Title	Extensions and Applications of Antichain Algorithms
Author(s)	白木澤，啓
Citation	
Issue Date	2017-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/14147
Rights	
Description	Supervisor:小川　瑞史，情報科学研究科，修士

Extensions and Applications of Antichain Algorithms

Kei Shirakizawa

March, 2017

This thesis is the contribution to the study of automata-theoretic theorem proving. Automata-theoretic theorem proving checks the satisfiability / validity of a first-order formula under a fixed interpretation. Elements in the universe are encoded into words and every predicate has a corresponding automaton which accepts the support of the formula.

There are several existing tools for automata-theoretic theorem proving. MONA [6] translates formulas in weak monadic second-order logic of k successor (WSkS) to finite tree automata. Recently FORT [15] is implemented for the first-order theory of term rewriting. It is based on tree automata and ground tree transducers (GTT) for left-liner right-ground term rewriting systems. They determine not only whether a formula is valid, but also generate counter-examples from the automata if the formula is not valid.

Difficulty of the automata construction comes from the state explosion problem. WS1S requires tower of computation task corresponding to the height of quantifiers in the input formula. The determinization of automata causes state explosion, which is necessary to translate the complementation.

Antichain Algorithms The commonly used optimization to tackle the state explosion is the on-the-fly state space generation [10]. Antichain algorithm, an additional technique originally developed in the model checking, combines the on-the-fly determinization and minimization [19]. Abdulla, et al. [1] combined antichains and a simulation technique and further reduced the state space of the universality/inclusion checking. An interesting empirical observation of FORT is that the flattening of a formula into a prenex normal form triggers further state explosion, which motivated our work. This paper investigates a generalized antichain algorithms without flattening. We focus on monadic first-order logic, which has neither set variables (as MONA) nor transitive closure (as FORT), as the most simple case study.

One of our aim was to directly handle a nested formula with an antichain algorithm (i.e., to avoid flattening the input formula). The aim was achieved by introducing the composition terms that represent the automata construction and by finding ordering inductively. So that the generalized antichain algorithm is inductively defined for the structure of composition terms. As an optimization, we further introduced

1. the conversion rules of composition terms which preserve the accepted language, and

2. the distributive laws of emptiness checking into the composition terms.

We performed experiments on randomly generated 3000 Presburger formulas. We could not observe that the generalized antichain algorithm improved the performance. Due to the overhead of calculating orderings, it did not work for small problems. In the most cases, conversion of the composition term performed better than the generalized antichain algorithm. Still, there were some cases where the generalized antichain algorithm outperformed other algorithms, when the problem was sufficiently large.

References

[1] P. A. Abdulla, Y. Chen, L. Holík, R. Mayr, and T. Vojnar. When simulation meets antichains. In *Proc. TACAS*, vol. 6015 *Lecture Notes in Computer Science*, pages 158–174. Springer, 2010.

[2] F. Baader and T. Nipkow. *Term Rewriting and All That*. Cambridge University Press, 1998.

[3] A. Bouajjani, P. Habermehl, L. Holík, T. Touili, and T. Vojnar. Antichain-based universality and inclusion testing over nondeterministic finite tree automata. In *Proc. CIAA*, volume 5148 of *Lecture Notes in Computer Science*, pages 57–67. Springer, 2008.

[4] A. Boudet and H. Comon. Diophantine equations, presburger arithmetic and finite automata. In *Proc. CAAP'96*, volume 1059 of *Lecture Notes in Computer Science*, pages 30–43. Springer, 1996.

[5] H. Comon, M. Dauchet, R. Gilleron, C. Löding, F. Jacquemard, D. Lugiez, S. Tison, and M. Tommasi. Tree automata techniques and applications. Available on: `http://www.grappa.univ-lille3.fr/tata`, 2007. release October, 12th 2007.

[6] J. Elgaard, N. Klarlund, and A. Møller. MONA 1.x: New techniques for WS1S and WS2S. In *Proc. CAV* , volume 1427 of *Lecture Notes in Computer Science*, pages 516–520. Springer, 1998.

[7] T. Fiedor, L. Holík, P. Janku, O. Lengál, and T. Vojnar. Lazy automata techniques for WS1S. *CoRR*, abs/1701.06282, 2017.

[8] T. Fiedor, L. Holík, O. Lengál, and T. Vojnar. Nested antichains for WS1S. In *Proc. TACAS* , volume 9035 of *Lecture Notes in Computer Science*, pages 658–674. Springer, 2015.

[9] D. Harel, J. Tiuryn, and D. Kozen. *Dynamic Logic*. MIT Press, Cambridge, MA, USA, 2000.

[10] G. Holzmann. On-the-fly model checking. *ACM Comput. Surv.*, 28(4es), Dec. 1996.

[11] L.Doyen and J.-F.Raskin. Improved algorithms for the automata-based approach to model-checking. In *Proc. TACAS*, volume 4424 of *Lecture Notes in Computer Science*, pages 451–465. Springer-Verlag, 2007.

[12] V. B. Marc and D. O. Gauwin. Visibly pushdown automata: Universality and inclusion via antichains. In *LATA*, volume 7810, pages 190–201. Springer, 2013.

[13] M.D.Wulf, L.Doyen, T.A.Henzinger, and J.-F.Raskin. Antichains: A new algorithm for checking universality of finite automata. In *Proc. CAV 2006*, volume 4144 of *Lecture Notes in Computer Science*, pages 17–30. Springer-Verlag, 2006.

[14] R.Alur and P.Madhusudan. Visibly pushdown languages. In *Proc. the 36th Annual ACM Symposium on Theory of Computing (STOC 2004)*, pages 202–211, 2004.

[15] F. Rapp and A. Middeldorp. Automating the first-order theory of rewriting for left-linear right-ground rewrite systems. In *FSCD* , pages 36:1–36:12, 2016.

[16] N. V. Tang. *Pushdown Automata and Inclusion Problems*. PhD thesis, JAIST, 3 2009.

[17] P. Techaveerapong. *Antichain algorithm, its theory and appilcations*. Master thesis, JAIST, 3 2009.

[18] W. Thomas. Handbook of formal languages, vol. 3. chapter Languages, Automata, and Logic, pages 389–455. Springer-Verlag , 1997.

[19] M. D. Wulf, L. Doyen, T. A. Henzinger, and J. Raskin. Antichains: A new algorithm for checking universality of finite automata. In *Proc. CAV* , volume 4144 of *Lecture Notes in Computer Science*, pages 17–30. Springer, 2006.

[20] M. D. Wulf, L. Doyen, N. Maquet, and J. Raskin. Alaska. In *Proc. ATVA*, volume 5311 of *Lecture Notes in Computer Science*, pages 240–245. Springer, 2008.