Inference of Family Models for Software Product Line Testing

Carlos Diego Nascimento Damasceno, Adenilso Simão
Instituto de Ciências Matemáticas e de Computação (ICMC-USP)
damascenodiego@usp.br, adenilso@icmc.usp.br

1º Encontro Paulista dos Pós-graduandos em Computação

Abstract

Background: MBT relies on explicit models as central artifacts. Family models allow for specifying and testing properties of SPLs that shall emerge from individual and combined features. Model inference aims at extracting test models of hardware/software systems.

Problem statement: Building and maintaing models is onerous. There are no inference techniques to extract family models.

Aims: Investigate model inference to extract in an efficient and effective way family models from SPLs.

Model-Based Testing (MBT)

"Software testing is model-based" [2]

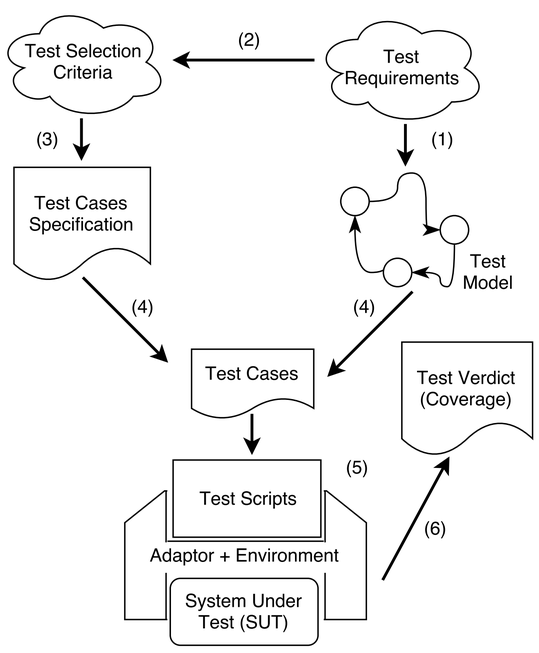

Fig. 1: The generic process of MBT [12]

(1) Specify an explicit test model of the SUT
(2) Define test selection criteria
(3) Abstract test generation
(4) Test concretization
(5) Test execution
(6) Coverage analysis

Challenges:

- Missing/incomplete models impose hurdles
- Build and maintain models is onerous
- Dependent on engineers expertise

Model Inference

Active automata learning [1]

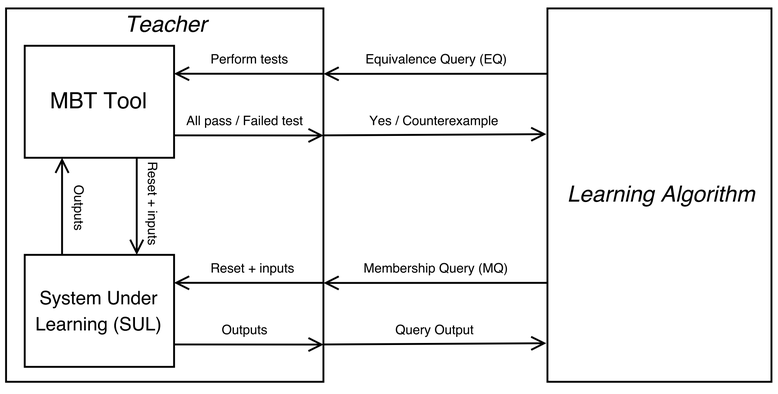

Fig. 2: Minimally Adequate Teacher (MAT) [1]

- Tests as queries
- Hypothesis construction (MQ)
- Hypothesis validation (EQ)
- Tool support: Learnlib [6], RAlib [10]
- Algorithms: L^* [1], L^*_M [11]
- Filters [7] and evolving systems [3]
- Learning-Based Testing (LBT) [8]
- Assertion violations [4]

Software Product Line Engineering (SPLE)

- Development of multiple products from a common set of shared assets [9]
- Feature model: Exponential number of valid products
- Family model-based testing: Featured Finite State Machine - FFSM [5]

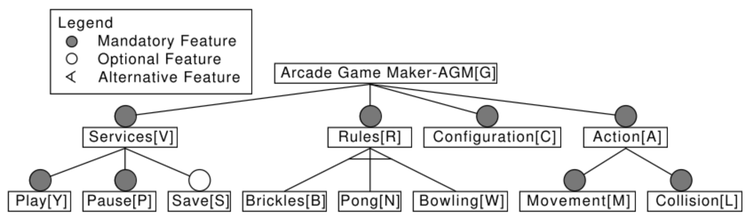

Fig. 3: Feature model

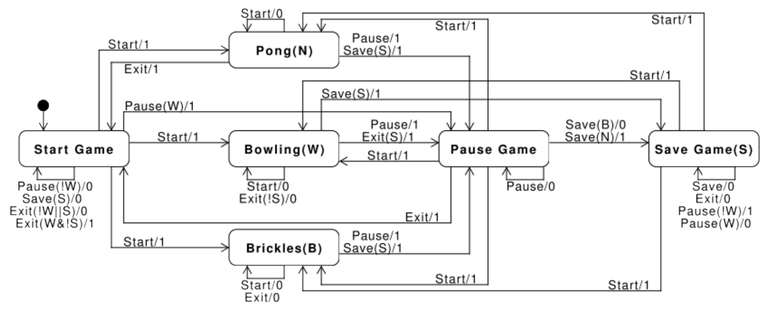

Fig. 4: FFSM model

Challenges:

Exhaustive analysis: Standard MBT is unsuitable and perform redundant computations
Feature interaction: Reused assets have to work as intended, regardless feature combinations

Research Proposal and Method

*How **model inference** can be lifted to the **family-based** level and enable an **efficient** and **effective** extraction of **family models of SPLs**?*

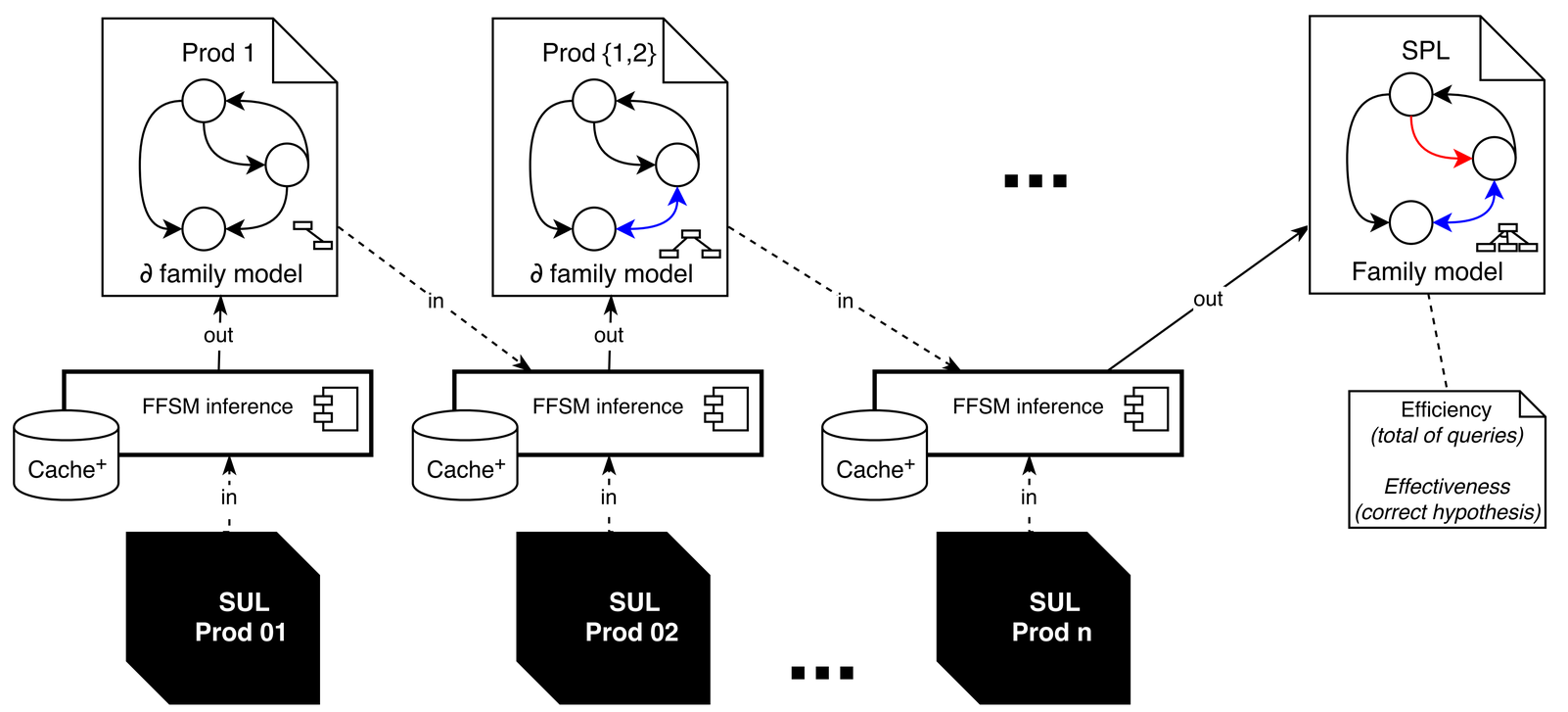

Fig. 5: Evaluation of family model inference

Research questions (RQ):

RQ1: How can we effectively infer Mealy machines from SPLs?
RQ2: How can we merge Mealy machines to generate FFSMs?
RQ3: How can we efficiently infer FFSMs from products of SPLs?
RQ4: How can we take advantage of LBT for testing SPLs?
RQ5: Can we use family model inference to detect feature interaction problems?
RQ6: How can we perform family model inference in a setting of Extended FSMs (EFSM)?

References

[1] Dana Angluin. Learning regular sets from queries and counterexamples. *Information and Computation*, 75(2), 1987.
[2] Robert V. Binder. *Testing Object-oriented Systems: Models, Patterns, and Tools*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1999.
[3] Sagar Chaki, Edmund Clarke, Natasha Sharygina, and Nishant Sinha. Verification of evolving software via component substitutability analysis. *Formal Methods in System Design*, 32(3):235–266, Jun 2008.
[4] Martin Chapman, Hana Chockler, Pascal Kesseli, Daniel Kroening, Ofer Strichman, and Michael Tautschnig. *Learning the Language of Error*, pages 114–130. Springer International Publishing, Cham, 2015.
[5] Vanderson Hafemann Fragal, Adenilso Simao, and Mohammad Reza Mousavi. *Validated Test Models for Software Product Lines: Featured Finite State Machines*, pages 210–227. Springer International Publishing, Cham, 2017.
[6] LearnLib. LearnLib: a framework for automata learning. `https://learnlib.de/`, 2017. [accessed 17-Out-2017].
[7] Tiziana Margaria, Harald Raffelt, and Bernhard Steffen. Knowledge-based relevance filtering for efficient system-level test-based model generation. *Innovations in Systems and Software Engineering*, 1(2):147–156, Sep 2005.
[8] K. Meinke and M. A. Sindhu. LBTest: A learning-based testing tool for reactive systems. In *2013 IEEE Sixth International Conference on Software Testing, Verification and Validation*, pages 447–454, March 2013.
[9] Klaus Pohl, Günter Böckle, and Frank J. van der Linden. *Software Product Line Engineering: Foundations, Principles and Techniques*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005.
[10] RALib. A LearnLib extension for inferring EFSMs. `bitbucket.org/learnlib/ralib/`, 2017. [accessed 17-Out-2017].
[11] Muzammil Shahbaz and Roland Groz. *Inferring Mealy Machines*, pages 207–222. Springer, Berlin, Heidelberg, 2009.
[12] Mark Utting, Alexander Pretschner, and Bruno Legeard. A taxonomy of model-based testing approaches. *Software Testing, Verification and Reliability*, 22(5):297–312, 2012.