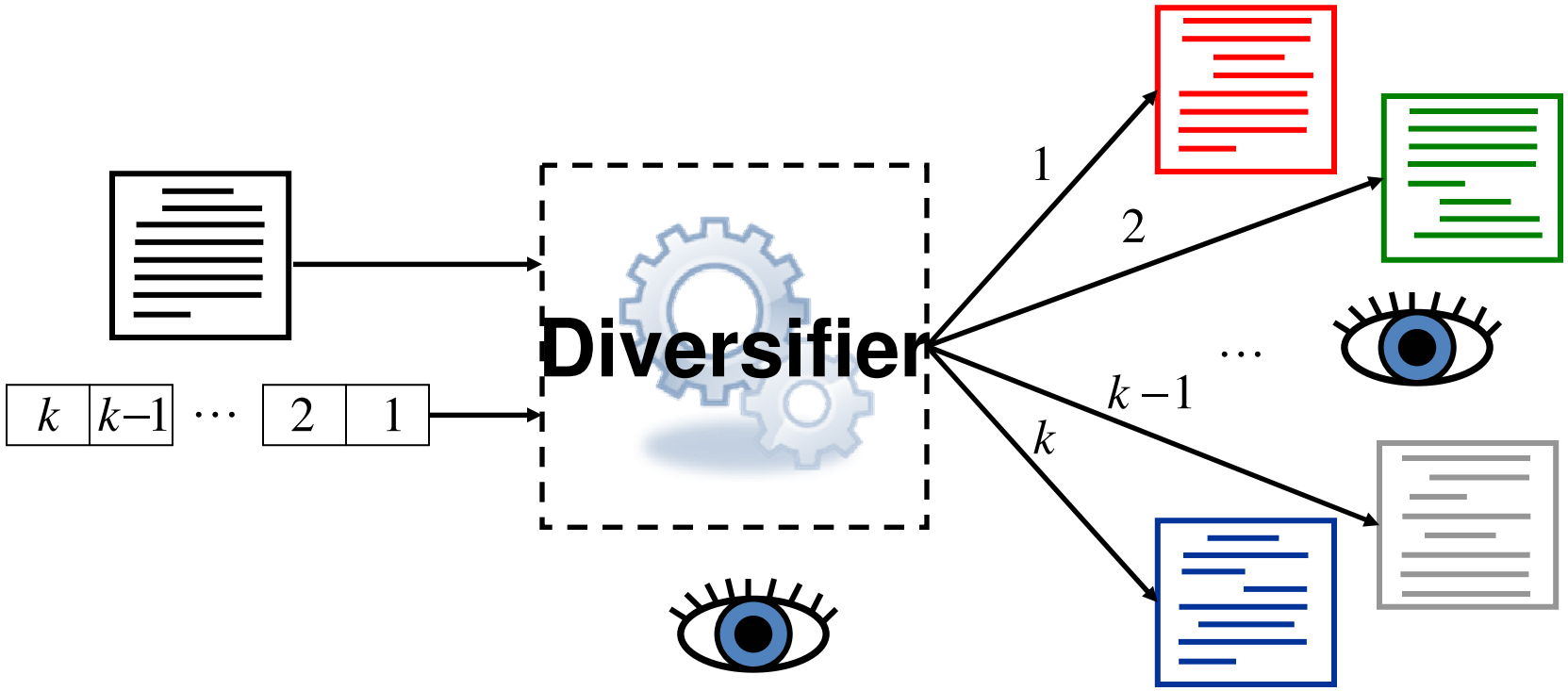
Applications

- Hide δ within Δ
- Mitigate break once break every time problem

Matching system

Diversity system

Diversity system



Diversifying Transformations

Folding

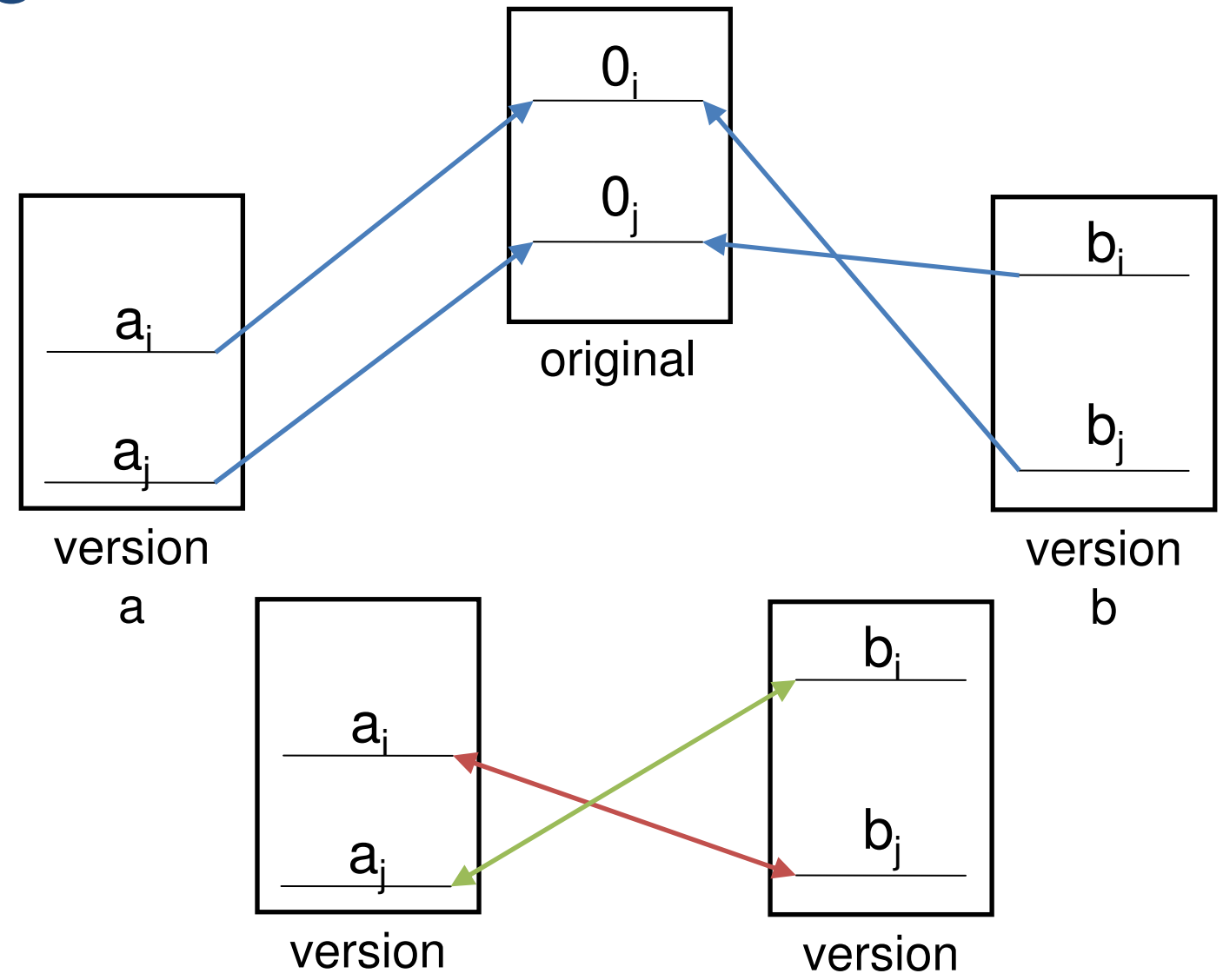
Self-modifying code

Control flow obfuscation

Unfolding

Code generation

Transformations keep track of the original addresses



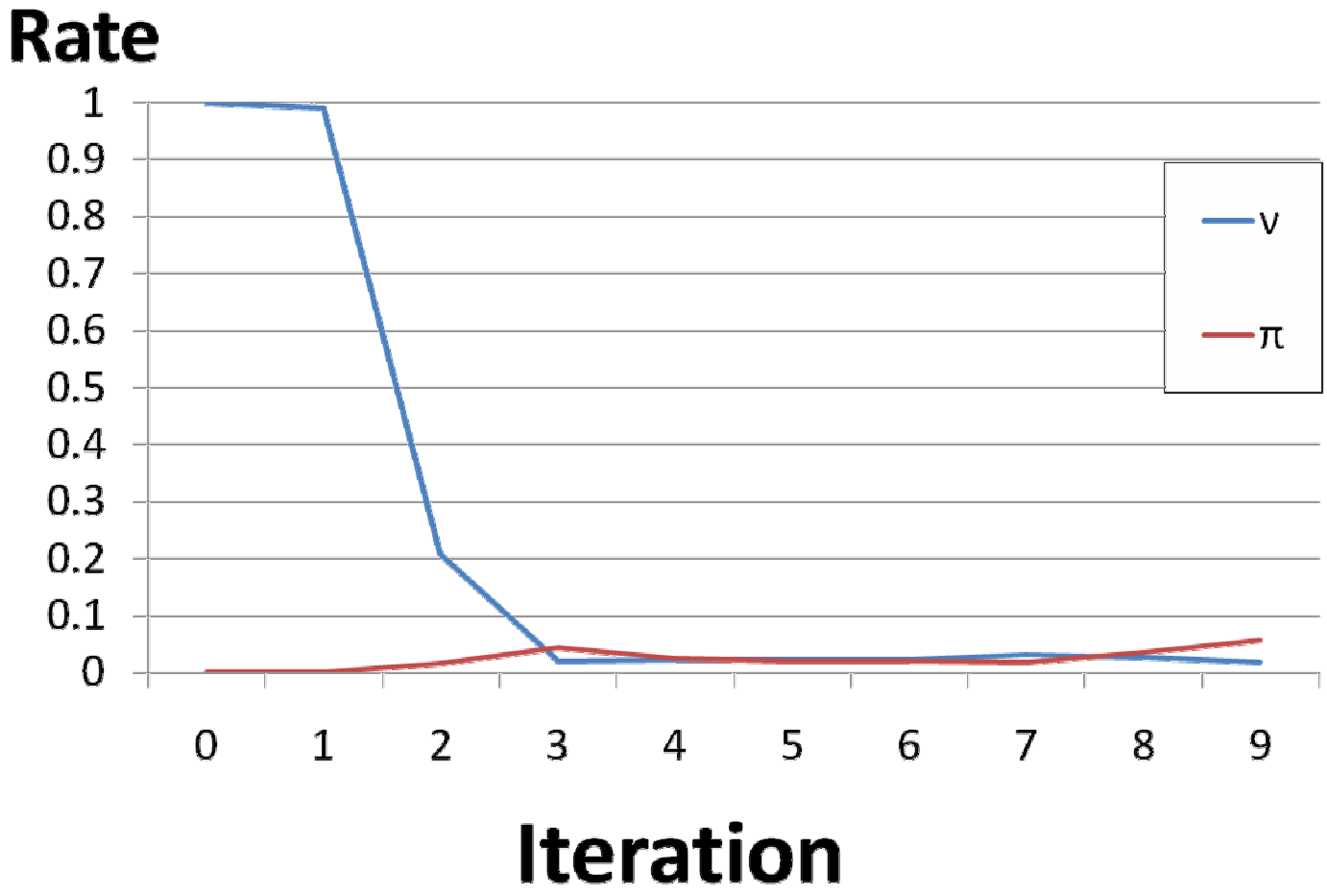
Code generation fools text-based classifiers

Change order through

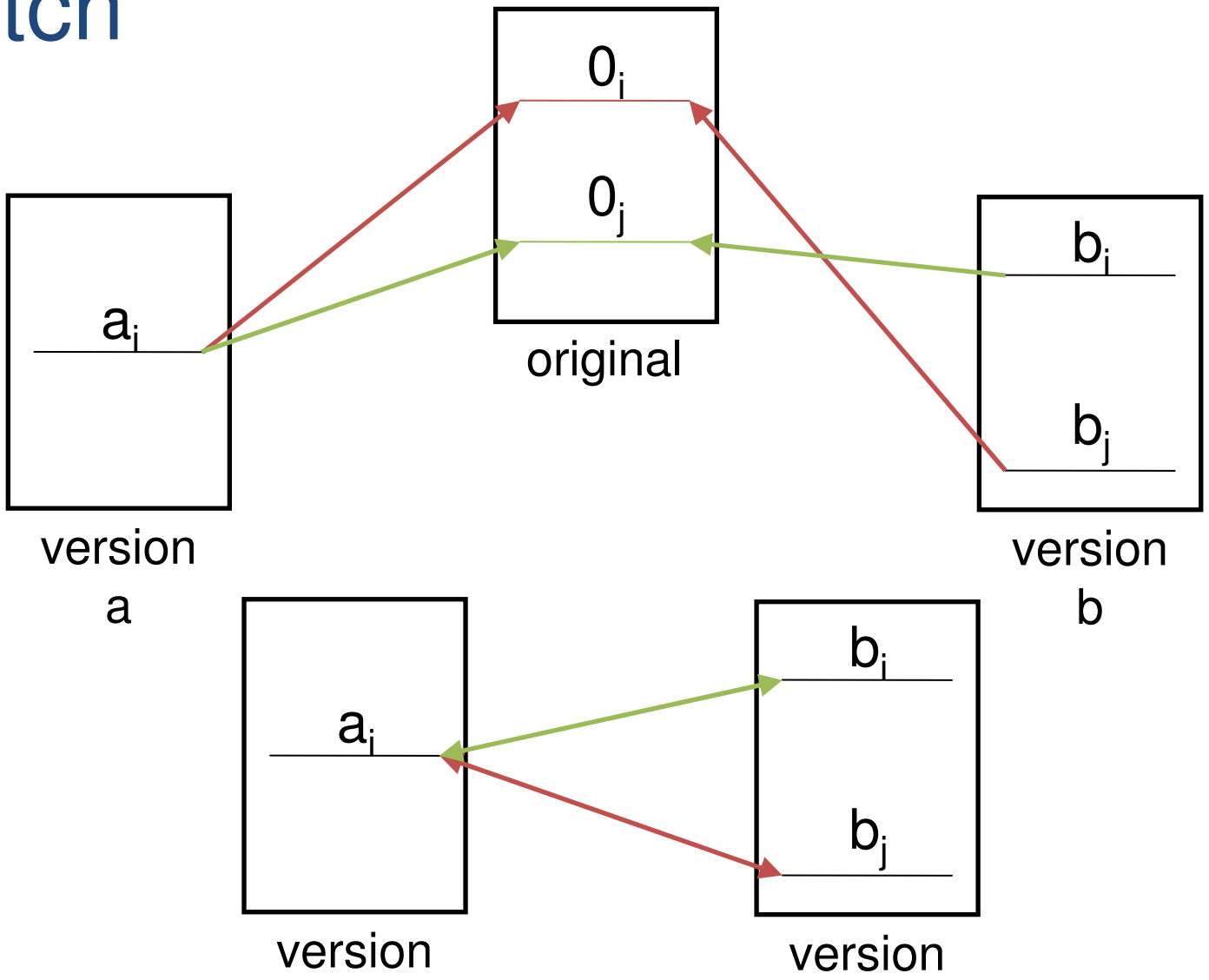
- Scheduling
- Code Layout

Change instruction syntax

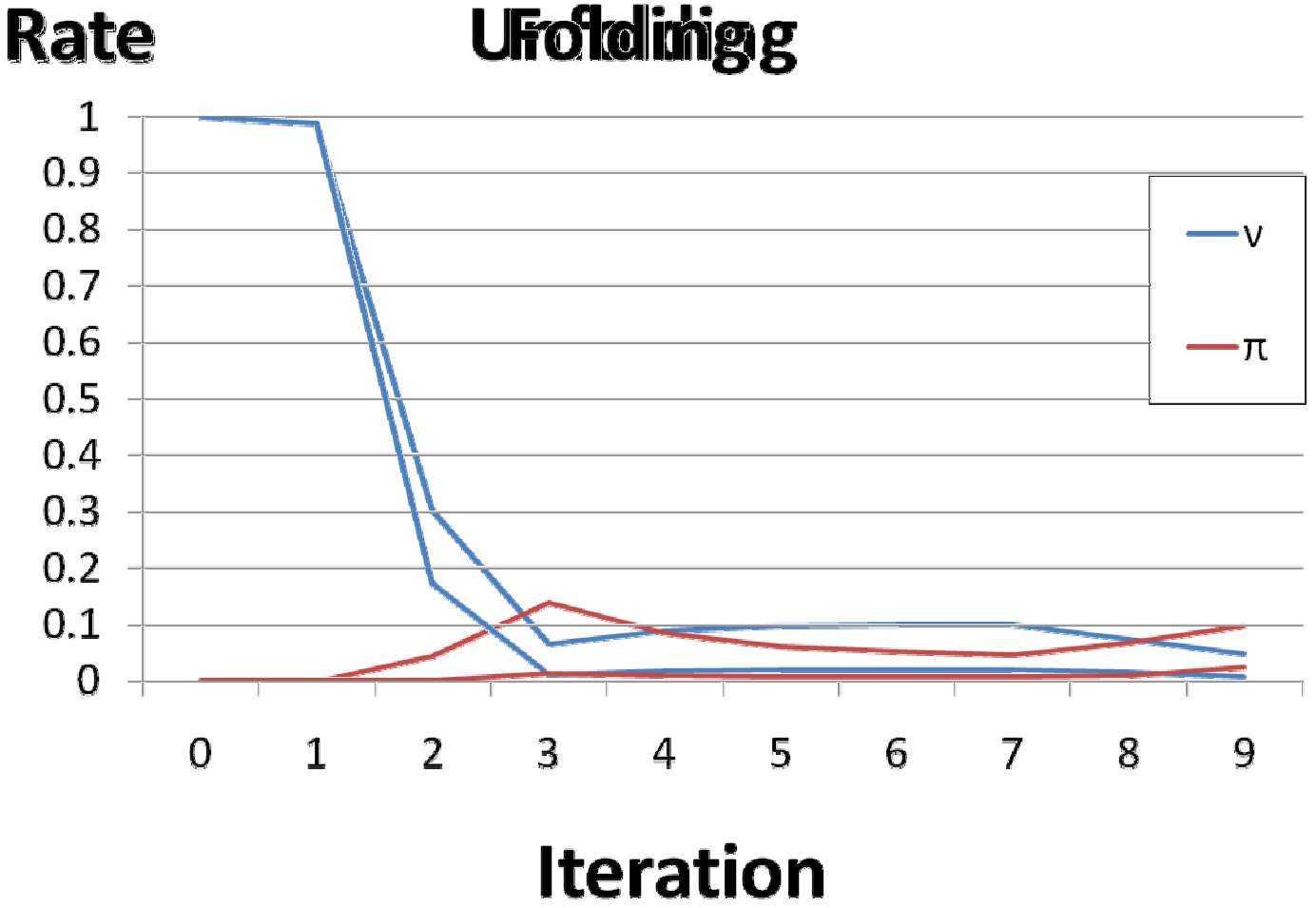
Code generation has marginal impact



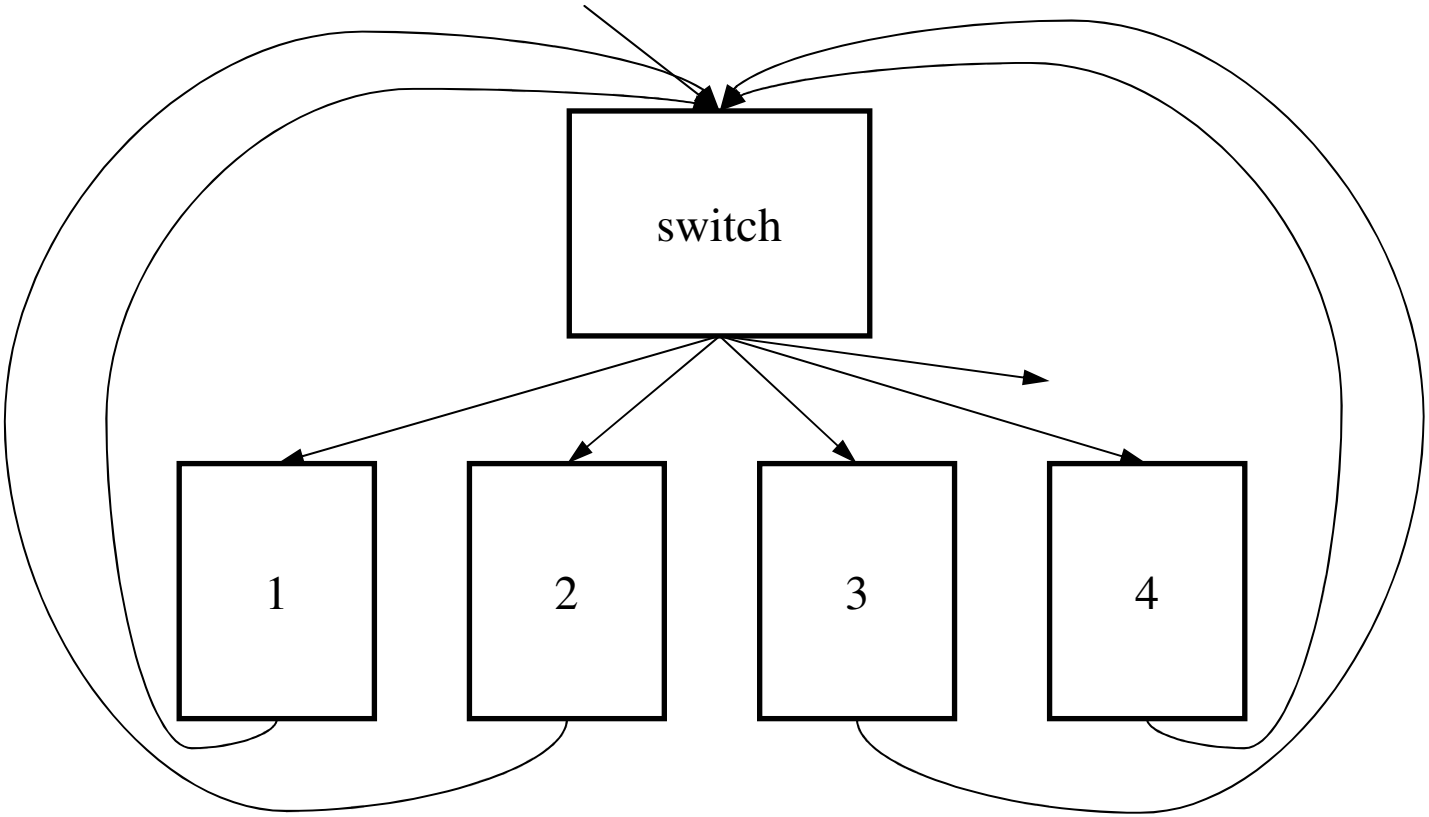
Folding and unfolding undermine the assumption of at most one match



Folding and Unfolding have little impact



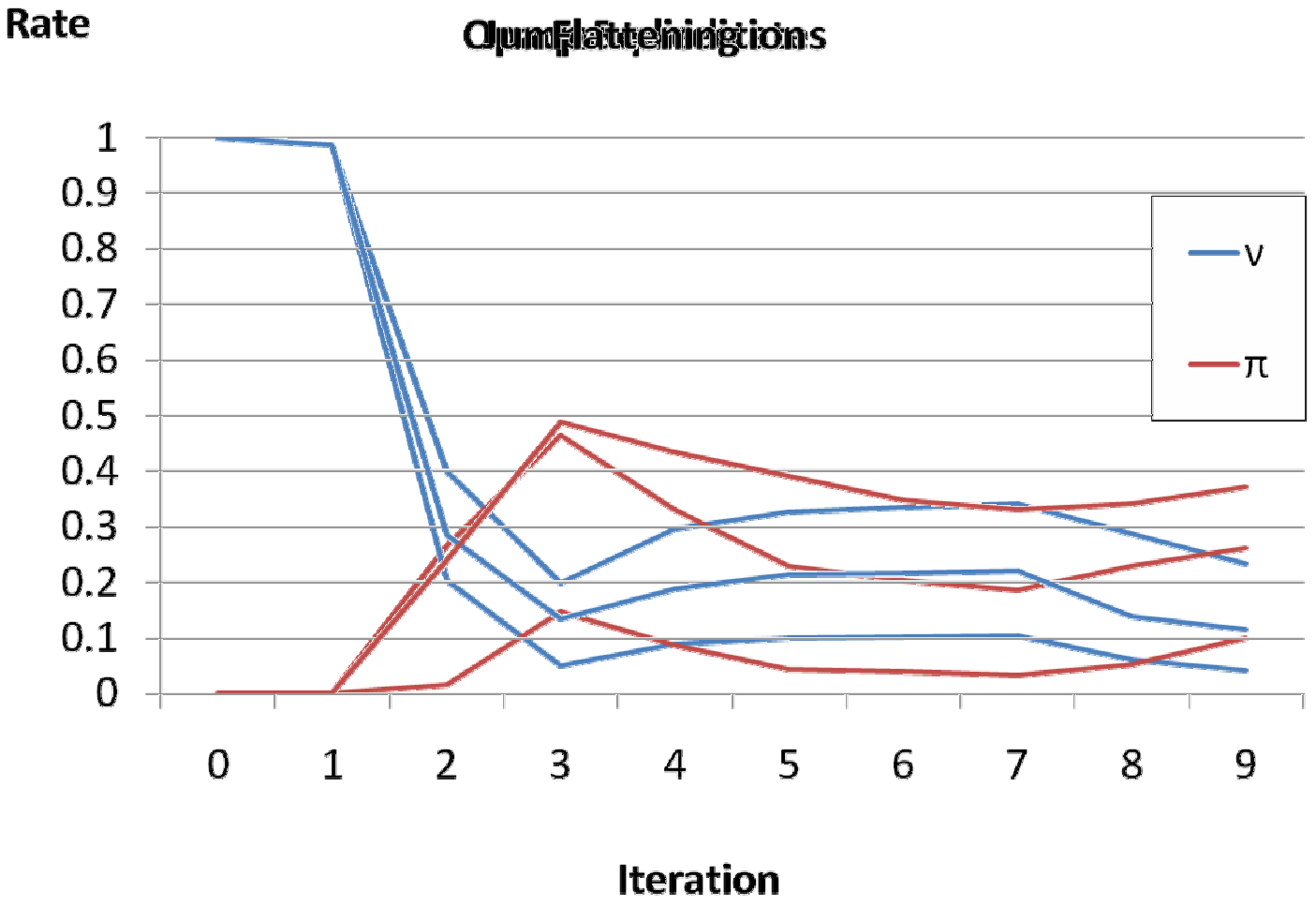
Control flow obfuscation affects the control flow classifier



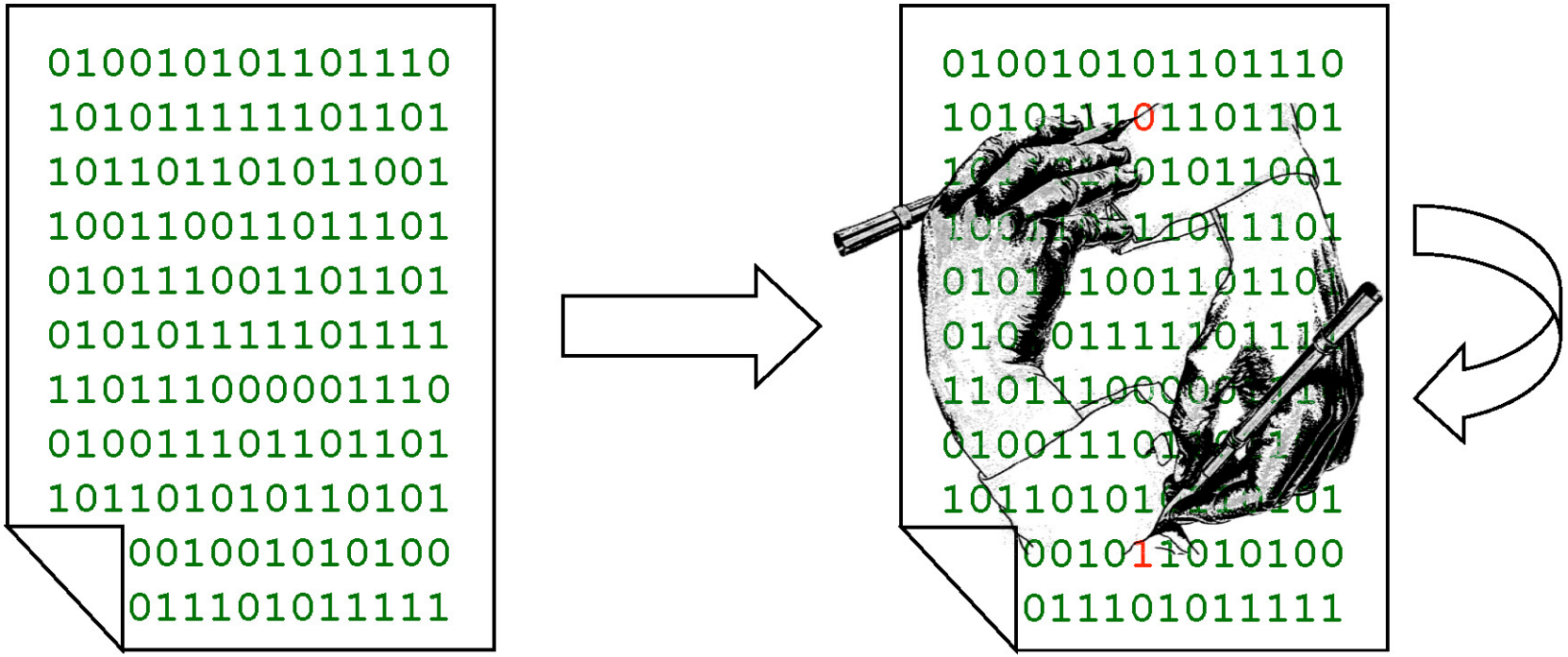
Less accurate

Slower

Control Flow Obfuscation has moderate impact



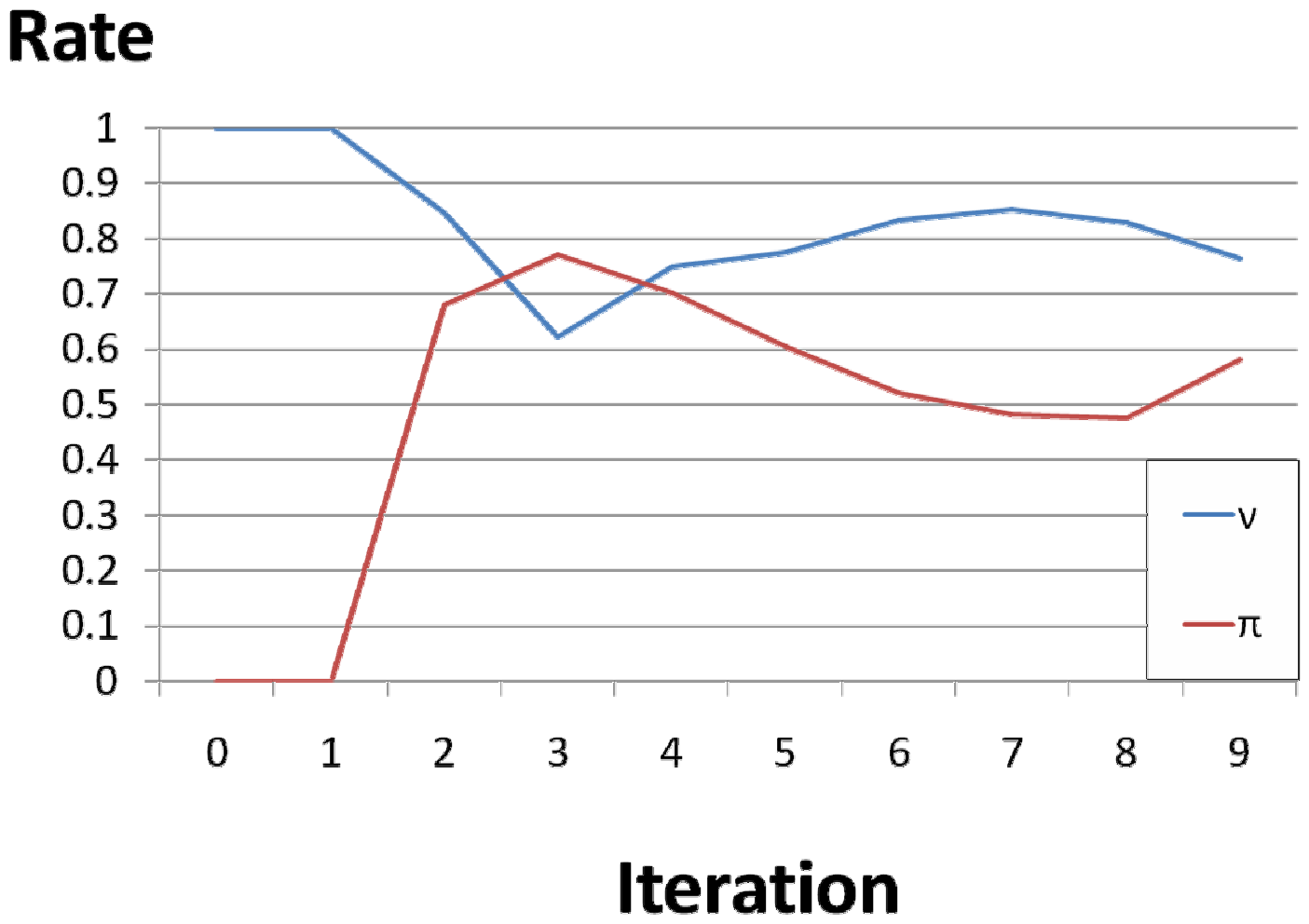
Self-Modifying code hampers the information collection



Undermines assumption of constant code

Slows down information collection

Combined impact is significant




FACULTEIT INGENIEURSWETENSCHAPPEN

Diversity for Software Protection

Bertrand Anckaert

Koen De Bosschere

Plenary workshop on Remote EnTrusting by RUn-time Software auThentication, Sept.

Diversity delays automation