Adaptive Behavioural Model Learning for Software Product Lines

Shaghayegh Tavassoli, University of Tehran

Diego Damasceno, Radboud University Nijmegen

Ramtin Khosravi, University of Tehran

Mohammad Mousavi, King’s College London

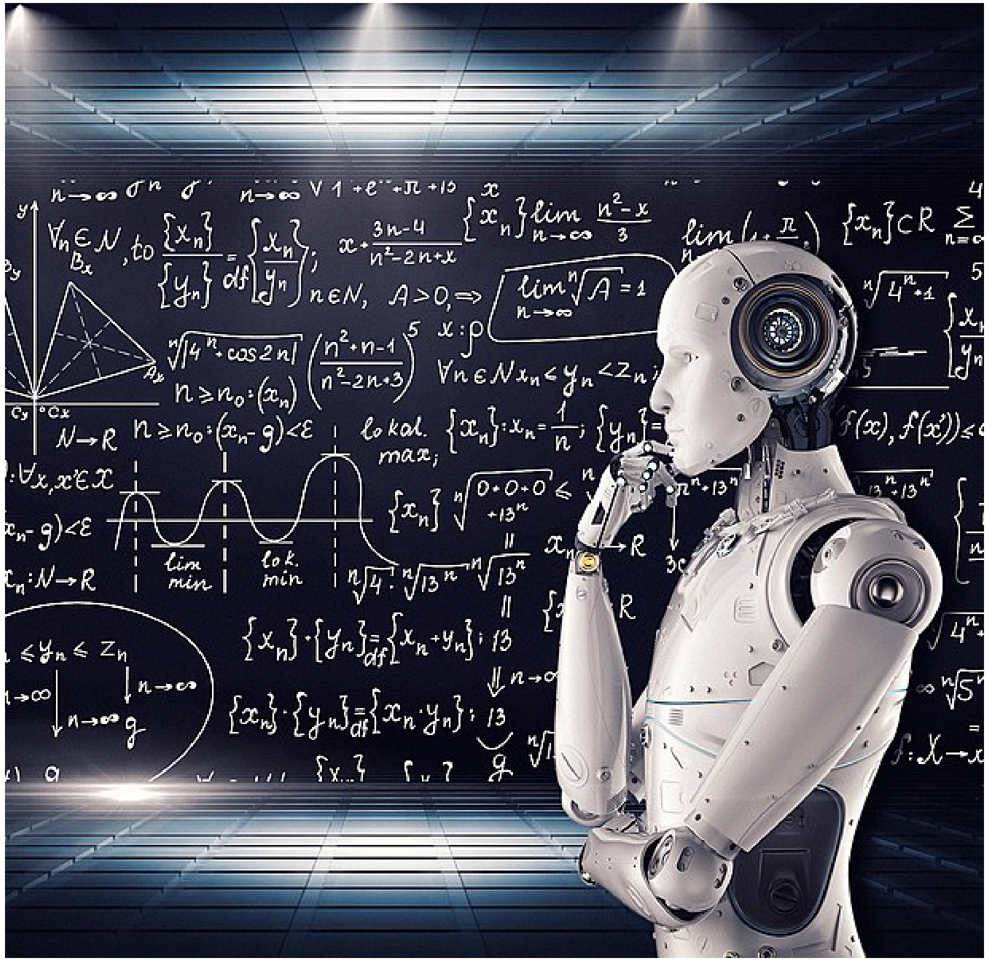
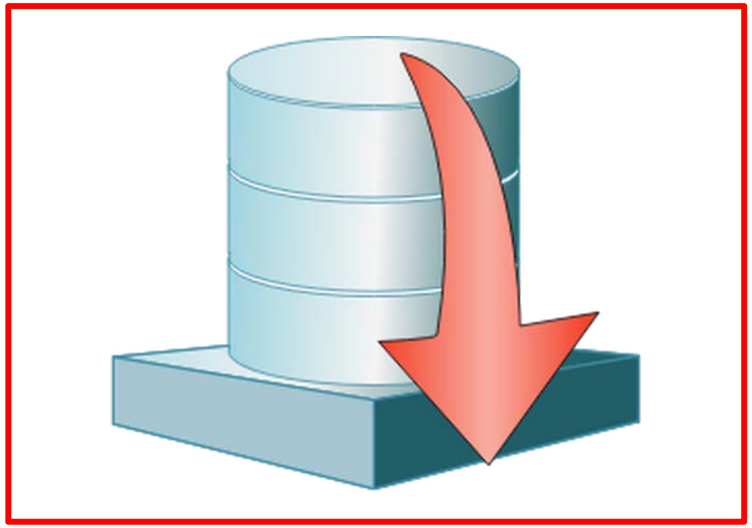


Photo: Wikimedia Commons

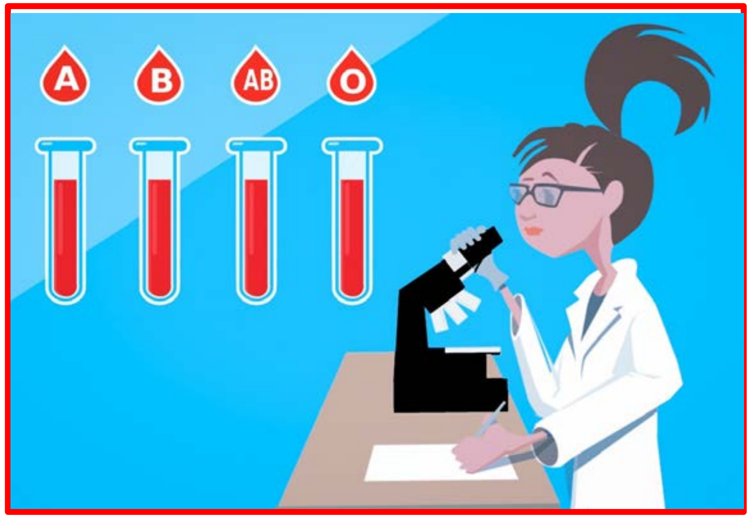
THE MESSAGE

Reusing queries makes active learning of behavioural models of a software product line more efficient.

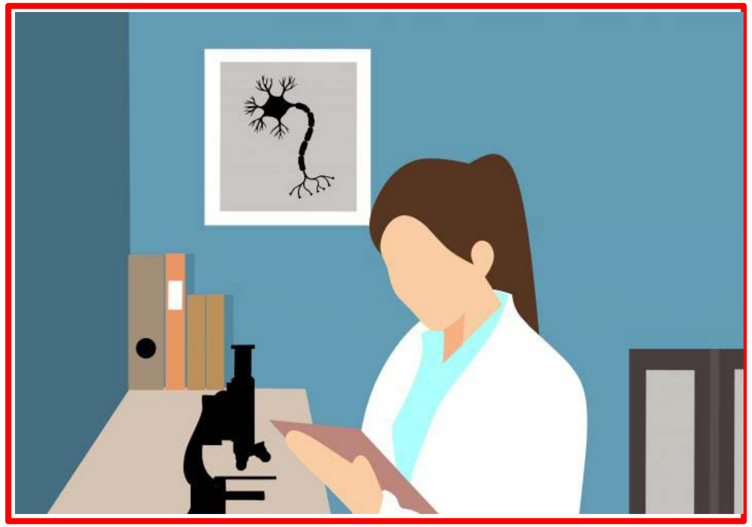
Active Learning



Query Repository



Sampling in Learning



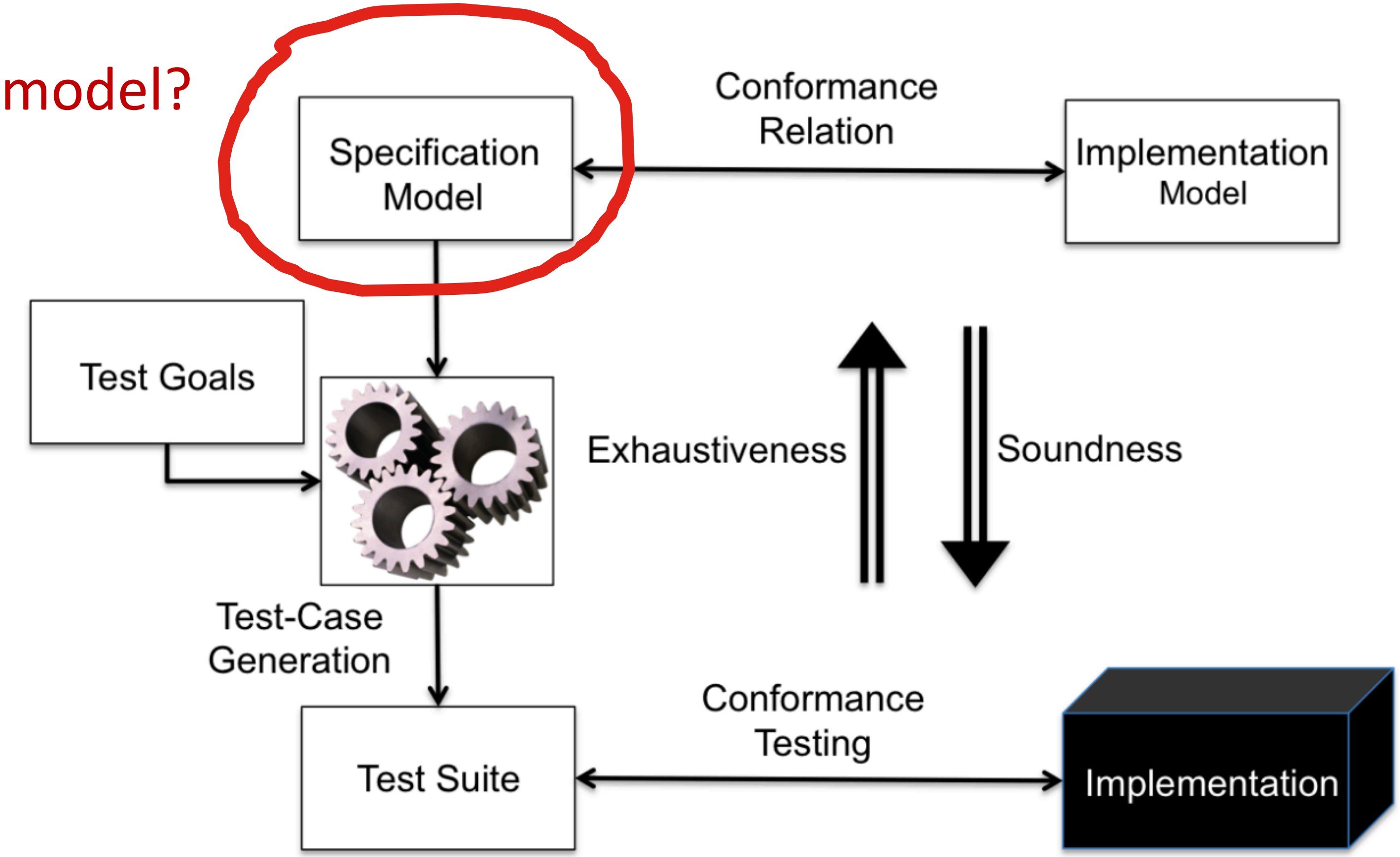
Empirical Evaluation

THE MESSAGE

Reusing queries makes active learning of behavioural models of a software product line more efficient.

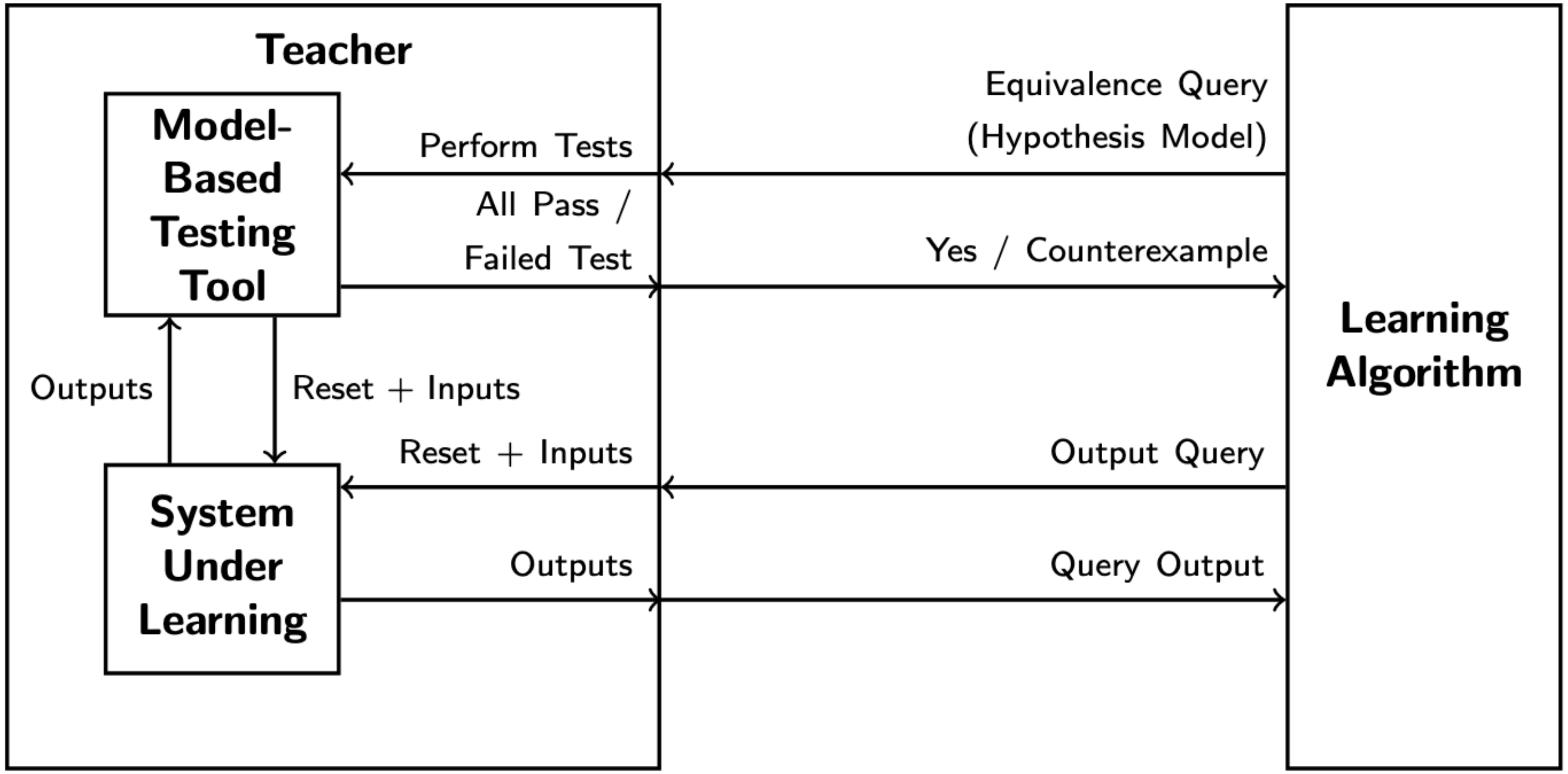
Active Learning: Why?

Model? What model?



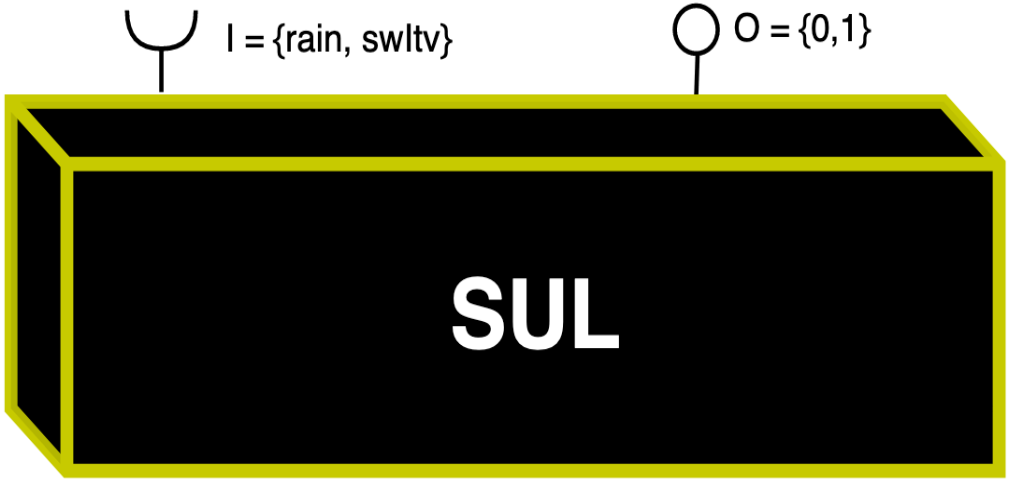
[Aichernig, Mostowski, Mousavi, Tappler and Taromirad. Model Learning and Model-Baed Testing]

Active Learning: What?

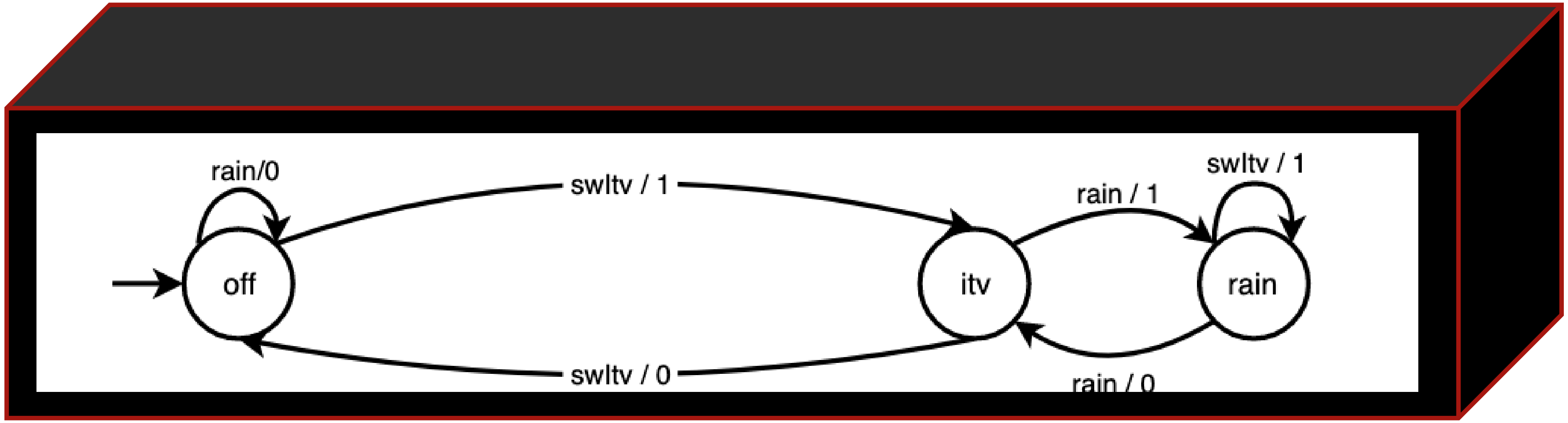


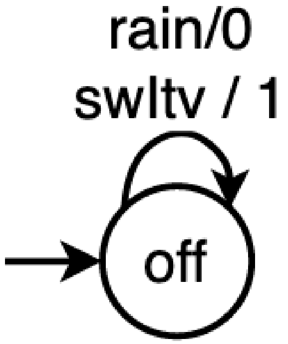
[Dana Angluin. Learning regular sets from queries and counterexamples.]

Active Learning: How?



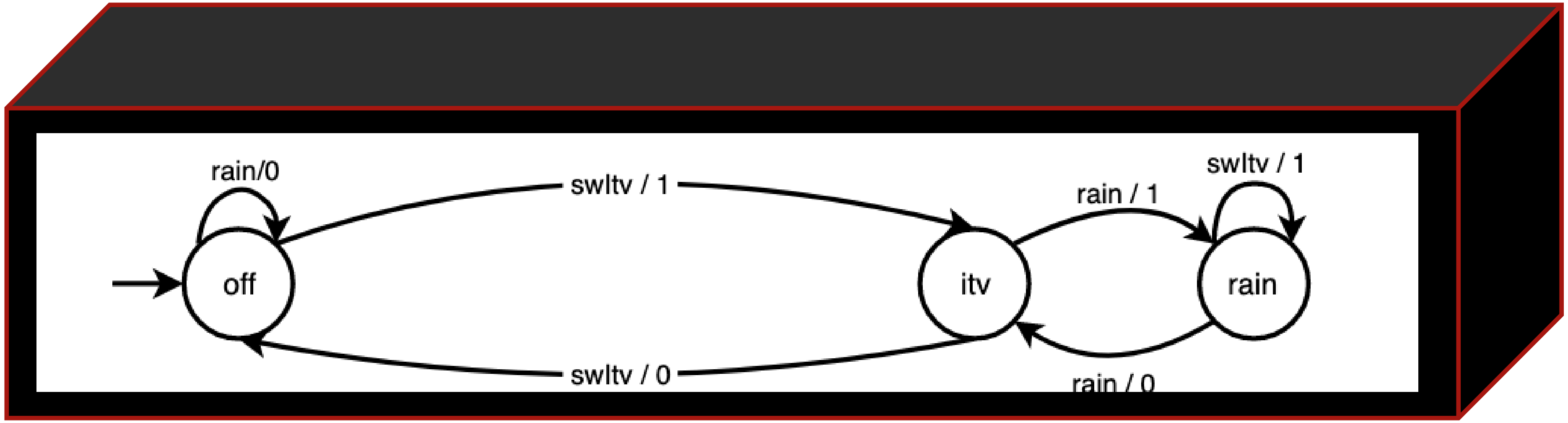
Active Learning: How?

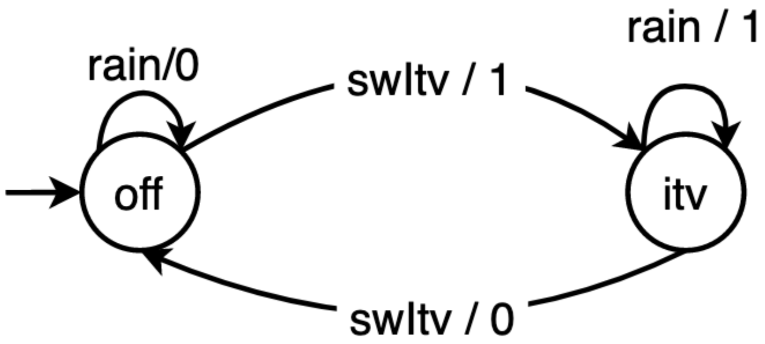




		rain	swItv
S	ϵ	0	1
$S \cdot I$	rain	0	1
	swItv	1	0

Active Learning: How?





		rain	swItv
S	ϵ	0	1
$S \cdot I$	rain	0	1
	swItv	1	0

Active Learning: How?

		$rain$	$swItv$	$rain \cdot rain$
S_r	ϵ	0	1	$0 \cdot 0$
	$swItv$	1	0	$1 \cdot 0$
	$swItv \cdot rain$	0	1	$0 \cdot 1$
$S_r \cdot I_r$	$rain$	0	1	$0 \cdot 0$
	$swItv \cdot swItv$	0	1	$0 \cdot 0$
	$swItv \cdot rain \cdot rain$	1	0	$1 \cdot 0$
	$swItv \cdot rain \cdot swItv$	0	1	$0 \cdot 1$

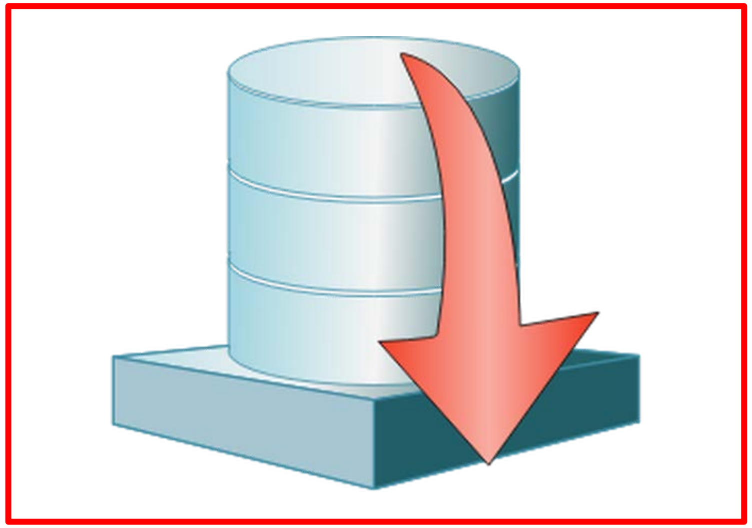
Consistent: $\forall p \in S_r . I_r \; \exists p' \in S_r \cdot p \cong p'$

Complete: $\forall p, p' \in S_r \cdot p \cong p' \Rightarrow \forall i \in I \; p.i \cong p'.i$

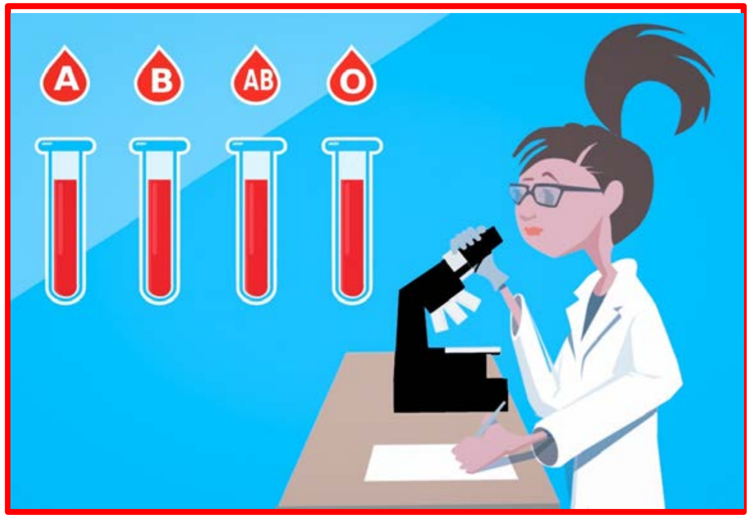
THE MESSAGE

Reusing queries makes active learning of behavioural models of a software product line more efficient.

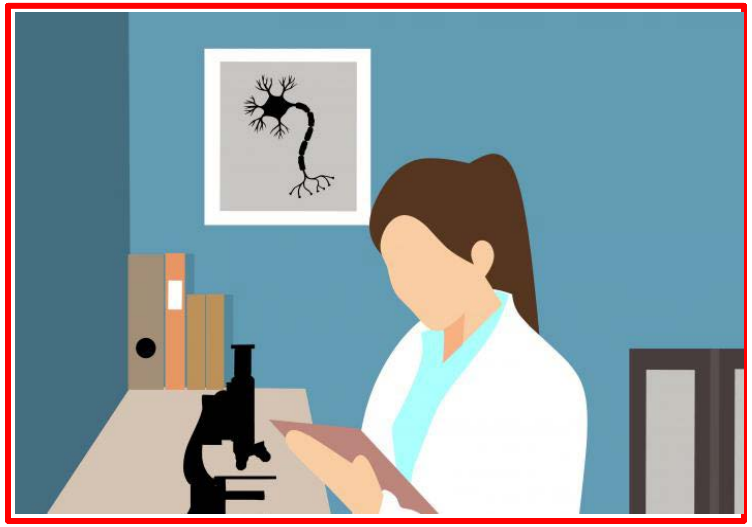
Active Learning



Query Repository

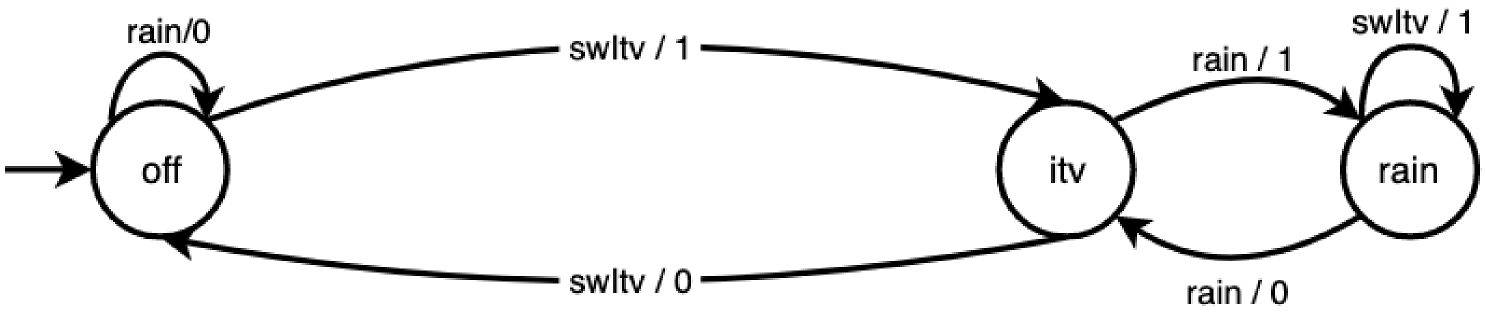


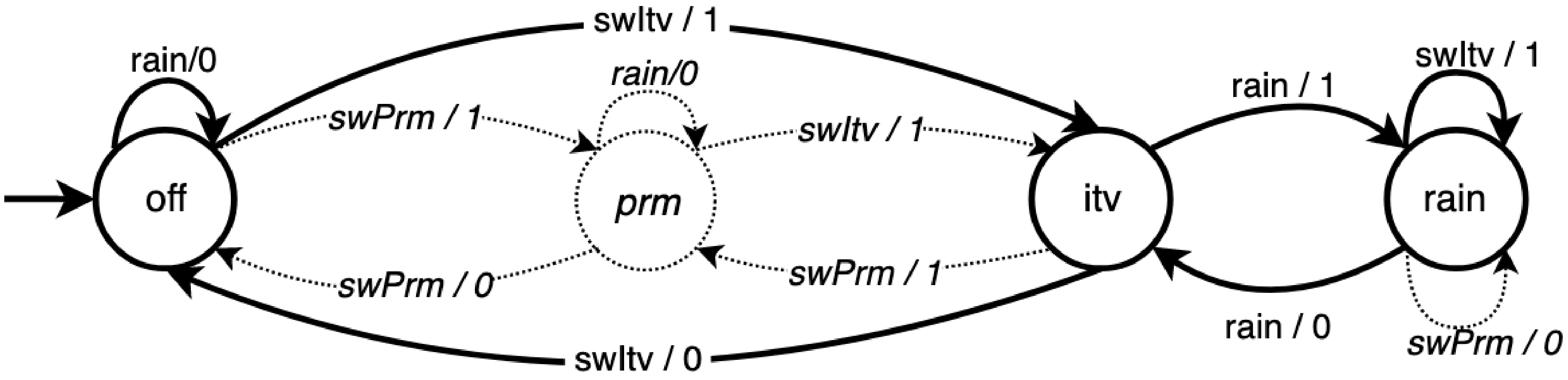
Sampling in Learning



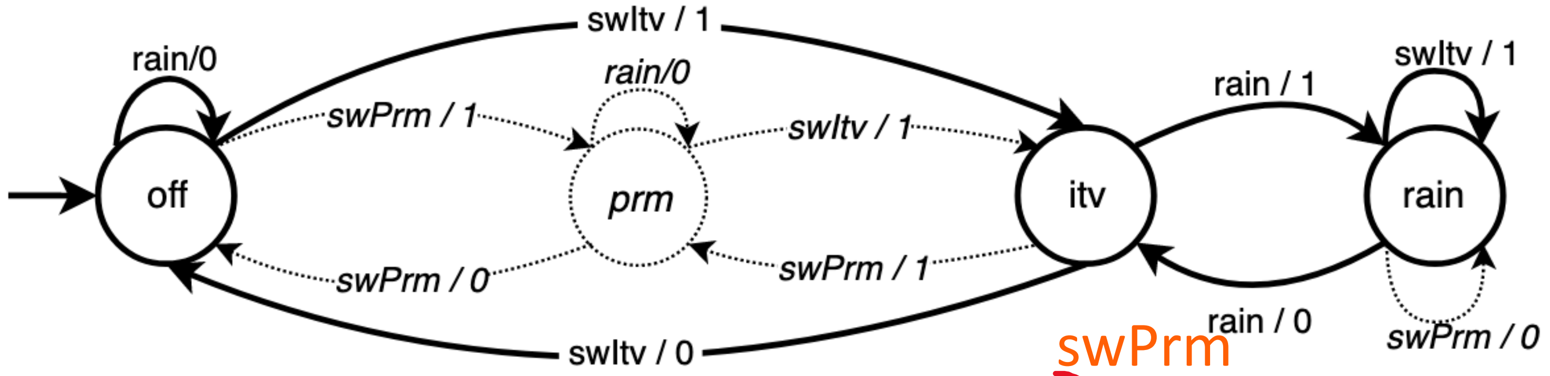
Empirical Evaluation

Learning Product Lines





Adaptive Learning



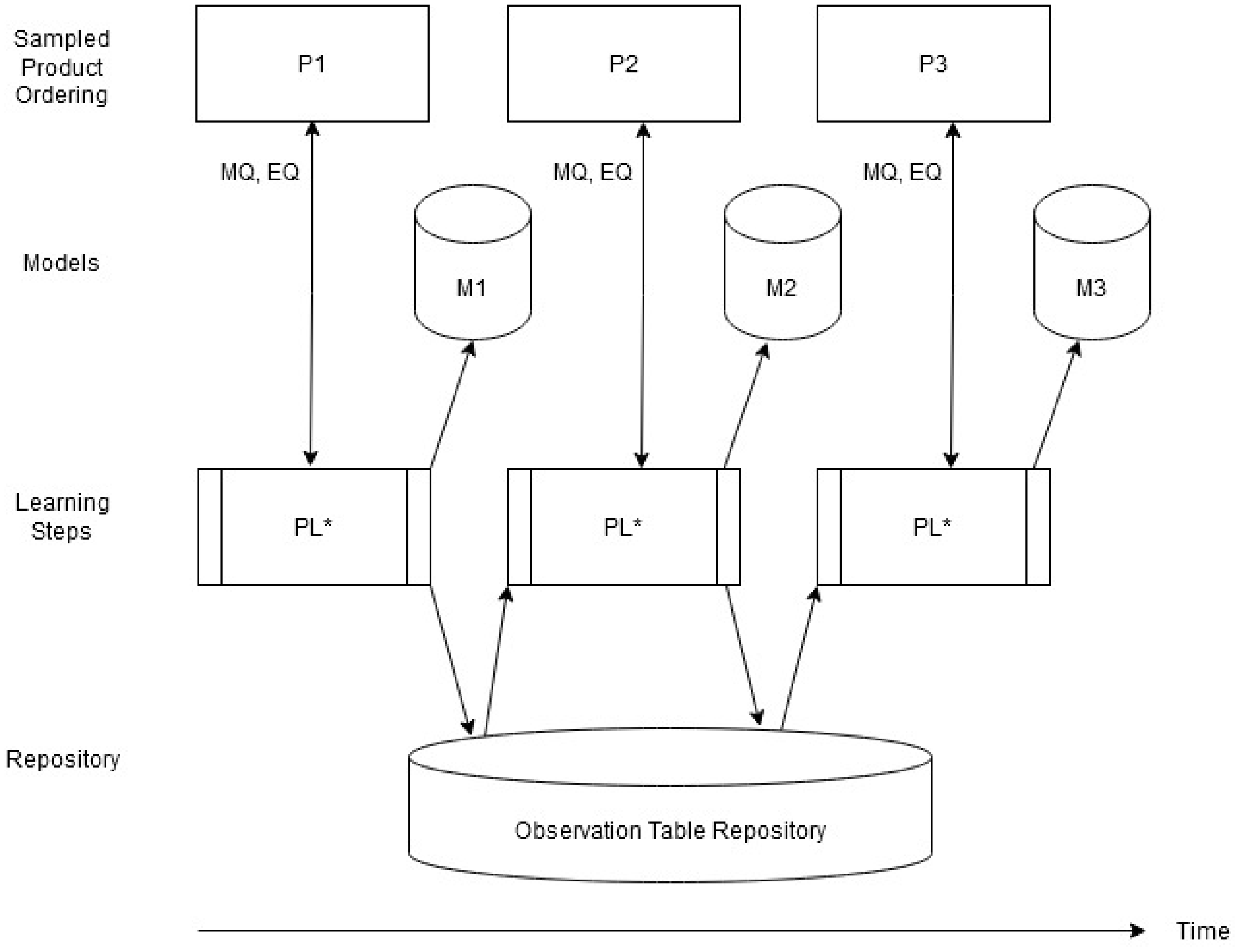
swPrm

Anything redundant?

		$rain$	$swItv$	$rain \cdot rain$
S_r	ϵ	0	1	$0 \cdot 0$
	$swItv$	1	0	$1 \cdot 0$
	$swItv \cdot rain$	0	1	$0 \cdot 1$
$S_r \cdot I_r$	$rain$	0	1	$0 \cdot 0$
	$swItv \cdot swItv$	0	1	$0 \cdot 0$
	$swItv \cdot rain \cdot rain$	1	0	$1 \cdot 0$
	$swItv \cdot rain \cdot swItv$	0	1	$0 \cdot 1$

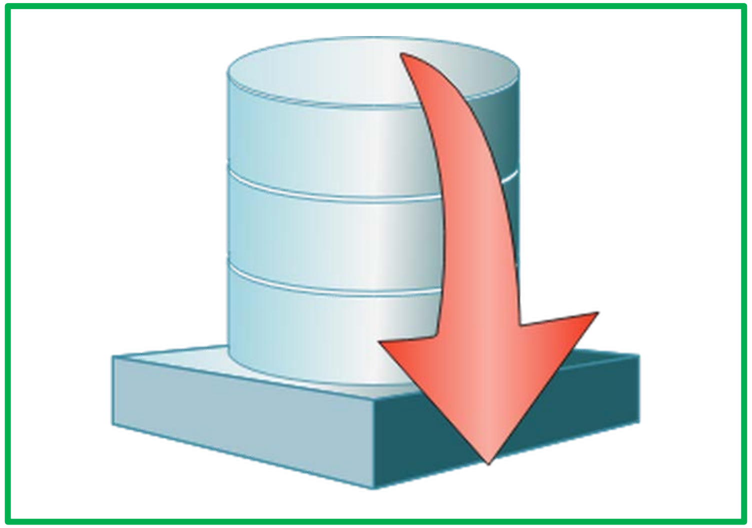
[Damasceno, M.R. Mousavi and A. Simao. Learning to Reuse: Adaptive Model Learning for Evolving Systems. iFM'19]

Query Repository

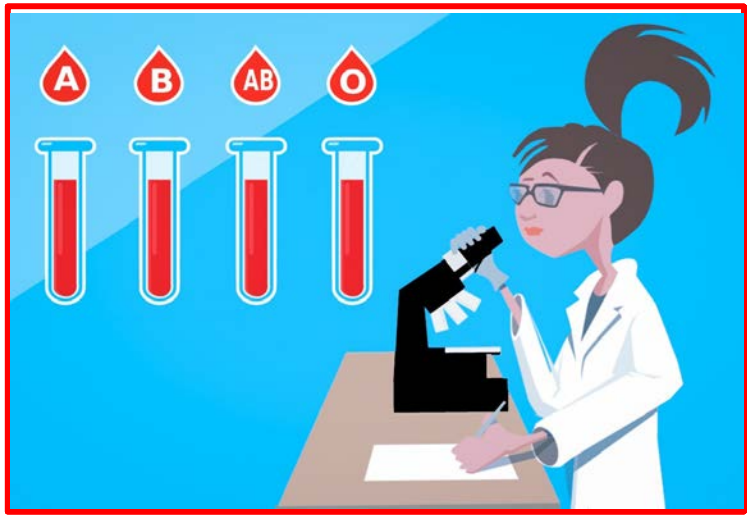


Adaptive Learning Process, Sequence of PL* Applications

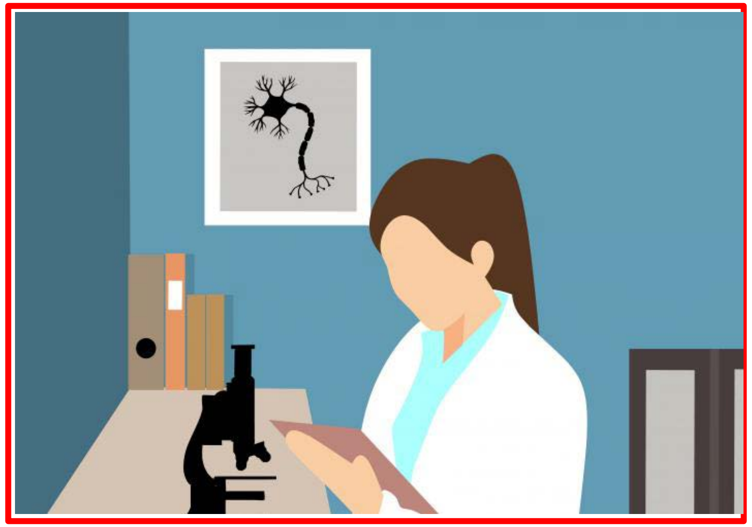
Active Learning



Query Repository



Sampling in Learning



Empirical Evaluation

THE MESSAGE

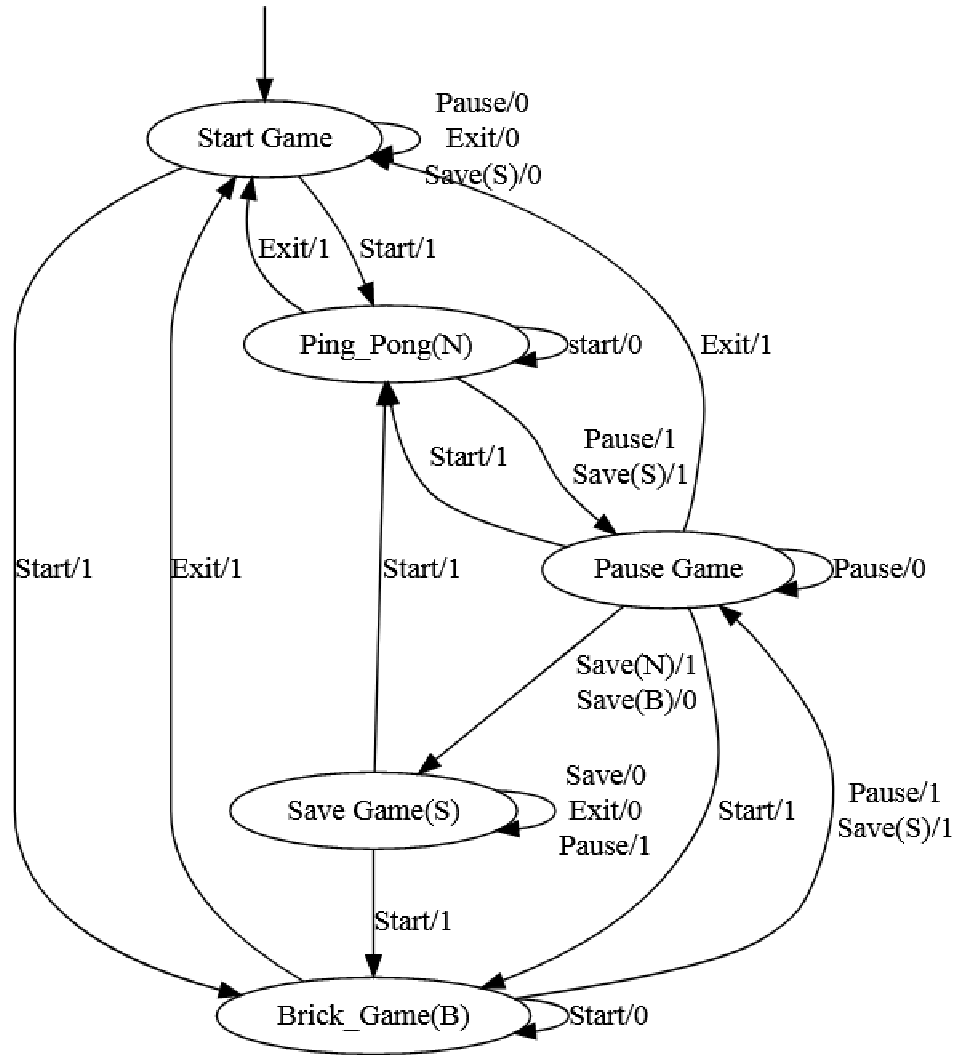
Reusing queries makes active learning of behavioural models of a software product line more efficient.

Efficiency-Precision Trade-Off

- Family model
  - A specification showing the behavior of various products
- Analysis of all products of an SPL is usually costly
- Product sampling:

Sampling → Efficiency ↑

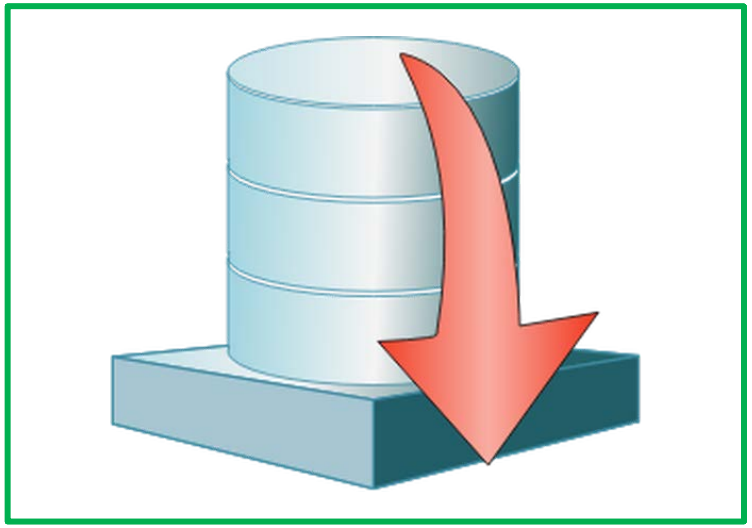
Sampling → Precision ↓



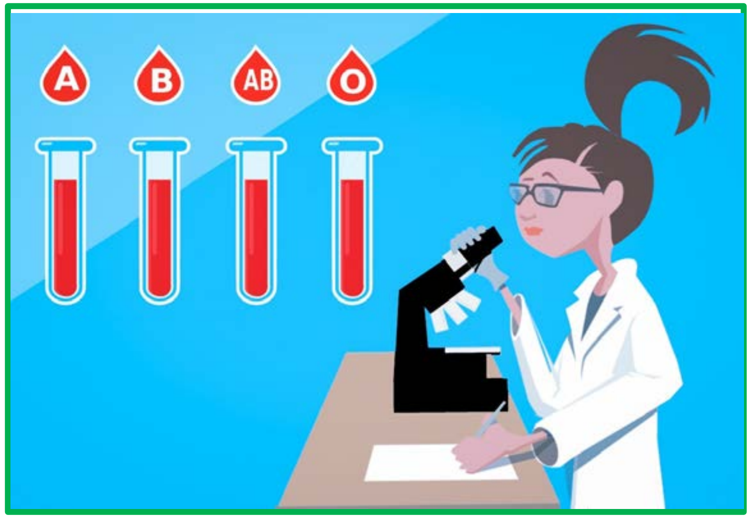
Product Sampling Methods for Software Product Lines

- Sampling algorithms
  - Greedy algorithms
  - Meta-heuristic algorithms
- Coverage criteria
  - Code coverage
  - Feature interaction coverage (T-wise sampling)

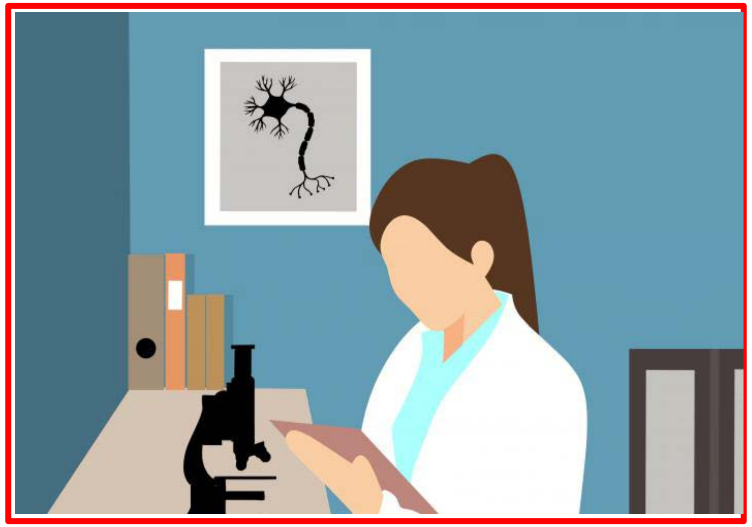
Active Learning



Query Repository



Sampling in Learning

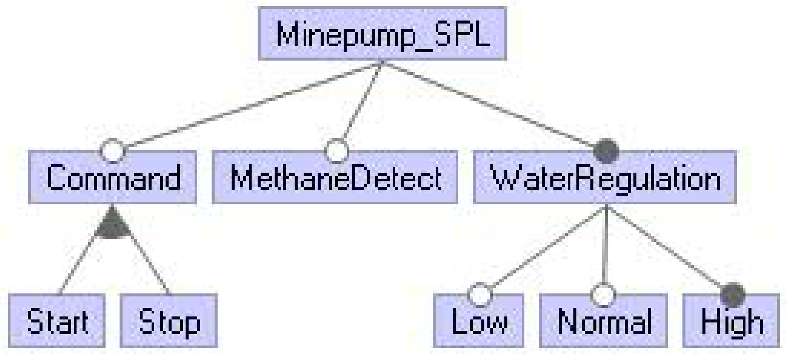


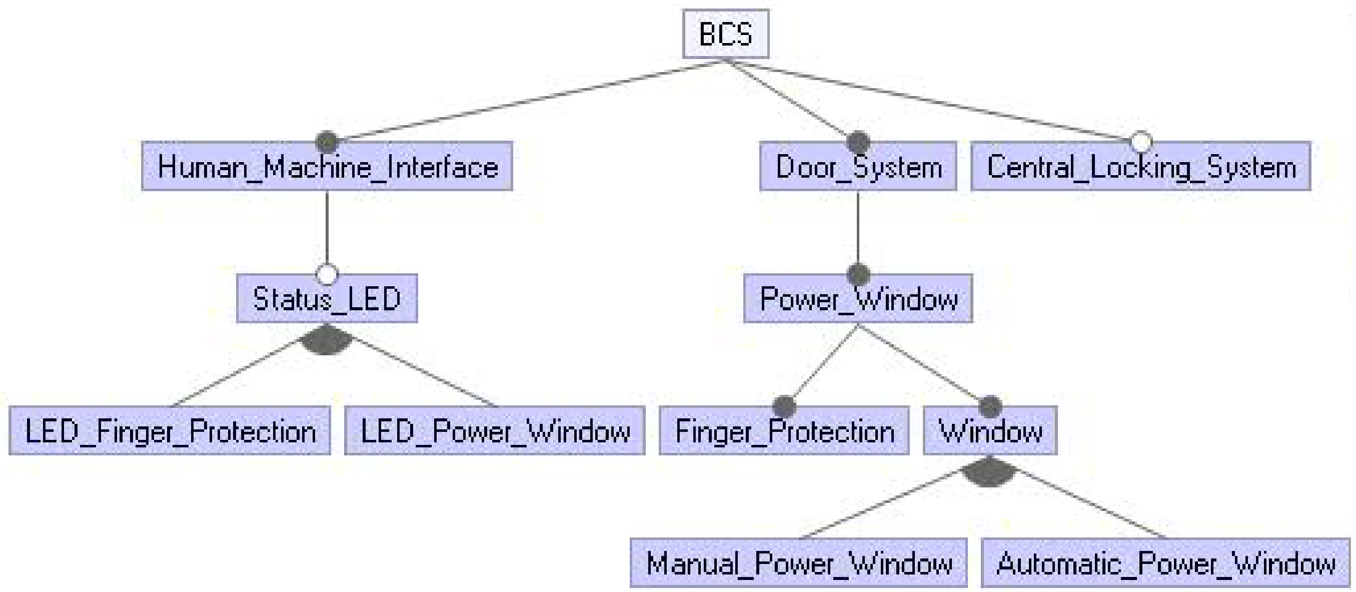
Empirical Evaluation

THE MESSAGE

Reusing queries makes active learning of behavioural models of a software product line more efficient.

Subject Systems



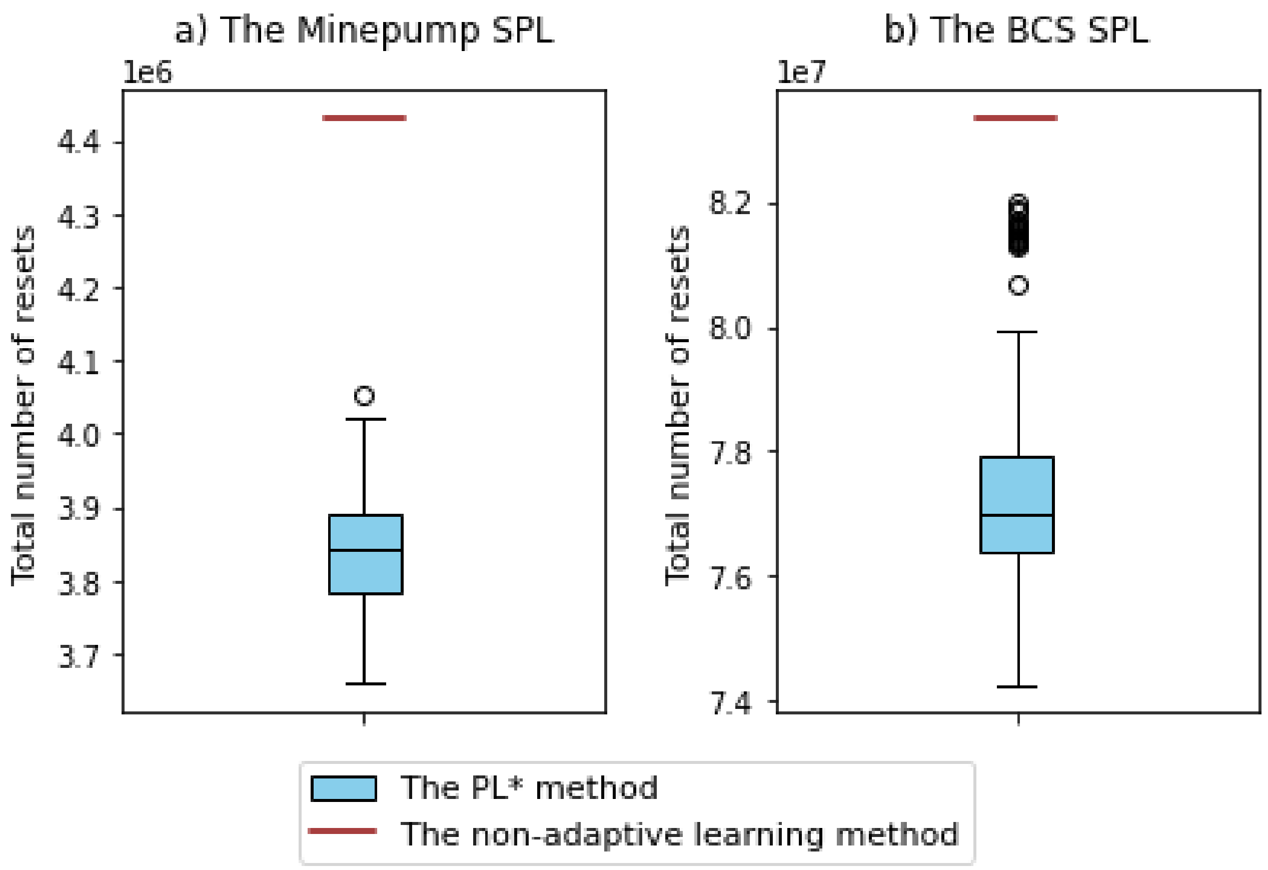


Subject system	Minimum |S|	Maximum |S|	Average |S|
The Minepump SPL	9	21	13.86
The BCS SPL	14	864	117.25

The Total Number of Rounds

SUL	PL* method, average	Non-adaptive learning method, average	Improvement percentage	p-value
The Minepump SPL	18.005	30.000	+39.9%	2.845e-204
The BCS SPL	16.910	22.000	+23.1%	7.034e-145

The Total Number of Resets



SUL	PL* method, average	Non-adaptive learning method, average	Improvement percentage	p-value
The Minepump SPL	3,838,078	4,429,400	+13.3%	1.095e-182
The BCS SPL	77,339,830	83,332,932	+7.1%	7.259e-120

The Number of Resets in MQs and EQs

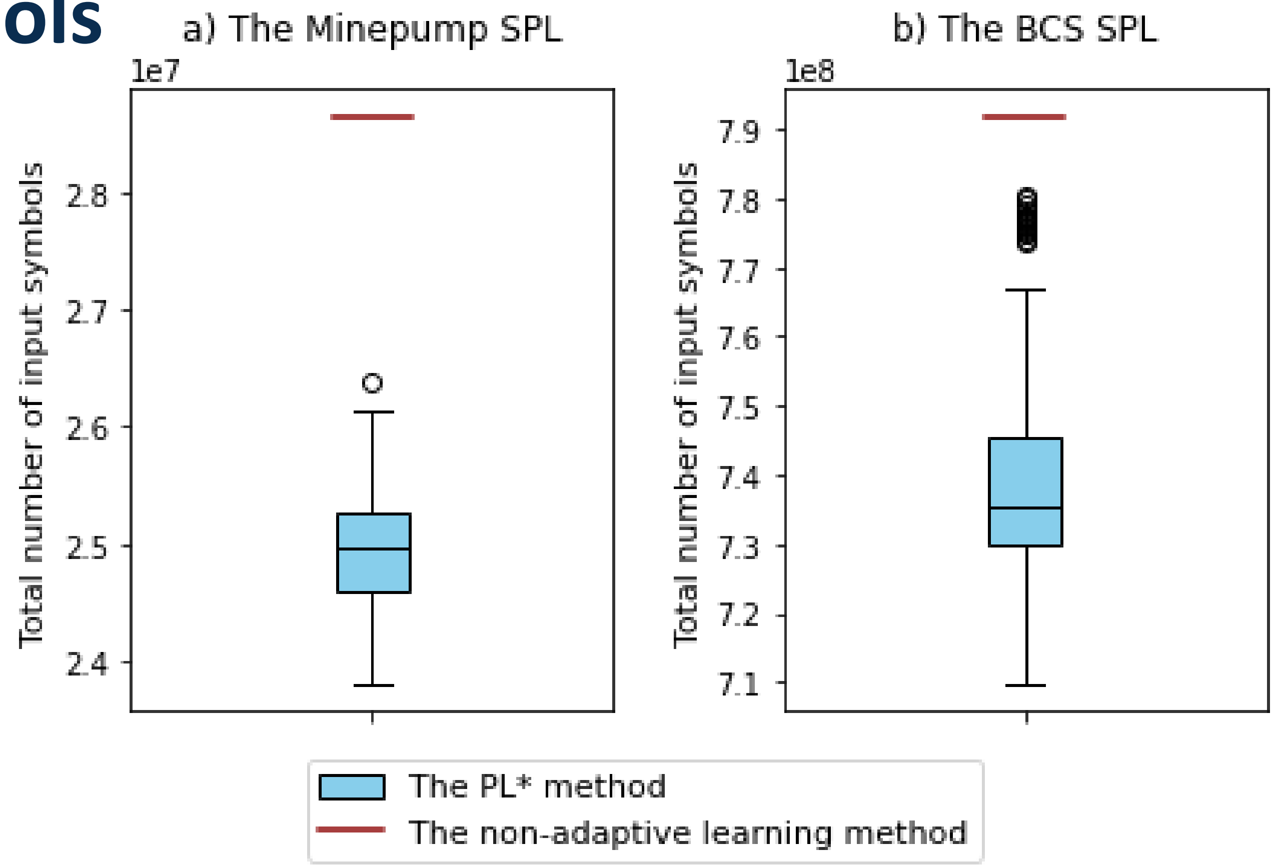
MQ resets:

SUL	PL* method, average	Non-adaptive learning method, average	Improvement percentage
The Minepump SPL	78,846	73,937	-6.7%
The BCS SPL	757,186	642,412	-17.9%

EQ resets:

SUL	PL* method, average	Non-adaptive learning method, average	Improvement percentage
The Minepump SPL	3,759,232	4,355,463	+13.6%
The BCS SPL	76,582,644	82,690,520	+7.3%

The Total Number of Input Symbols



SUL	PL* method, average	Non-adaptive learning method, average	Improvement percentage	p-value
The Minepump SPL	24,950,092	28,637,112	+12.8%	5.103e-182
The BCS SPL	739,258,253	791,674,093	+6.6%	7.150e-115

The Number of Input Symbols in MQs and EQs

MQ input symbols:

SUL	PL* method, average	Non-adaptive learning method, average	Improvement percentage
The Minepump SPL	433,967	401,613	-8.1%
The BCS SPL	5,826,720	4,804,082	-21.3%

EQ input symbols:

SUL	PL* method, average	Non-adaptive learning method, average	Improvement percentage
The Minepump SPL	24,516,124	28,235,499	+13.1%
The BCS SPL	733,431,533	786,870,011	+6.7%

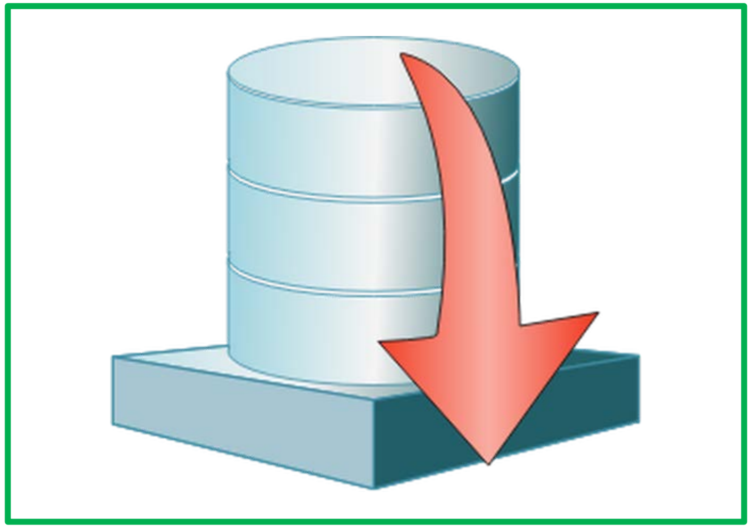
Comparing the Learning Methods

- PL* can improve the learning efficiency in terms of:
  - The total number of learning rounds
  - The total number of resets
  - The total number of input symbols

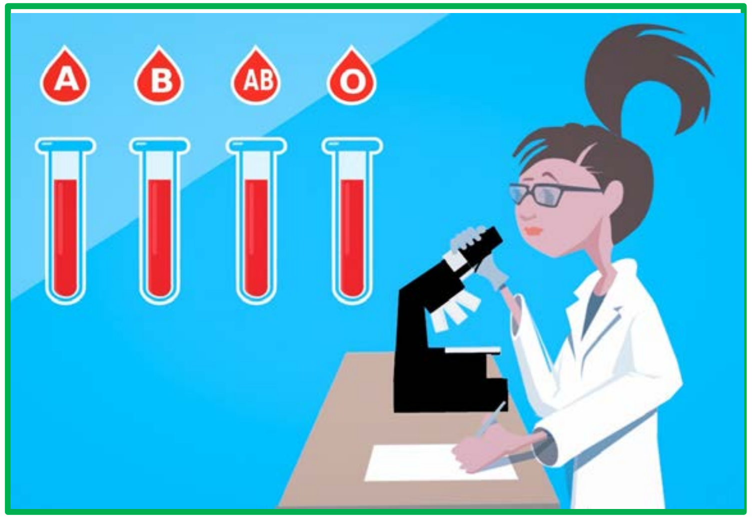
- PL* increases the number of MQs.
- PL* reduces the total cost of learning by reducing the number of EQs.

The number of EQ resets $>$ 10 $\times$ the number of MQ resets

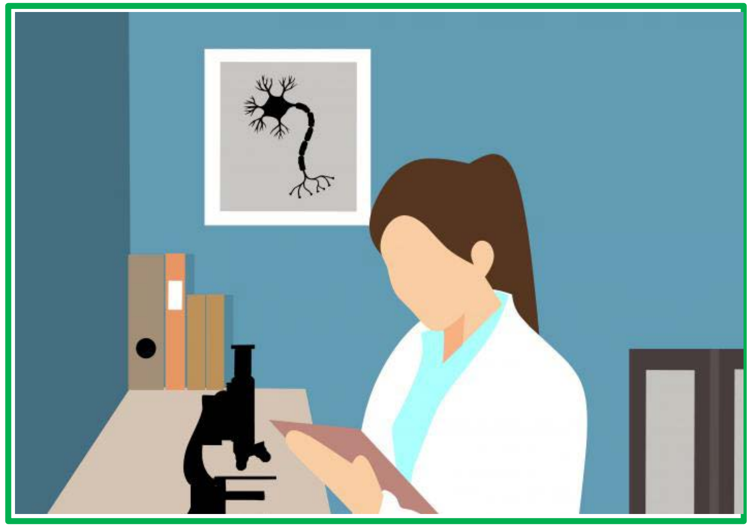
Active Learning



Query Repository



Sampling in Learning



Empirical Evaluation

THANK YOU!

Reusing queries makes active learning of behavioural models of a software product line more efficient.

ACKNOWLEDGMENTS

UKRI Engineering and Physical Sciences Research Council

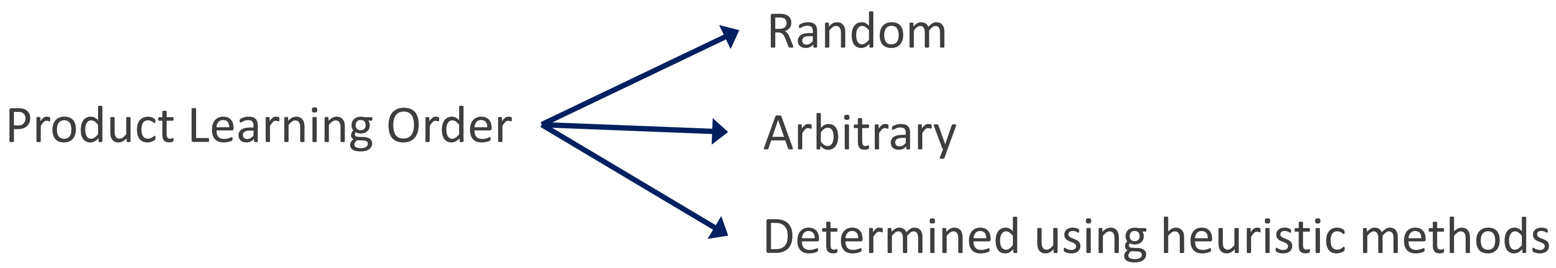
UKRI Trustworthy Autonomous Systems Hub

UKRI Verifiability Node

BACK UP SLIDES

The Product Learning Order

- **Observation: the product learning order can affect the efficiency of the PL* method.**



- Based on the results of the experiments, a heuristic is presented to determine the desired learning orders.

The Product Learning Order

- **Observation: when the number of new non-mandatory features that are added simultaneously is small, the efficiency of the PL* method increases.**

$O = \langle p_1, p_2, \ldots, p_n \rangle$ → A product learning order (a sequence of products in a sample)

$i < j$ → The FSM of p_i must be learned earlier than the FSM of p_j

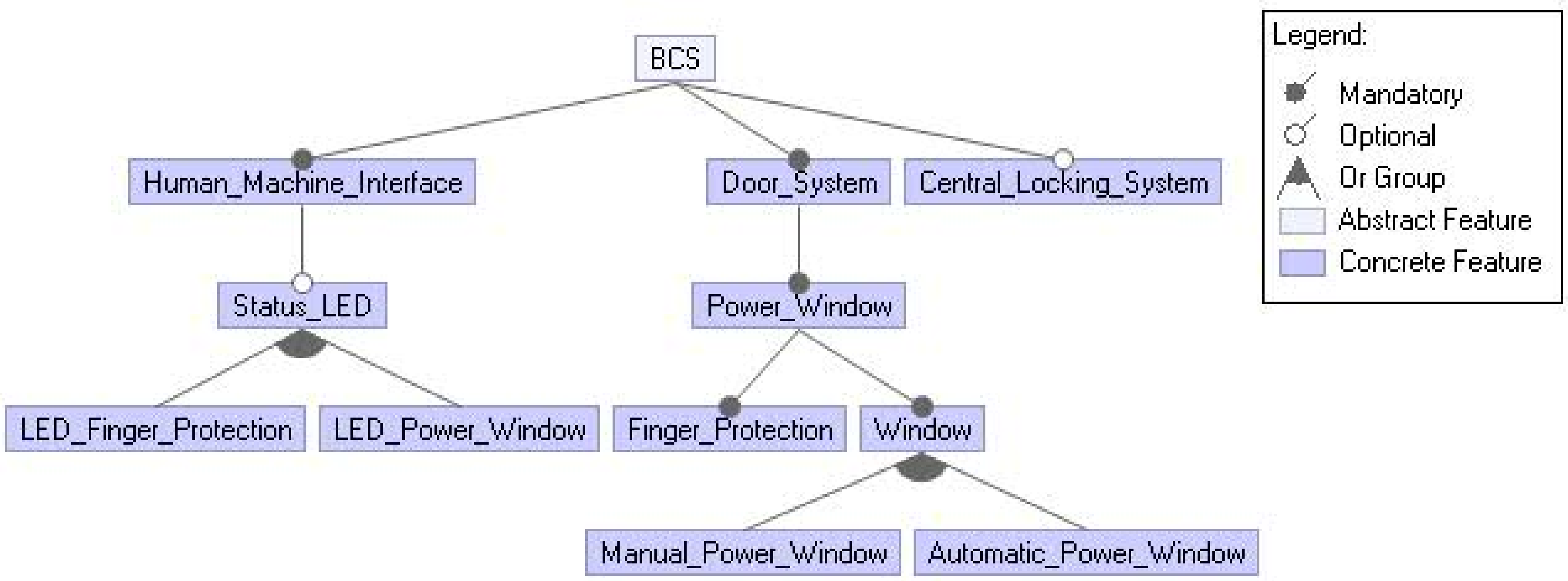
$$D_i = \begin{cases} 0, & \text{if } F_i = 0 \\ \frac{1}{F_i}, & \text{if } F_i \neq 0 \end{cases}$$

$$D = \sum_{i=1}^{n} D_i$$

Why $\frac{1}{F_i}$ is used?

F_i → **The number of new non-mandatory features added by p_i**

An Example from BCS SPL



Number of non-mandatory features: 6

16 products are sampled from BCS SPL using 3-wise sampling method.

High efficiency:

Learning order: $\langle 3, 1, 7, 6, 15, 9, 11, 10, 16, 13, 4, 14, 12, 2, 5, 8 \rangle$

Status_LED
LED_Finger_Protection
Manual_Power_Window

$F_1 = 3$

Automatic_Power_Window

$F_2 = 1$

Manual_Power_Window
Automatic_Power_Window
Central_Locking_System

$F_3 = 1$

$F_4 = 1$

Status_LED
LED_Power_Window
Manual_Power_Window
Automatic_Power_Window

The total number of resets (PL*): $\mathbf{75.1 \times 10^6}$

$$\mathbf{D = \frac{1}{3} + \frac{1}{1} + \frac{1}{1} + \frac{1}{1} \approx 3.33}$$

Low efficiency:

Learning order: $\langle 10, 12, 11, 8, 6, 15, 16, 13, 4, 3, 14, 5, 9, 7, 2, 1 \rangle$

Status_LED
LED_Finger_Protection
LED_Power_Window
Manual_Power_Window
Automatic_Power_Window
Central_Locking_System

$F_1 = 6$

The total number of resets (PL*): $\mathbf{82.0 \times 10^6}$

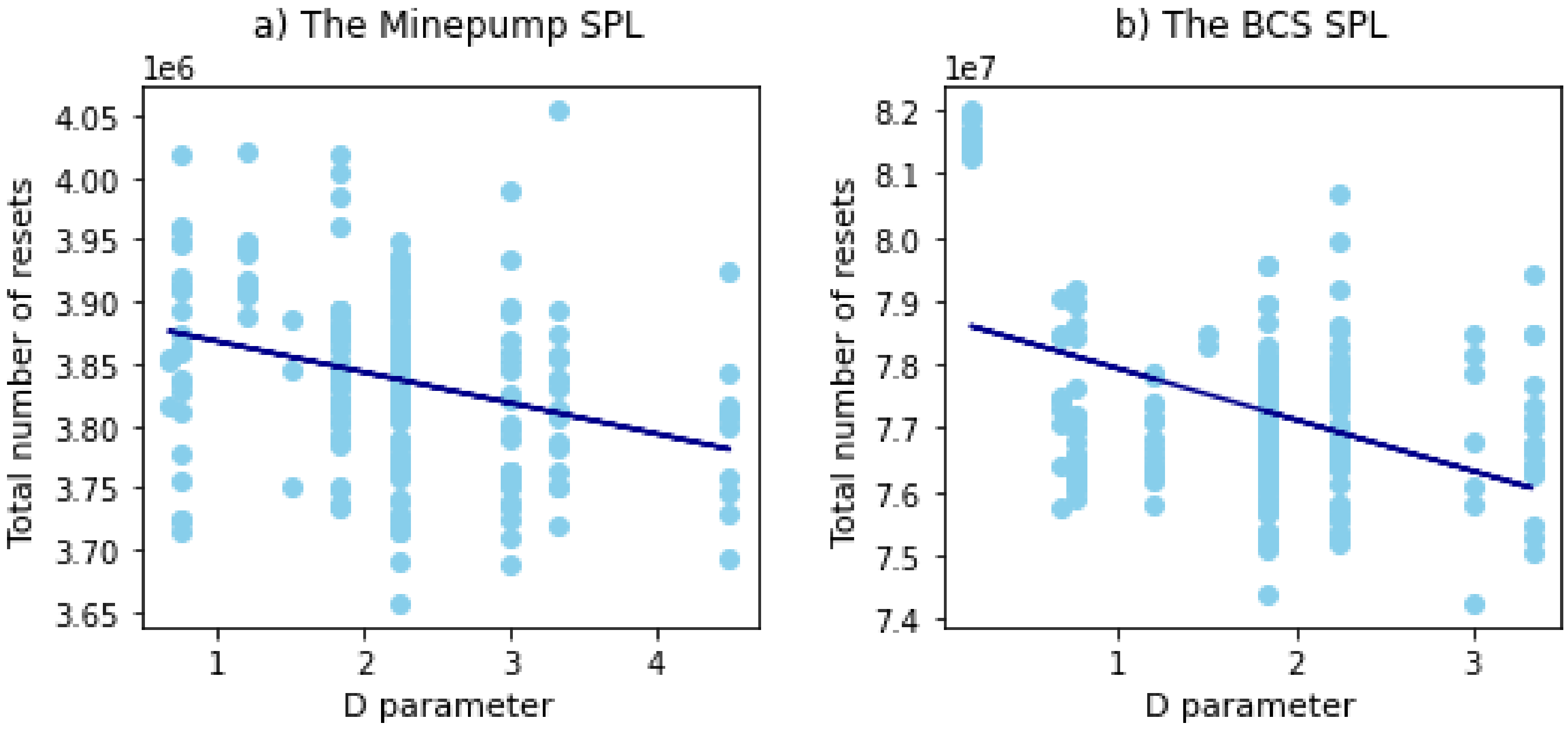
$$\mathbf{D = \frac{1}{6} \approx 0.17}$$

More Examples

Learning order	F_1	F_2	F_3	F_4	F_5	F_6	F_7	Total resets ($\times 10^6$)	D
$\langle 3, 1, 7, 6, 15, 9, 11, 10, 16, 13, 4, 14, 12, 2, 5, 8 \rangle$	3	1	1	1	0	0	0	75.1	3.33
$\langle 2, 11, 12, 3, 16, 6, 10, 5, 14, 8, 1, 7, 15, 9, 4, 13 \rangle$	4	1	1	0	0	0	0	76.6	2.25
$\langle 7, 16, 2, 15, 11, 3, 6, 12, 13, 10, 14, 9, 8, 1, 4, 5 \rangle$	3	2	0	0	0	1	0	77.3	1.83
$\langle 10, 12, 11, 8, 6, 15, 16, 13, 4, 3, 14, 5, 9, 7, 2, 1 \rangle$	6	0	0	0	0	0	0	82.0	0.16

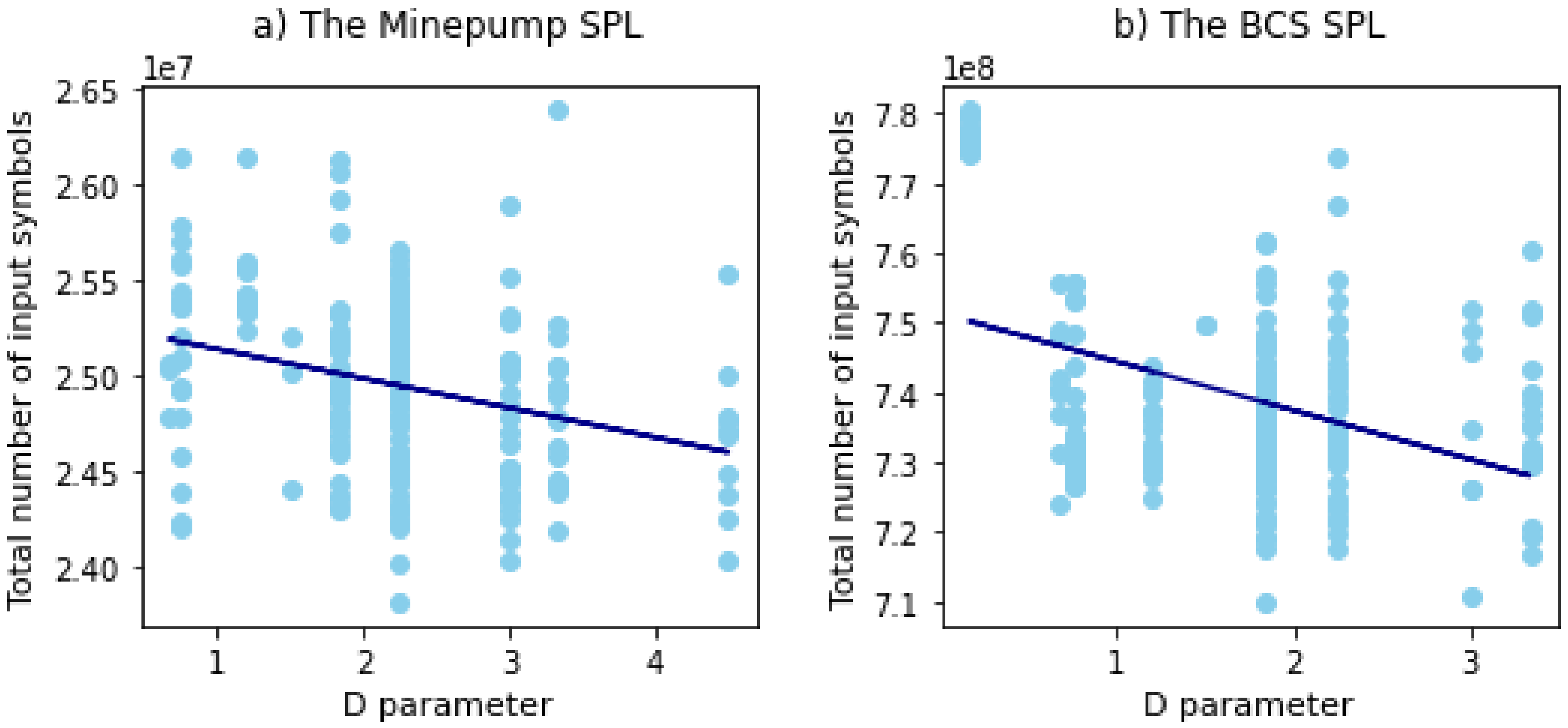
The total number of resets in non-adaptive learning method: 83.0×10^6

Correlation Between Parameter D and the Number of Resets



SUL	r	p-value
The Minepump SPL	-0.305	1.127e-05
The BCS SPL	-0.431	1.963e-10

Correlation Between Parameter D and the Number of Input Symbols



SUL	r	p-value
The Minepump SPL	-0.301	1.484e-05
The BCS SPL	-0.403	3.429e-09