A Categorical Framework for Testing Generalised Tree Automata

Bálint Kocsis
Department of Software Science, Radboud University, Nijmegen, The Netherlands

Abstract

In recent work, we introduced a framework for proving completeness of test suites at the abstract level of automata in monoidal closed categories, and provided a generalisation of a classical conformance testing technique, the W-method. As a continuation of this line of work, we present a generalisation of this result for proving completeness of test suites for automata whose transition structure is described by an arbitrary endofunctor satisfying certain conditions.

2012 ACM Subject Classification Theory of computation → Algebraic language theory; Theory of computation → Automata extensions; Theory of computation → Tree languages

Keywords and phrases Algebra, Automata, Conformance testing, Monoidal categories

Funding This research is supported by the NWO grant No. VI.Vidi.223.096.

1 Learning and testing

Active automata learning [11, 12, 19] is the process of inferring an automaton model by interacting with a system (the *system under learning*, SUL). A popular setting for approaching this problem is Angluin's *minimally adequate teacher* (MAT) framework [1], in which the *learner* can pose two types of queries to the *teacher*. One of them is called the *equivalence query*, which is used to evaluate whether a learned model – the *hypothesis* – is correct.

In the case when the SUL is a black-box – a scenario that arises often in practice –, the equivalence query can only be implemented through *testing*. In this context, the known hypothesis is called the *specification*, the black-box SUL is called the *implementation*, and the problem of checking equivalence is called *conformance testing* [5].

Although testing can never be exhaustive [16], under certain assumptions on the implementation, it is possible to construct so called *complete* test suites that guarantee equivalence.

2 The W-method for automata in categories

In order to obtain good implementations of active automata learning algorithms, we need an efficient way to generate complete test suites from the specification. The *W-method* [6, 21] is a classical construction of such a test suite for deterministic finite automata [17]. The method asks for two sets of words, called a *state cover* and a *characterisation set*, satisfying certain properties. The possible concatenations of the words in these sets form the test suite.

Recently, in joint work with Rot [13], the author provided a categorical framework for proving completeness of test suites, along with a generalisation of the W-method. We considered deterministic automata in monoidal closed categories [10], whose transition structure is given by a map of type $Q \otimes \Sigma \to Q$, where Q is the state space and Σ is the input alphabet.

This framework is adequate for deriving complete test suites for various kinds of automata including weighted automata [4, 9] and deterministic nominal automata [3], as was done in [13]. However, it does not apply to automata with more sophisticated transition structures such as sorted automata [18] or tree automata [7].

3 Generalisation

Building on [13], we provide a more general framework for proving completeness of test suites that encompasses the automata classes mentioned above. We work in a monoidal closed category $\mathcal{C}$ with binary coproducts, equipped with an endofunctor $F\colon \mathcal{C} \to \mathcal{C}$. We assume that the forgetful functor $\mathrm{Alg}(F) \to \mathcal{C}$ has a left adjoint, giving rise to a *free F-algebra monad* (T, η, μ). (Such a functor F has been called an *input process* [2] and a *varietor* [20] in the literature.) Finally, we assume that T is a strong monad, and we fix an output object O.

These assumptions are satisfied, for instance, when F is a strong endofunctor that preserves filtered colimits. In particular, the latter condition holds of any left-adjoint functor and of polynomial endofunctors. Thus, the framework applies to all adjoint automata [18] and to tree automata [7]. To the best of the author's knowledge, this provides the first account of complete test suites for tree automata. It is left to future work to investigate these applications.

In the remainder of this section, we provide some technical details. Let us denote the tensor product of $\mathcal{C}$ by $- \otimes -$, its tensor unit by I, and its internal hom by $[-,-]$. We denote the F-algebra structure of TX by $\gamma_X\colon FTX \to TX$.

We think of TI as the object of inputs for automata, and of $[TI, O]$ as the object of behaviours. The object TI carries a monoid structure (η_I, m), where $\eta_I\colon I \to TI$ is the 'empty' input, and $m\colon TI \otimes TI \to TI$ is a generalised concatenation operation.

An *automaton* $\mathcal{A}$ is given by an F-algebra (Q, δ) together with an input morphism $i\colon I \to Q$ and an output morphism $f\colon Q \to O$. Using the freeness of TI, the morphism i can be extended to a morphism $r_{\mathcal{A}}\colon TI \to Q$, called the *reachability map* of $\mathcal{A}$ [2]. The *behaviour* of $\mathcal{A}$ [10] is then defined as the morphism $B_{\mathcal{A}} = TI \xrightarrow{r_{\mathcal{A}}} Q \xrightarrow{f} O$. The identity id_Q extends to a map $\delta^*\colon TQ \to Q$ called the *run map* of $\mathcal{A}$ [2]. Finally, we define the *observability map* $o_{\mathcal{A}}$ of $\mathcal{A}$ [2] as the adjunct of the composite $Q \otimes TI \to T(Q \otimes I) \xrightarrow{\cong} TQ \xrightarrow{\delta^*} Q \xrightarrow{f} O$.

State covers and characterisation sets are generalised as follows. Call a morphism $c\colon C \to TI$ a *weak state cover* for $\mathcal{A}$ if there exists a map $\delta_C\colon FC \to C$ such that $r_{\mathcal{A}} \circ c \circ \delta_C = r_{\mathcal{A}} \circ \gamma_I \circ Fc$, and call $w\colon W \to TI$ a *characterisation morphism* for $\mathcal{A}$ if $[w, \mathrm{id}] \circ o_{\mathcal{A}}$ is monic.

Before we state our main theorem, we introduce two operations on morphisms. For $a\colon A \to TI$ and $b\colon B \to TI$, let a^+ and $a \cdot b$ be defined as $[a, \gamma_I \circ Fa]$ and $m \circ (a \otimes b)$, respectively.

▶ **Theorem.** *Let $\mathcal{S}$ and $\mathcal{M}$ be two automata. Suppose $c\colon C \to TI$ is a weak state cover for $\mathcal{M}$ and $w\colon W \to TI$ is a characterisation morphism for $\mathcal{S}$ such that η_I factors through both c and w. Let $t = c^+ \cdot w$. Then $B_{\mathcal{S}} \circ t = B_{\mathcal{M}} \circ t$ implies $B_{\mathcal{S}} = B_{\mathcal{M}}$.*

Intuitively, the theorem states that in order to check the equivalence of two automata, it is sufficient to check that their behaviour agrees for all inputs in the test suite t. The test suite t is represented as a morphism into TI, and is thought of as a set of test inputs. The statement of the theorem is similar to [13, Theorem 4.9]. A difference is that we do not explicitly assume minimality of the specification since that is implied by the above definition of characterisation morphisms, which is stronger than the original one in [13].

4 Future work

Our general theorem is a starting point for the development of complete test suites with categorical tools. To put our framework to use, it would be beneficial to develop algorithms for computing state covers and characterisation sets. Another direction for future research is to generalise more refined test methods based on the W-method (see [8, 14, 15]).

References

1 Dana Angluin. Learning regular sets from queries and counterexamples. *Information and Computation*, 75(2):87–106, 1987. doi:10.1016/0890-5401(87)90052-6.

2 Michael A. Arbib and Ernest G. Manes. Adjoint machines, state-behavior machines, and duality. *J. Pure Appl. Algebra*, 6(3):313–344, 1975. doi:10.1016/0022-4049(75)90028-6.

3 Mikolaj Bojańczyk, Bartek Klin, and Slawomir Lasota. Automata theory in nominal sets. *Logical Methods in Computer Science*, 10(3), 2014. doi:10.2168/LMCS-10(3:4)2014.

4 Filippo Bonchi, Marcello M. Bonsangue, Michele Boreale, Jan J. M. M. Rutten, and Alexandra Silva. A coalgebraic perspective on linear weighted automata. *Information and Computation*, 211:77–105, 2012. doi:10.1016/J.IC.2011.12.002.

5 Manfred Broy, Bengt Jonsson, Joost-Pieter Katoen, Martin Leucker, and Alexander Pretschner, editors. *Model-Based Testing of Reactive Systems*, volume 3472 of *Lecture Notes in Computer Science*. Springer, 2005. doi:10.1007/B137241.

6 Tsun S. Chow. Testing software design modeled by finite-state machines. *IEEE Transactions on Software Engineering*, 4(3):178–187, 1978. doi:10.1109/TSE.1978.231496.

7 Hubert Comon, Max Dauchet, Rémi Gilleron, Florent Jacquemard, Denis Lugiez, Christof Löding, Sophie Tison, and Marc Tommasi. *Tree Automata Techniques and Applications*. HAL, 2008.

8 Rita Dorofeeva, Khaled El-Fakih, Stéphane Maag, Ana R. Cavalli, and Nina Yevtushenko. FSM-based conformance testing methods: A survey annotated with experimental evaluation. *Inf. Softw. Technol.*, 52(12):1286–1297, 2010. doi:10.1016/J.INFSOF.2010.07.001.

9 Manfred Droste and Dietrich Kuske. Weighted automata. In *Handbook of Automata Theory (I.)*, pages 113–150. European Mathematical Society Publishing House, Zürich, Switzerland, 2021. doi:10.4171/AUTOMATA-1/4.

10 Joseph A. Goguen. Discrete-time machines in closed monoidal categories. I. *Journal of Computer and System Sciences*, 10(1):1–43, 1975. doi:10.1016/S0022-0000(75)80012-2.

11 Falk Howar and Bernhard Steffen. Active automata learning in practice - an annotated bibliography of the years 2011 to 2016. In *Machine Learning for Dynamic Software Analysis*, volume 11026 of *Lecture Notes in Computer Science*, pages 123–148. Springer, 2018. doi:10.1007/978-3-319-96562-8_5.

12 Malte Isberner. *Foundations of active automata learning: an algorithmic perspective*. PhD thesis, Technical University Dortmund, Germany, 2015.

13 Bálint Kocsis and Jurriaan Rot. Complete test suites for automata in monoidal closed categories. In *FoSSaCS*, volume 15691 of *Lecture Notes in Computer Science*, pages 198–219. Springer, 2025. doi:10.1007/978-3-031-90897-2_10.

14 David Lee and Mihalis Yannakakis. Principles and methods of testing finite state machines-a survey. *Proceedings of the IEEE*, 84(8):1090–1123, 1996. doi:10.1109/5.533956.

15 Joshua S. Moerman. *Nominal techniques and black box testing for automata learning*. PhD thesis, Radboud University Nijmegen, The Netherlands, 2019.

16 Edward F. Moore. *Gedanken-Experiments on Sequential Machines*, pages 129–154. Princeton University Press, 1956. doi:doi:10.1515/9781400882618-006.

17 Michael O. Rabin and Dana S. Scott. Finite automata and their decision problems. *IBM Journal of Research and Development*, 3(2):114–125, 1959. doi:10.1147/RD.32.0114.

18 Henning Urbat and Lutz Schröder. Automata learning: An algebraic approach. In *LICS*, pages 900–914. ACM, 2020. doi:10.1145/3373718.3394775.

19 Frits W. Vaandrager. Model learning. *Communications of the ACM*, 60(2):86–95, 2017. doi:10.1145/2967606.

20 Gerco van Heerdt, Tobias Kappé, Jurriaan Rot, Matteo Sammartino, and Alexandra Silva. A categorical framework for learning generalised tree automata. In *CMCS*, volume 13225 of *Lect. Notes Comput. Sci.*, pages 67–87. Springer, 2022. doi:10.1007/978-3-031-10736-8_4.

21 M. P. Vasilevskii. Failure diagnosis of automata. *Cybernetics*, 9(4):653–665, 1973. doi:10.1007/BF01068590.