Learning by Sampling:
Learning behavioral family models from software product lines

Access the paper!

ICIS Seminar

Carlos Diego Nascimento Damasceno, Mohammad Reza Mousavi, Adenilso Simao

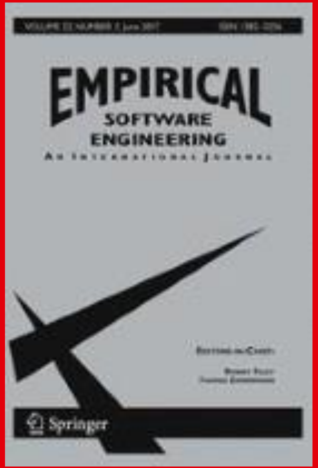
Journal paper published at the Empirical Software Engineering Journal

PhD research at University of Sao Paulo and University of Leicester

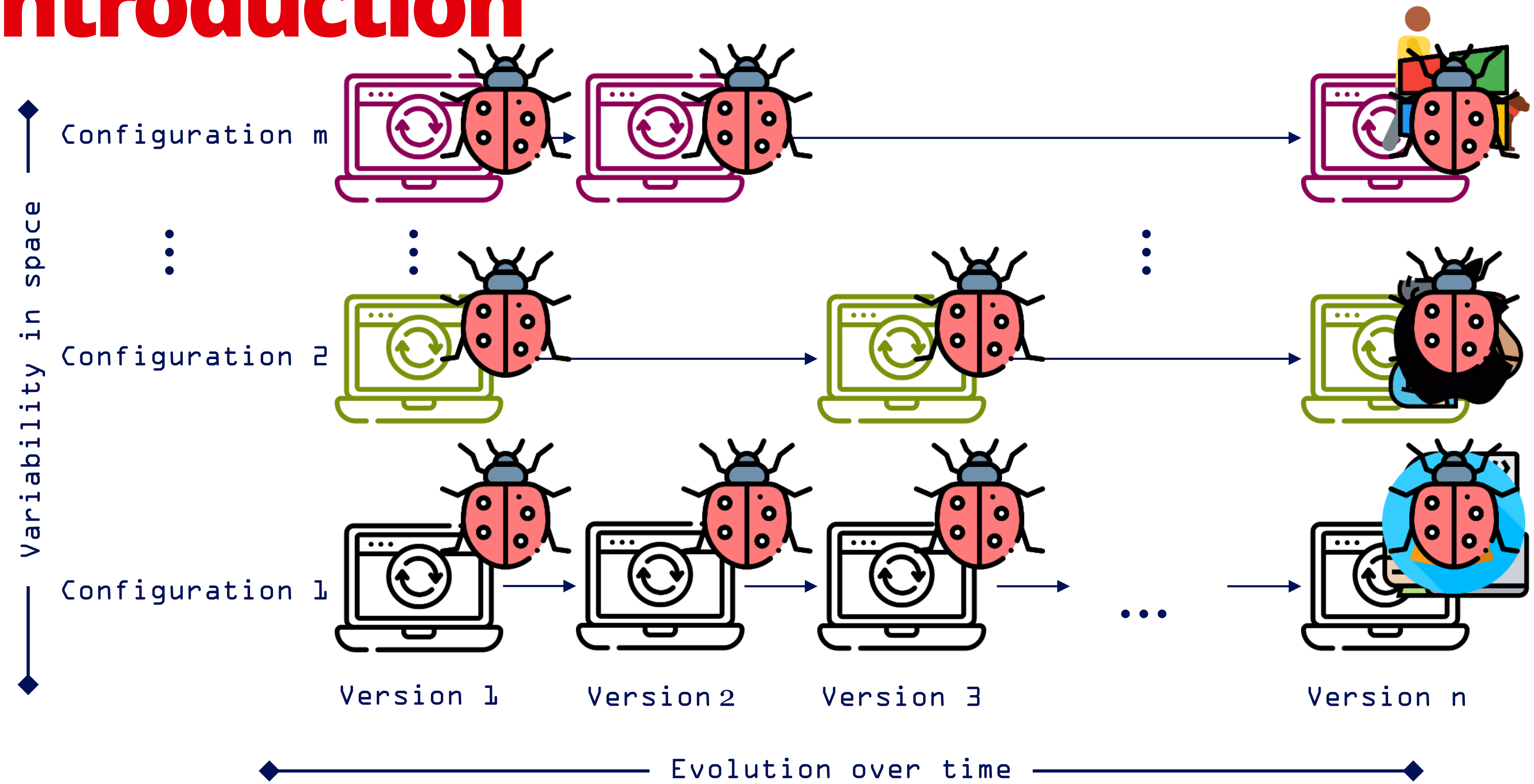
Universidade de São Paulo

UNIVERSITY OF LEICESTER

Radboud University



Introduction



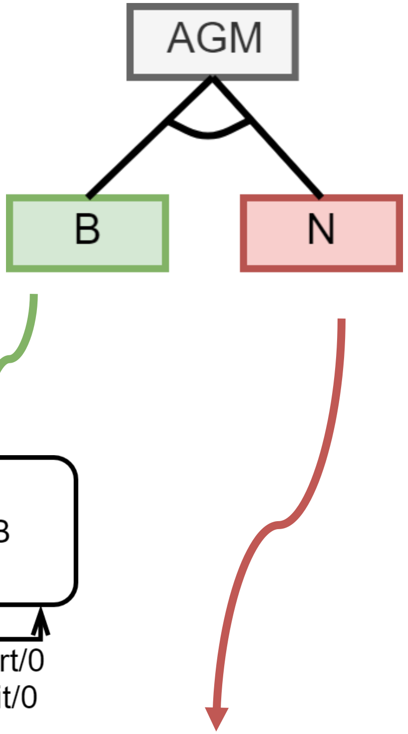
K. Pohl, G. Böckle, and F. J. van der Linden, Software Product Line Engineering: Foundations, Principles and Techniques. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2005.
H. Chockler, D. Kroening, L. Mariani, and N. Sharygina, Eds., Validation of Evolving Software. Cham: Springer International Publishing, 2015.
M. S. Deutsch, 'Tutorial Series 7 Software Project Verification and Validation', Computer, vol. 14, no. 4, pp. 54–70, Apr. 1981

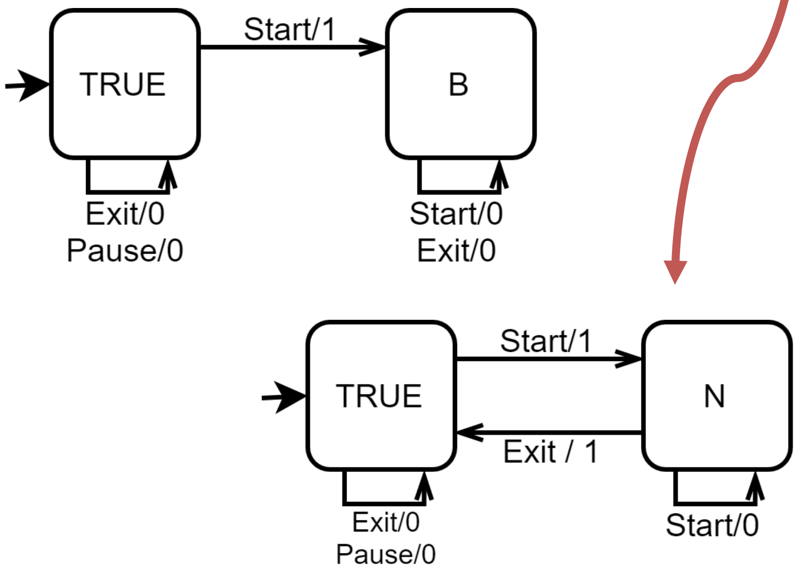
Context

Analysis and modeling of SPLs

Product-based strategies

- Missing models
- Redundant analysis
- Scalability (e.g., exponential)





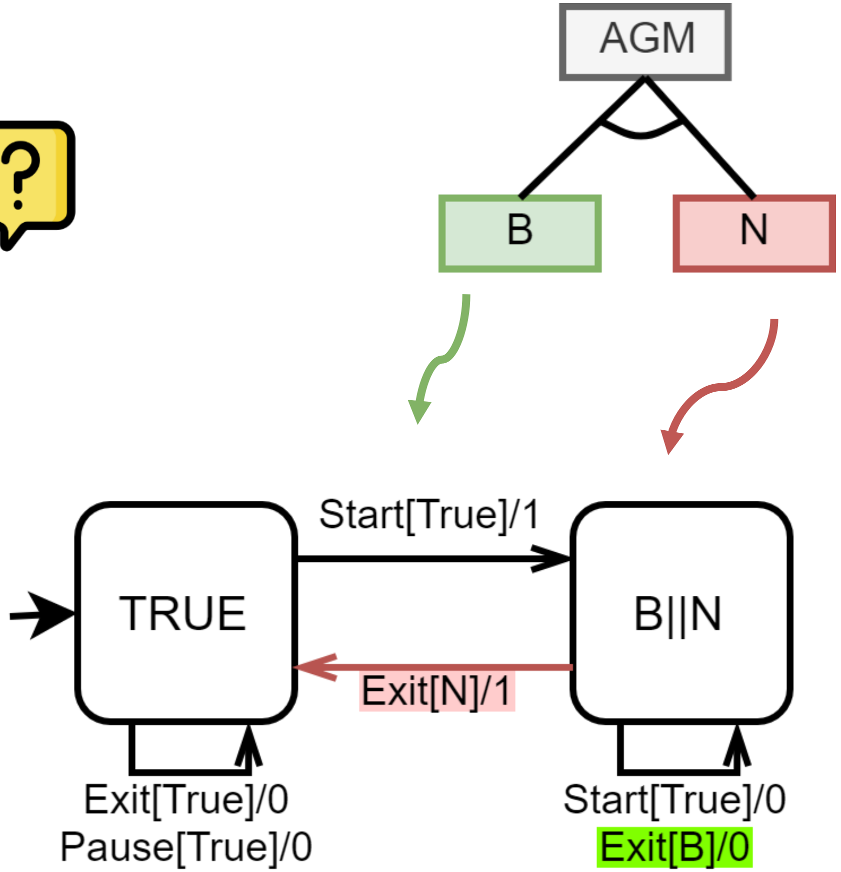
T. Thüm, S. Apel, C. Kästner, I. Schaefer, and G. Saake, ‘A Classification and Survey of Analysis Strategies for Software Product Lines’, ACM Comput. Surv., vol. 47, Jun. 2014
S. Oster, ‘Feature Model-based Software Product Line Testing’, PhD Thesis, Technische Universität, Darmstadt, 2012.

Context

Analysis and modeling of SPLs

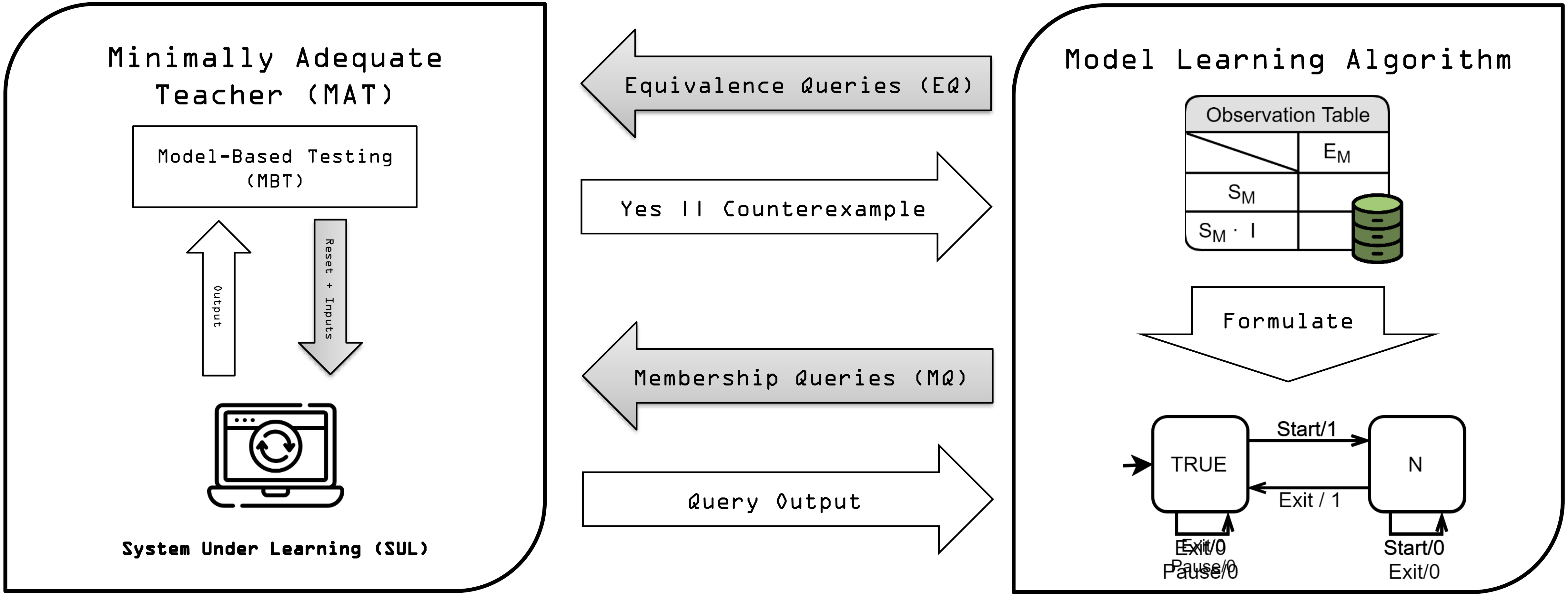
Family-based strategies

- Missing family models
- Model maintenance and evolution
- Commonalities/variabilities are unknown



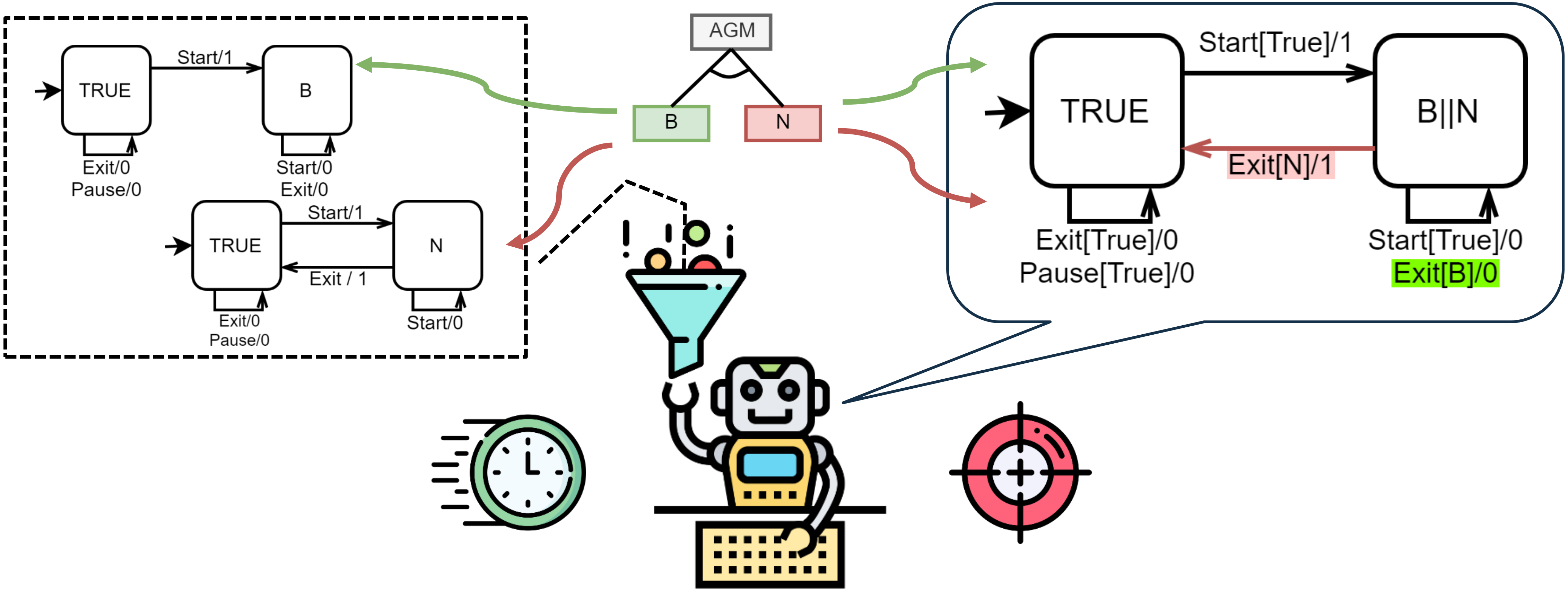
T. Thüm, S. Apel, C. Kästner, I. Schaefer, and G. Saake, 'A Classification and Survey of Analysis Strategies for Software Product Lines', ACM Comput. Surv., vol. 47, no. 1, p. 6:1–6:45, Jun. 2014
S. Oster, 'Feature Model-based Software Product Line Testing', PhD Thesis, Technische Universität, Darmstadt, 2012.
V. Hafemann Fragal, A. Simao, and M. R. Mousavi, 'Validated Test Models for Software Product Lines: Featured Finite State Machines', in `FACS 2016.

Context



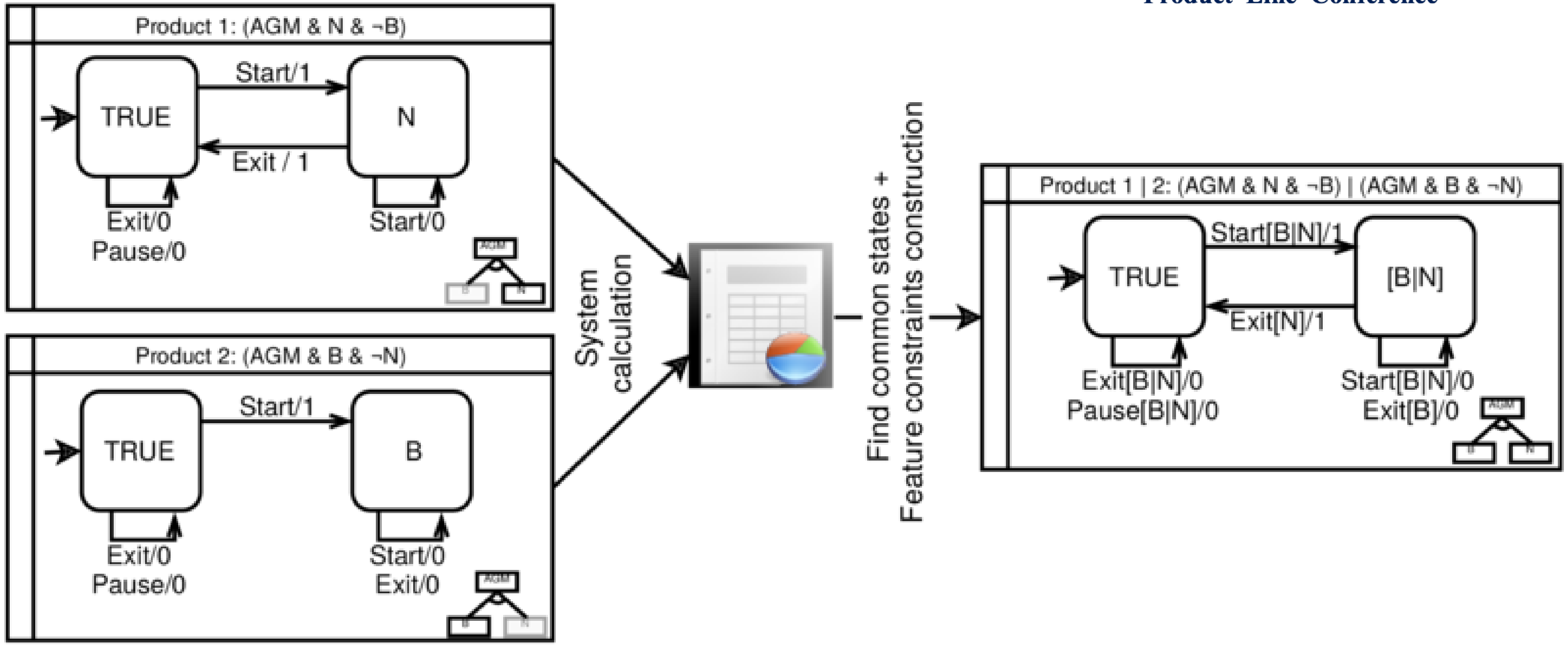
D. Angluin, ‘Learning regular sets from queries and counterexamples’, Information and Computation, vol. 75, no. 2, pp. 87–106, 1987
F. Vaandrager, ‘Model Learning’, Commun. ACM, vol. 60, no. 2, pp. 86–95, Jan. 2017
C. D. N. Damasceno, M. R. Mousavi, and A. Simao, ‘Learning to reuse: Adaptive model learning for evolving systems’, in 15th Integrated Formal Methods, Bergen, Norway, 2019

Research Problem



How can we ***leverage model learning*** *concepts to the task of* ***behavioral variability modeling?***
Can we obtain models precise enough if we ***sample configurations****?*

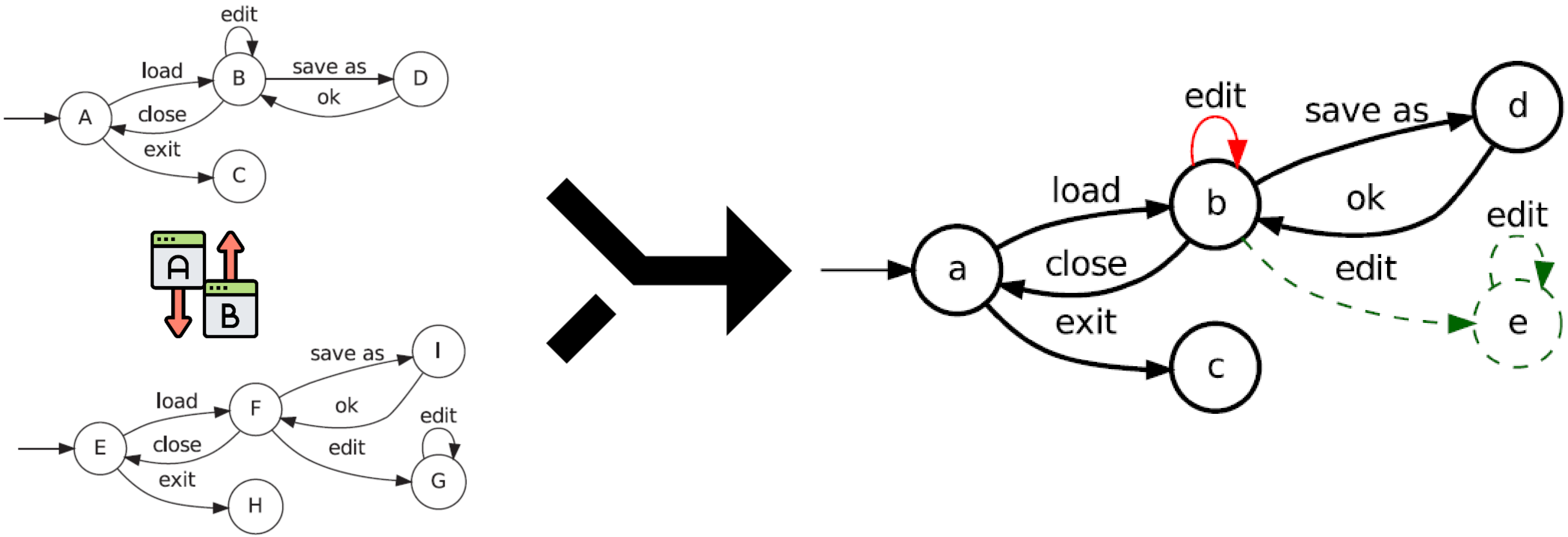
FFSM Difference (FFSM $_{Diff}$)



The FFSM $_{Diff}$ can learn FFSMs from a product models by employing state-based model comparison and express product-specific behaviors with feature constraints using feature model analysis

C. D. N. Damasceno, M. R. Mousavi, and A. Simao, 'Learning from Difference: An Automated Approach for Learning Family Models from Software Product Lines', in 23rd International Systems and Software Product Line Conference, 2019

State-based model comparison (LTS Diff algorithm)



Comparing the Structures of Two State Machines of a Text Editor

N. Walkinshaw and K. Bogdanov, 'Automated Comparison of State-Based Software Models in Terms of Their Language and Structure', ACM TOSEM, vol. 22, 2013

State-based model comparison (LTS Diff algorithm)

$$S^G_{Succ}(a,b) = \frac{1}{2}\frac{\sum_{(c,d,i,o)\,\in\, Succ_{a,b}}(1 + k \times S^G_{Succ}(c,d))}{|\sum^{out}_r(a) - \sum^{out}_u(b)| + |\sum^{out}_r(b) - \sum^{out}_u(a)| + \boxed{|Succ_{a,b}|}}$$

Figure: Global similarity score [4]

Global similarity score (Outgoing and incoming transitions)

- Pairwise similarity based on surrounding matching transitions and connected state pairs.
- Attenuation ratio k gives precedence to the closest state pairs.
- Matching transitions and distinct transitions.

N. Walkinshaw and K. Bogdanov, 'Automated Comparison of State-Based Software Models in Terms of Their Language and Structure', ACM TOSEM, vol. 22, 2013

State-based model comparison (LTS Diff algorithm)

$$S^G_{Succ}(Pa, Pa) = \frac{1}{2} \times \frac{3 + k \times [S^G_{Succ}(St, St) + S^G_{Succ}(Bo, Po) + S^G_{Succ}(Pa, Pa)]}{0 + 0 + 3} = 0.58$$

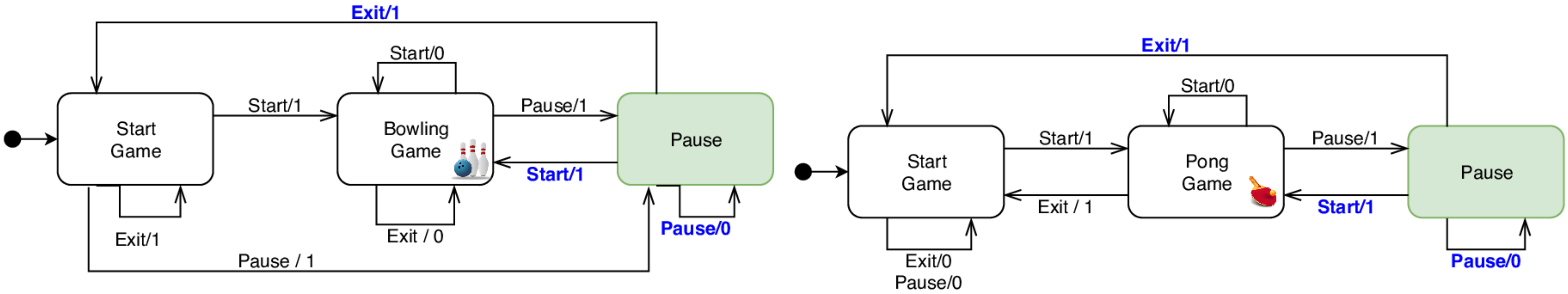


Figure: Two examples of product FSMs and their similarity scores

State-based model comparison (LTS $_{\text{Diff}}$

Pair	(St,St)	(St,Po)	(St,Pa)	(Bo,St)	(Bo,Po)	(Bo,Pa)	(Pa,St)	(Pa,Po)	(Pa,Pa)	#Match
(St,St)	10.0	0.0	0.0	0.0	-0.5	0.0	0.0	0.0	0.0	1
(St,Po)	-0.5	8.0	0.0	0.0	0.0	0.0	0.0	0.0	-0.5	2
(St,Pa)	-0.5	0.0	8.0	0.0	-0.5	0.0	0.0	0.0	0.0	2
(Bo,St)	0.0	0.0	0.0	9.5	0.0	0.0	0.0	0.0	0.0	1
(Bo,Po)	0.0	0.0	0.0	0.0	7.5	0.0	0.0	0.0	-0.5	2
(Bo,Pa)	0.0	0.0	0.0	0.0	0.0	12.0	0.0	0.0	0.0	0
(Pa,St)	0.0	0.0	0.0	0.0	-0.5	0.0	7.5	0.0	0.0	2
(Pa,Po)	-0.5	0.0	0.0	0.0	0.0	0.0	0.0	10.0	0.0	1
(Pa,Pa)	-0.5	0.0	0.0	0.0	-0.5	0.0	0.0	0.0	5.5	3

Table 1: Illustration of a system of linear equations

The FFSM $_{\text{Diff}}$ algorithm

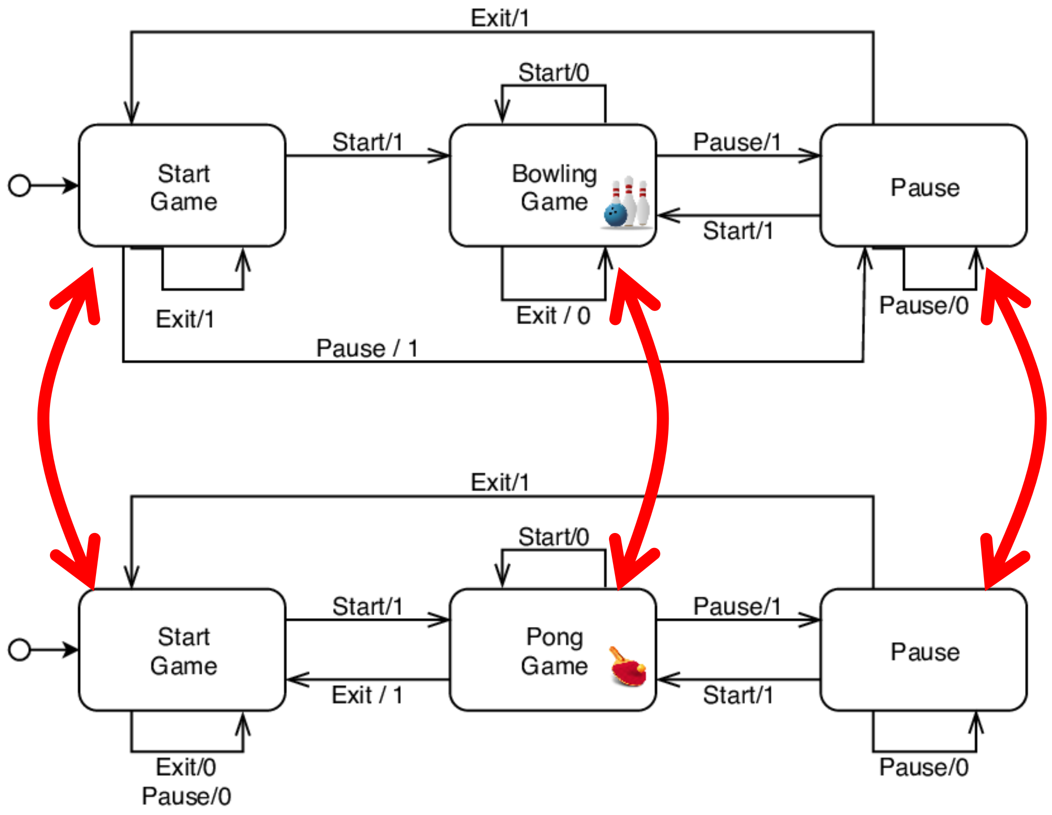


Figure: Two examples of product FSMs

$\text{pair}(\text{St}, \text{St}) = 0.12$

$\text{pair}(St, Po) = 0.29$

$\text{pair}(St, Pa) = 0.28$

$\text{pair}(Bo, St) = 0.11$

$\text{pair}(\text{Bo}, \text{Po}) = 0.31$

$\text{pair}(Bo, Pa) = 0$

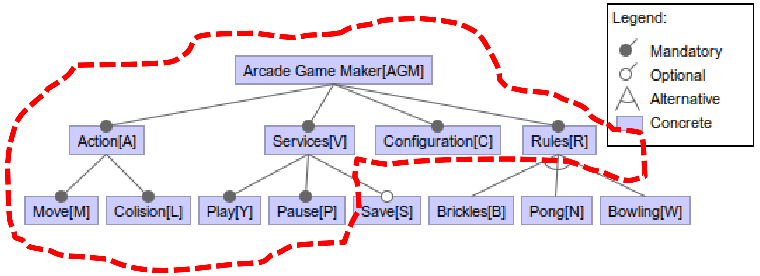
$\text{pair}(Pa, St) = 0.29$

$\text{pair}(Pa, Po) = 0.11$

$\text{pair}(\text{Pa}, \text{Pa}) = 0.58$

Figure: Pairwise state similarity

The $FFSM_{Diff}$ algorithm



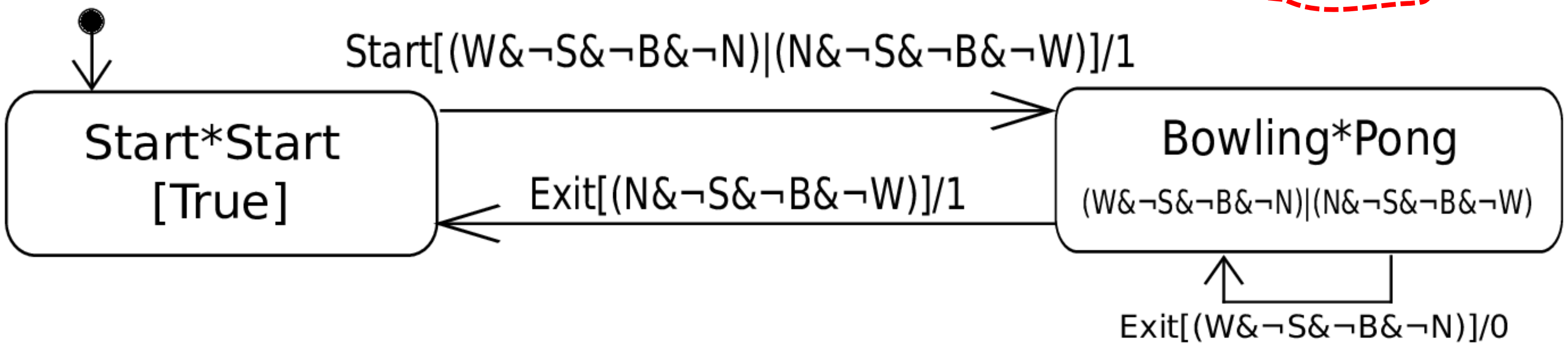


Figure: Fragment of the FFSM learnt from two products of the AGM SPL.

Simplified configuration – Example

$$\rho_{Bowling} = (W \land \neg S \land \neg B \land \neg N)$$
$$\rho_{Pong} = (N \land \neg S \land \neg B \land \neg W)$$

V. Hafemann Fragal, A. Simao, and M. R. Mousavi, 'Validated Test Models for Software Product Lines: Featured Finite State Machines', in `FACS 2016.
D. Benavides, S. Segura, and A. Ruiz-Cortés, 'Automated analysis of feature models 20 years later: A literature review', Information Systems, vol. 35, no. 6, pp. 615–636, 2010

EMPIRICAL EVALUATION

Research Questions

RQ1) Effectiveness on learning succinct family models, given the total size of the product pairs under learning

RQ2) Size of learned family models vs. configuration similarity

RQ3) Effectiveness in learning succinct family models, given the total size of the hand-crafted family models

RQ4) Effectives on learning precise family models by sampling vs. exhaustive?

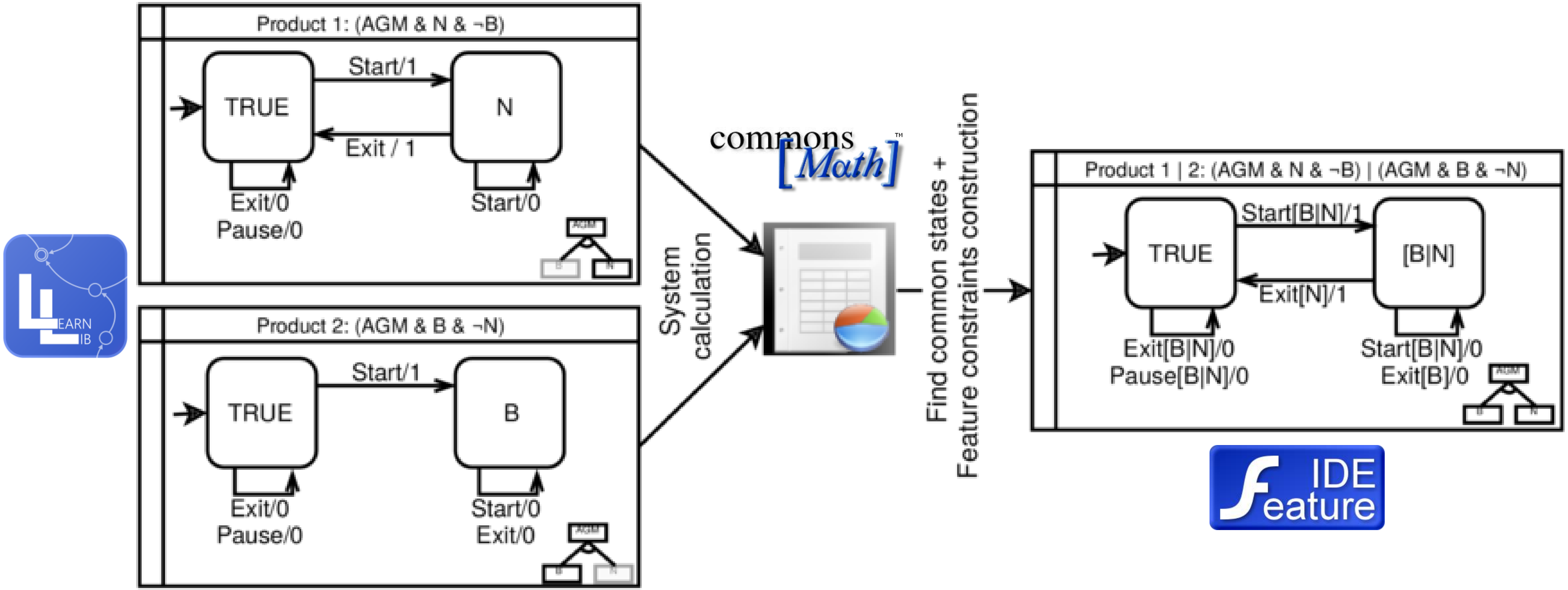
Subject Systems

SPL		Feature model		Family model	
ID	Name	Features	Valid conf.	States	Transitions
AGM	Arcade Game Maker	13	6	6	35
VM	Vending Machine	9	20	14	197
WS	Wiper System	8	8	13	112
AEROUC5	Aero UC5	7	9	25	450
CPTERMINAL	Card Payment	13	30	11	176
MINEPUMP	Minepump	9	32	25	575

Table 10 – Description of the SPLs under learning - Feature and family models

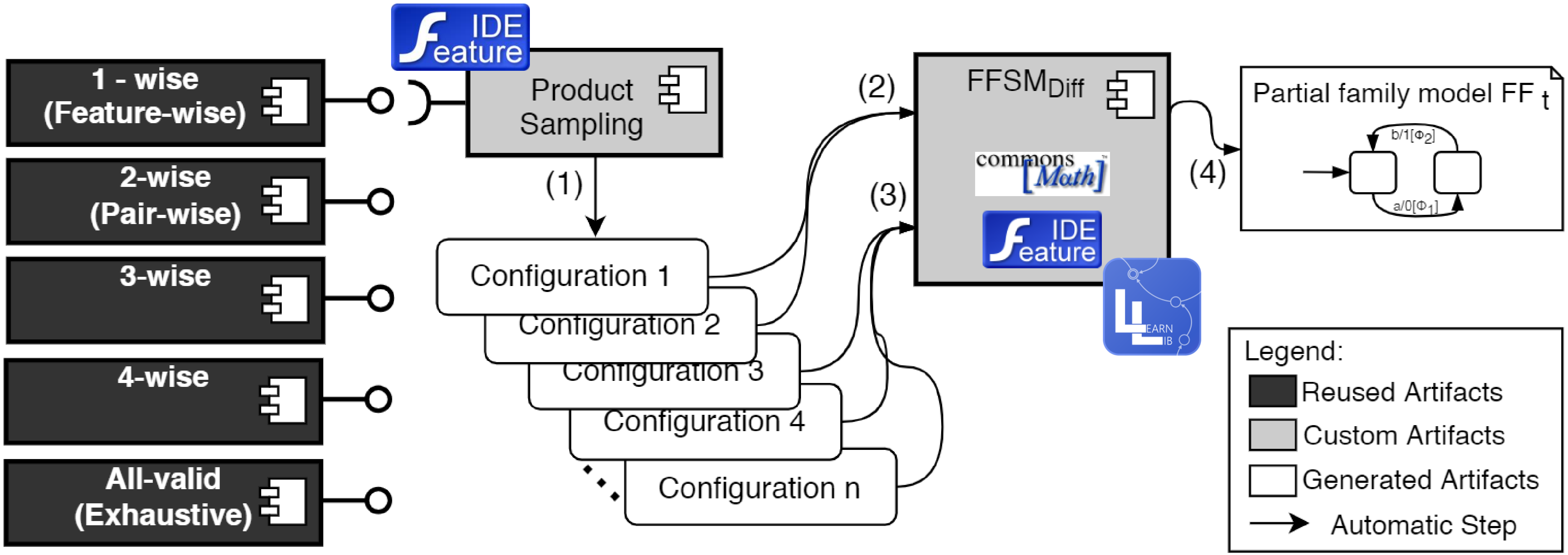
A. Classen, 'Modelling with FTS: a Collection of Illustrative Examples'. 2010, [Online]. Available: https://researchportal.unamur.be/en/publications/modelling-with-fts-a-collection-of-illustrative-examples
H. Samih, H. L. Guen, R. Bogusch, M. Acher, and B. Baudry, 'Deriving Usage Model Variants for Model-Based Testing: An Industrial Case Study', in Proceedings of the ICECCS 2014
X. Devroey, G. Perrouin, A. Legay, P.-Y. Schobbens, and P. Heymans, 'Search-based Similarity-driven Behavioural SPL Testing', in Proceedings of the VaMoS 2016
V. Hafemann Fragal, A. Simao, and M. R. Mousavi, 'Validated Test Models for Software Product Lines: Featured Finite State Machines', in `FACS 2016.

Experiment Design



M. Isberner, F. Howar, and B. Steffen, 'The Open-Source LearnLib', in CAV 2015
Apache, Commons Math: The Apache Commons Mathematics Library. 2016.
T. Thüm, et al., 'FeatureIDE: An extensible framework for feature-oriented software development', Science of Computer Programming, vol. 79, 2014

Experiment Design (cont.)



M. Isberner, F. Howar, and B. Steffen, 'The Open-Source LearnLib', in CAV 2015
Apache, Commons Math: The Apache Commons Mathematics Library. 2016.
T. Thüm, et al., 'FeatureIDE: An extensible framework for feature-oriented software development', Science of Computer Programming, vol. 79, 2014

ANALYSIS OF RESULTS

Analysis of Results (RQ1 and RQ3 – Size of Product Pairs/Handcrafted)

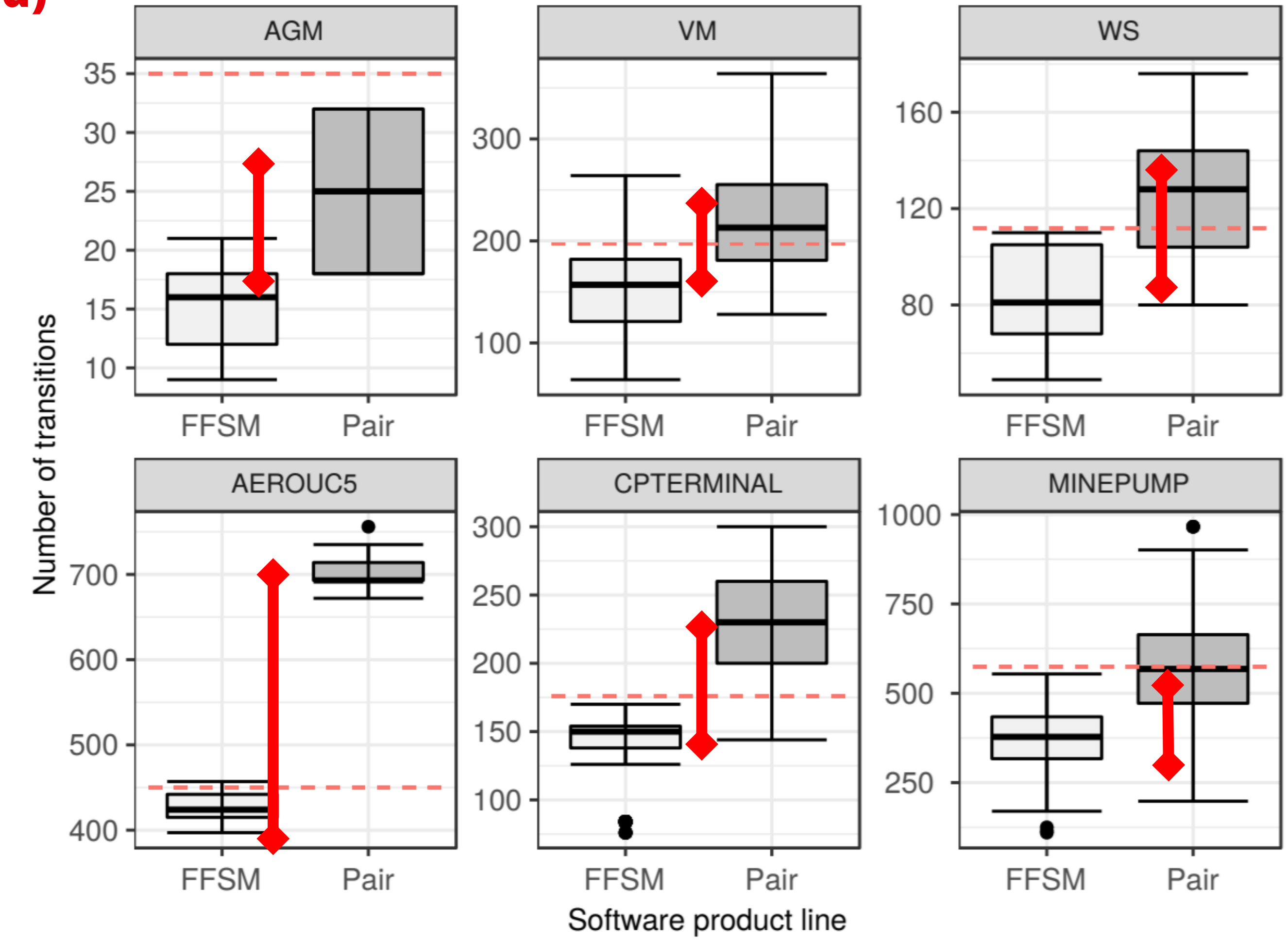


Figure 26 – Number of transitions in the learned FFSMs and pairs of products

Mann-Whitney-Wilcoxon statistical test
Vargha-Delaney's Â effect size

Analysis of Results (RQ2 – Configuration similarity)

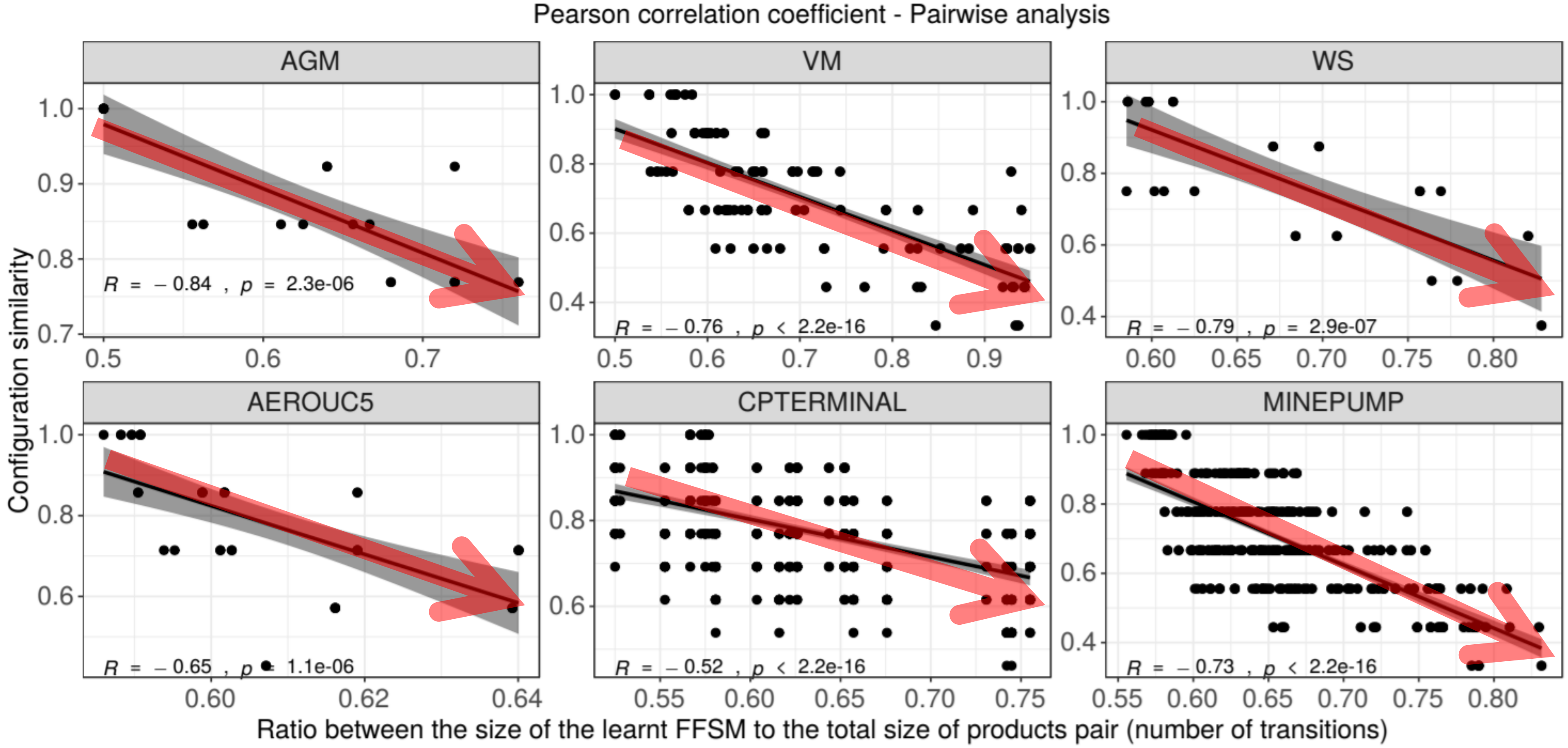


Figure 28 – Scatter plots for the relationship between the normalized size of the learned FFSM and configuration similarity

Pearson correlation coefficient

Analysis of Results (RQ4 – Learning by Sampling)

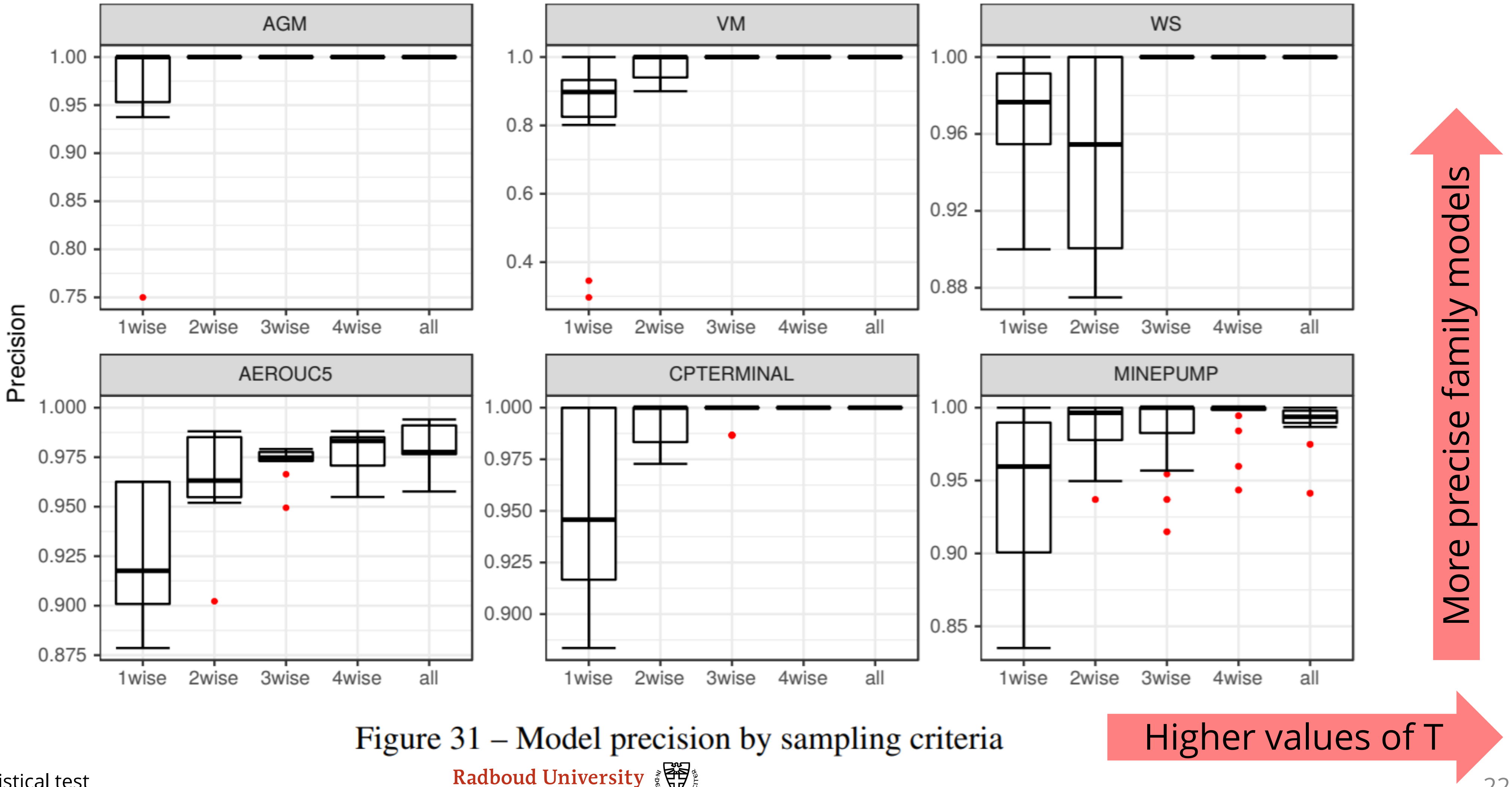
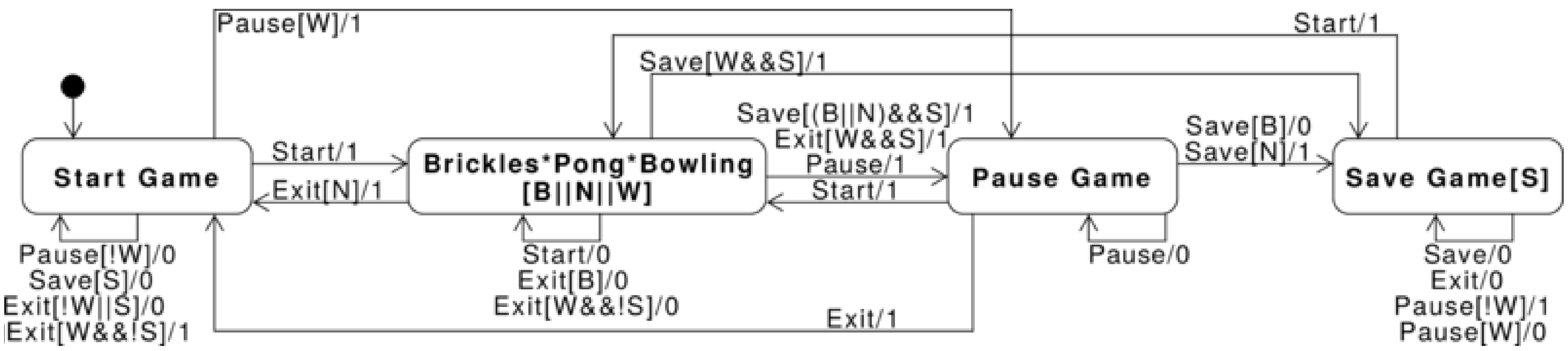


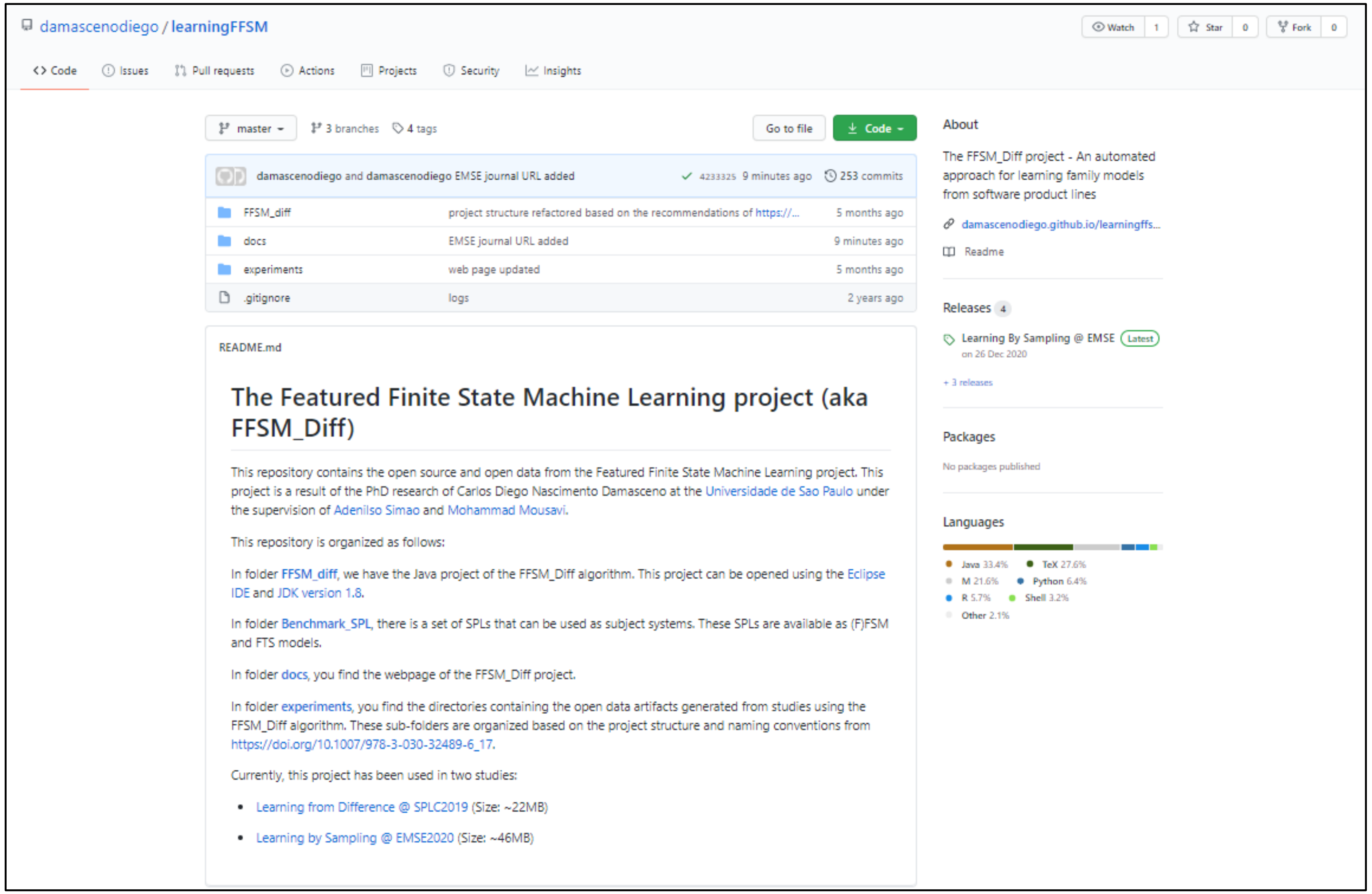
Figure 31 – Model precision by sampling criteria

Mann-Whitney-Wilcoxon statistical test
Vargha-Delaney's Â effect size

Analysis of Results (RQ4 – Learning by Sampling)



Analysis of Results (Software artifacts)


damascenodiego / learningFFSM

Watch 1 | Star 0 | Fork 0

Code | Issues | Pull requests | Actions | Projects | Security | Insights

master | 3 branches | 4 tags | Go to file | Code

damascenodiego and damascenodiego EMSE journal URL added ✓ 4233325 9 minutes ago 253 commits

FFSM_diff	project structure refactored based on the recommendations of https://...	5 months ago
docs	EMSE journal URL added	9 minutes ago
experiments	web page updated	5 months ago
.gitignore	logs	2 years ago

README.md

The Featured Finite State Machine Learning project (aka FFSM_Diff)

This repository contains the open source and open data from the Featured Finite State Machine Learning project. This project is a result of the PhD research of Carlos Diego Nascimento Damasceno at the Universidade de Sao Paulo under the supervision of Adenilso Simao and Mohammad Mousavi.

This repository is organized as follows:

In folder FFSM_diff, we have the Java project of the FFSM_Diff algorithm. This project can be opened using the Eclipse IDE and JDK version 1.8.

In folder Benchmark_SPL, there is a set of SPLs that can be used as subject systems. These SPLs are available as (F)FSM and FTS models.

In folder docs, you find the webpage of the FFSM_Diff project.

In folder experiments, you find the directories containing the open data artifacts generated from studies using the FFSM_Diff algorithm. These sub-folders are organized based on the project structure and naming conventions from https://doi.org/10.1007/978-3-030-32489-6_17.

Currently, this project has been used in two studies:

- Learning from Difference @ SPLC2019 (Size: ~22MB)
- Learning by Sampling @ EMSE2020 (Size: ~46MB)

About

The FFSM_Diff project - An automated approach for learning family models from software product lines

damascenodiego.github.io/learningffs...

Readme

Releases 4

Learning By Sampling @ EMSE Latest on 26 Dec 2020

+ 3 releases

Packages

No packages published

Languages

Java 33.4% | TeX 27.6% | M 21.6% | Python 6.4% | R 5.7% | Shell 3.2% | Other 2.1%

https://github.com/damascenodiego/learningFFSM

Artifacts Evaluated acm Functional

SPLC Paris 2019 23rd Systems and Software Product Line Conference

FINAL REMARKS

Summary

1. Learn fresh FFSMs from products pairs
   – Especially if there is high feature reuse (i.e., configuration similarity)
2. Incorporate new product behaviour into an existing FFSM
   – Family model recovery (e.g., reverse engineering, re-engineering)
3. Sampling lead to models as precise as those from exhaustive learning
   – Higher “T” values lead to higher coverage
   – Sampling can be helpful to family model learning

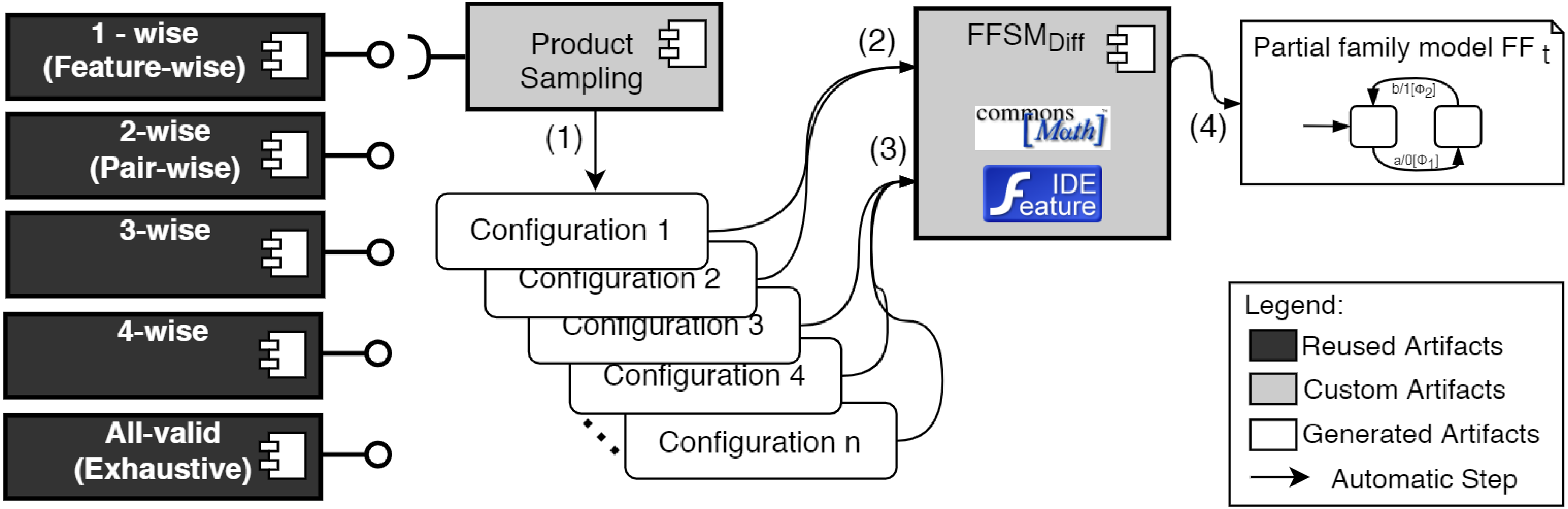
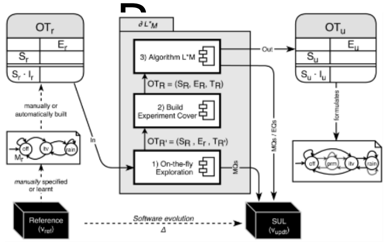


Fig. 8: Experiment design - Learning FFSMs by product sampling

Future Work

Learning to

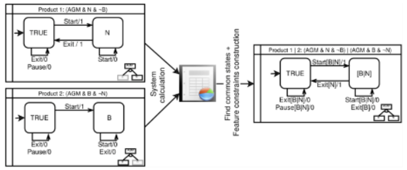


Active family model learning

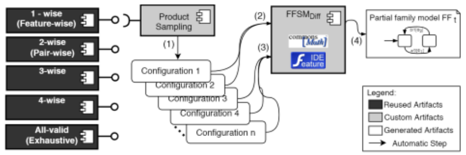
Fingerprinting evolving systems

Incremental Configurable Queries

Learning from Differences



Learning by Sampling



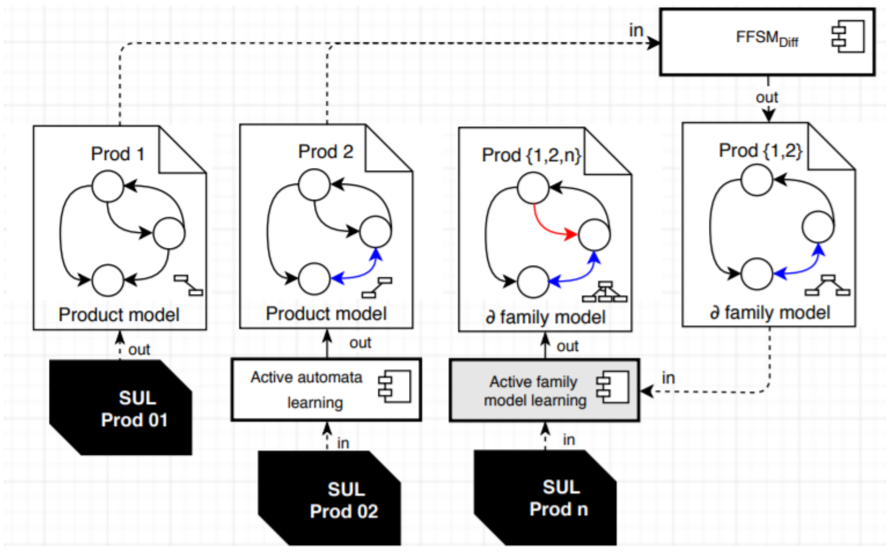


Figure 32 – Active family model learning

Source: Damasceno (2019).

C. D. N. Damasceno, 'Learning From Families: Inferring Behavioral Variability From Software Product Lines', presented at the PhD Symposium at Integrated Formal Methods, Bergen, Norway, 2019.
M. Al-Hajjaji, S. Krieter, T. Thüm, M. Lochau, and G. Saake, 'IncLing: Efficient Product-Line Testing Using Incremental Pairwise Sampling', in Proceedings of the GPCE 2016
G. Shu and D. Lee, 'Network Protocol System Fingerprinting - A Formal Approach', in Proceedings of the IEEE INFOCOM 2006

THANK YOU

Radboud University