

Minimalne automaty Büchi

Andrzej Jasiński

Instytutu Matematyki i Informatyki Uniwersytetu Opolskiego

Metoda „*Model Checking*” służy do weryfikacji systemów, które mogą być modelowane za pomocą struktury Kripke’go. W metodzie tej (patrz [10]) używamy formuł Liniowej Logiki Temporalnej (LTL) oraz automatów Büchi. Zazwyczaj negacja formuły φ , która opisuje weryfikowaną własność systemu, jest przekształcana na równoważny automat Büchi $\mathbb{A}_{\neg\varphi}$, a następnie rozważany jest produkt $\mathbb{S} \times \mathbb{A}_{\neg\varphi}$, gdzie \mathbb{S} jest automatem, który modeluje system [7]. Złożoność tej metody w dużym stopniu zależy od rozmiaru automatu $\mathbb{A}_{\neg\varphi}$. Poszukując efektywnego translatora formuł LTL na automaty Büchi, rozważano różne typy automatów Büchi (Generalized Büchi Automata, Very Weak Alternating Automata, Testing Automata i inne), a także zaproponowano wiele interesujących algorytmów przekształcających (see [4], [3], [9],[6]). Głównym celem tych algorytmów jest wygenerowanie równoważnej formule automatu o jak najmniejszej liczbie stanów w rozsądnym czasie.

W pracy [11] wyznaczyliśmy automaty Büchi o minimalnej liczbie stanów, które kodują pewne relatywnie proste klasy formuł LTL. Formuły te, rozważane w literaturze (patrz [1], [2], [5], [8]), są dosyć często używane w procesie weryfikacji systemów. Zauważyliśmy, że nawet dla bardzo prostych formuł odpowiadające im automaty Büchi mają wykładniczą liczbę stanów. Na przykład, minimalny automat Büchi kodujący formułę $\Diamond p_1 \wedge \dots \wedge \Diamond p_n$ ma 2^n stanów zaś minimalny automat kodujący formułę $\Diamond(p_1 \wedge \Diamond q_1) \wedge \dots \wedge \Diamond(p_n \wedge \Diamond q_n)$ posiada 3^n stanów. Nasze wyniki porównaliśmy z automatami generowanymi przez dwa translatory LTL2BA oraz SPOT formuł języka LTL na automaty Büchi.

Literatura

- [1] J.R. Büchi, *On a decision method in restricted second-order arithmetic*, Proceedings of International Congress on Logic Method and Philosophy of Science, 1960, pp.1–12, Stanford University Press, 1962.
- [2] K. Etessami and G. J. Holzmann, *Optimizing Büchi automata*, Proceedings of the 11th International Conference of Concurrency Theory, pp. 154–167, Springer-Verlag, 2000.
- [3] P. Gastin, D.Oddoux, *Fast LTL to Büchi Automata Translation*, Proceedings of the 13th International Conference on Computer Aided Verification, pp. 53–65, Springer-Verlag, 2001.
- [4] R. Gerth, D. Peled, M. Y. Vardi, and P. Wolper, *Simple on-the-fly automatic verification of linear temporal logic*, Proceedings of the 15th IFIP Symp. of Protocol Specification, Testing, and Verification, pp. 3–18, North-Holland, 1995.
- [5] J. Geldenhuys and H. Hansen, *Larger automata and less work for LTL model checking*, In Model Checking Software, 13th Int’l SPIN Workshop, volume 3925 of LNCS, pp. 53–70, Springer-Verlag, 2006.
- [6] J. Geldenhuys, A. Valmari, *Tarjan’s algorithm makes on-the-fly LTL verification more efficient*, Proceedings of 10th Internat. Conf. on Tools and Algorithms for the Construction and Analysis of Systems (TACAS’04), volume 2988 of LNCS, pp. 205–219, Springer-Verlag, 2004. .
- [7] G. J. Holtzmann, *The Spin Model Checker*, Addison-Wesley, 2003.
- [8] K.Y. Rozier, M.Y. Vardi, *LTL Satisfiability Checking*, accepted for SPIN’07.
- [9] S. Schwon, J. Esperza, *A note on on-the-fly verification algorithms*, Proceedings of 11th Internat. Conf. on Tools and Algorithms for the Construction and Analysis of Systems (TACAS’05), volume 3440 of LNCS, pp. 174–190, Springer-Verlag, 2005.
- [10] P. Wolper, M.Y. Vardi, A.P. Sistla, *Reasoning about Infinite Computation Paths*, Proceedings of the 24th Symposium on Foundations of Computer Science, pp. 185–194, 1983.
- [11] J. Cichoń, A. Czubak, A. Jasiński, *Minimal Büchi Automata for Certain Classes of LTL Formulas*, to be published