Ameur-Boulifa, Rabéa; Henrio, Ludovic; Madelaine, Eric

Compositional equivalences based on open pNets. (English) Zbl 07649239

Summary: Establishing equivalences between programs is crucial both for verifying correctness of programs and for justifying optimisations and program transformations. There exist several equivalence relations for programs, and bisimulations are among the most versatile of these equivalences. Among bisimulations one distinguishes strong bisimulation that requires that each action of a program is simulated by a single action of the equivalent program, and weak bisimulation that allows some of the actions to be invisible, and thus not simulated.

pNet is a generalisation of automata that model open systems. They feature variables and hierarchical composition. Open pNets are pNets with holes, i.e. placeholders that can be filled later by sub-systems. However, there is no standard tool for defining the semantics of an open system in this context. This article first defines open automata that are labelled transition systems with parameters and holes. Relying on open automata, it then defines bisimilarity relations for the comparison of systems specified as pNets. We first present a strong bisimilarity for open pNets called FH-bisimilarity. Next we offer an equivalence relation similar to the classical weak bisimulation equivalence, and study its properties. Among these properties we are interested in compositionality: if two systems are proven equivalent they will be indistinguishable by their context, and they will also be indistinguishable when their holes are filled with equivalent systems. The contributions of this article are illustrated using a transport protocol as running example.

MSC:
68Q85 Models and methods for concurrent and distributed computing (process algebras, bisimulation, transition nets, etc.)
68N30 Mathematical aspects of software engineering (specification, verification, metrics, requirements, etc.)

Keywords: bisimulation; compositionality; automata; semantics

Software: LOTOS

Full Text: DOI arXiv

References:
[8] Bludz, Simon; Henrio, Ludovic; Madelaine, Eric, Verification of concurrent design patterns with data, (Riis Nielson, Hanne; Tuosto, Emilio, COORDINATION 2019 - 21st International Conference on Coordination Models and Languages, volume


Calder, Muffy; Shankland, Caron, A symbolic semantics and bisimulation for full LOTOS, (FORTE 2001 - 21st International Conference on Formal Techniques for Networked and Distributed Systems (2001), Springer), 185-200 - Zbl 0977.68068


Henrio, Ludovic; Kulankhina, Oleksandra; Li, Siqi; Madelaine, Eric, Integrated environment for verifying and running distributed components, (FASCE 2016 - 19th International Conference on Formal Fundamental Approaches to Software Engineering (2016), Springer: Springer Berlin Heidelberg), 66-83

Henrio, Ludovic; Madelaine, Eric; Zhang, Min, A theory for the composition of concurrent processes, (FORTE 2016 - 36th International Conference on Formal Techniques for Distributed Objects, Components, and Systems, vol. 9688 (2016), Springer), 175-194 - Zbl 1347.68267

Hou, Zechen; Madelaine, Eric, Symbolic bisimulation for open and parameterized systems, (PEPM 2020 - Workshop on Partial Evaluation and Program Manipulation (2020), ACM SIGPLAN)

Hou, Zechen; Madelaine, Eric; Liu, Jing; Deng, Yuxin, Symbolic Bisimulation for Open and Parameterized Systems - Extended version (November 2019), Inria & Université Cote d’Azur, CNRS, IIS/East China Normal University: Inria & Université Cote d’Azur, CNRS, IIS/East China Normal University Sophia Antipolis, France/Shanghai, Research Report RR-9304


Johnson, Kenneth; Calinescu, Radu, Efficient re-resolution of SMT specifications for evolving software architectures, (QoSA

Edited by FIZ Karlsruhe, the European Mathematical Society and the Heidelberg Academy of Sciences and Humanities © 2023 FIZ Karlsruhe GmbH
[34] Johnson, Kenneth; Calinescu, Radu; Kikuchi, Shinji, An incremental verification framework for component-based software systems, (CBSE 2013 - 16th International ACM Sigsoft Symposium on Component-Based Software Engineering (2013), ACM), 33-42


[37] Laveaux, Maurice; Willems, Tim A. C., Decomposing monolithic processes in a process algebra with multi-actions, (ICE, EPTCS, vol. 347 (2021)), 57-76

[38] Lin, Huimin, Symbolic transition graph with assignment, (CONCUR’96 - Concurrency Theory, 7th International Conference, vol. 1119 (1996), Springer: Springer Berlin Heidelberg), 50-65


[44] Qin, Xudong; Bliudze, Simon; Madelaine, Eric; Zhang, Min, Using SMT engine to generate Symbolic Automata -Extended version (June 2018), Inria ; & Université Côte d’Azur, CNRS, I3S; Inria ; & Université Côte d’Azur, CNRS, I3S Sophia Antipolis, France; Inria, Research Report RR-9177


[46] Wang, Biyang; Madelaine, Eric; Zhang, Min, New symbolic model and equivalences checking for open automata, (SMC 2021 - IEEE International Conference on Systems, Man, and Cybernetics (2021), IEEE), 2360-2367

[47] Wang, Biyang; Madelaine, Eric; Zhang, Min, Symbolic Weak Equivalences: Extension, Algorithms, and Minimization - Extended version (2021), Inria, Université Côte d’Azur, CNRS, I3S; East China Normal University; Inria, Université Côte d’Azur, CNRS, I3S; East China Normal University Sophia Antipolis, France/Shanghai, Research Report RR-9389

This reference list is based on information provided by the publisher or from digital mathematics libraries. Its items are heuristically matched to zbMATH identifiers and may contain data conversion errors. It attempts to reflect the references listed in the original paper as accurately as possible without claiming the completeness or perfect precision of the matching.