THEORIE-TAG 2023

A. Lin, G. Zetzsche (Eds.): Theorietag 2023, Kaiserslautern, 4. – 6.10.2023
RPTU & MPI-SWS Kaiserslautern, Technical Report, pp. 8–12.

RPTU

On the Satisfiability of Context-free String Constraints with Subword-Ordering

C. Aiswarya[(A)] Soumodev Mal[(B)] Prakash Saivasan[(C)]

[(A)]Chennai Mathematical Institute and CNRS IRL ReLaX, India
aiswarya@cmi.ac.in

[(B)]Chennai Mathematical Institute, India
soumodevmal@cmi.ac.in

[(C)]The institute of Mathematical Sciences, HBNI and CNRS IRL ReLaX, India
prakashs@imsc.res.in

Short abstract We consider a variant of string constraints given by membership constraints in context-free languages and subword relation between variables. The satisfiability problem for this variant turns out to be undecidable. We consider a fragment in which the subword-order constraints do not impose any cyclic dependency between variables. We show that this fragment is nexptime-complete. As an application of our result, we settle the complexity of control state reachability in acyclic lossy channel pushdown systems, which was shown to be decidable in Atig-Bouajjani-Touilli-08. We show that this problem is nexptime-complete.

This work is published in the proceedings of LICS 2022 [6].

Long abstract The study of string constraints has been at the center of research for many decades now, foremost being the seminal work of Makanin [19]. The problem is of particular interest due its close connections to the Hilbert's tenth problem [20]. There have been several studies of string constraints (as word equations). While the decidability status of the general word equation when it includes length constraints is still open [13], the satisfiability of word equations without the length constraints was shown to be decidable by [19]. Subsequently there have been several attempts to simplify the proof and pin down the precise complexity of the problem [22, 21].

In the recent times, the topic has gained much momentum due to its applicability to identifying security vulnerabilities in programs [26, 23, 4, 25, 24]. The fact that almost every program manipulates strings, especially in very diverse ways does not allow for a uniform way to analyse these programs for security vulnerabilities arising out of string processing. There have been many works in this direction, each accounting for different sets of string manipulations and comparisons. In fact there are several string constraint solvers that have been successfully build and employed to this effect [17, 8, 1, 16, 15]. Our work [6] is also an effort in this direction.

One important aspect of checking security vulnerability is to verify if the input to a program is safe. For example the SQL injection attack masquerades program code as an SQL query.

This operation usually involves relating two strings that arise out of programs. To this effect, an interesting class of string constraints is given by 1) membership constraints – which confines the domain of each variable to a class of language and 2) relational constraints – which allows for comparisons between the variables. The problem of interest here is satisfiability which asks whether there is an assignment to the variables that satisfies the constraints.

This formalism has turned out to be quite useful for the modeling capabilities and has been well studied. While most of the work in literature, mostly confines the membership constraints to regular languages, several comparison operations have been considered. Some of them being ReplaceAll [11, 12], relations due to transducers [18, 14], transducer with length constraints [3].

While these kind of models enjoy a very high expressive power, they are often plagued by undecidability. An ongoing research has been to identify subclasses which recover decidability for the satisfiability problem. For example, the straight-line fragment in [11] imposes an acyclicity requirement among the relational constraints between the variables to obtain decidability.

In this work [6] we consider a class of string constraints, given by 1) membership constraints – which confines the domain of each variable to a *context-free language* and 2) relational constraints which relate variables by the subword-order. As far as our knowledge goes, this is the first attempt to include a context-free language in the membership constraint. We believe this is useful and interesting since vulnerable inputs to programs include strings that are generated by programs or are programs themselves [10]. In both of these cases, the generated string can be a context-free language.

While it is possible to recover equality checking through subword relation, we show that the satisfiability problem is undecidable in the presence of cyclic dependencies between variables. We then consider a subclass called the acyclic fragment, inspired by [11, 3]. We show that satisfiability checking under this assumption is nexptime complete. In fact we show that the problem is already nexptime hard when the membership constraints are given as regular languages. Towards the nexptime algorithm, we show that our model enjoys a small model property. That is if the given constraints is satisfiable then it is satisfiable by an assignment of a bounded size. While this technique is not new, the presence of context-free membership constraints makes the problem harder. Towards this, we derive new insights about the combinatorial structure of the parse trees of a context-free grammar embedding a given word as a subword.

As an important application of our result, we derive a complexity upper bound for acyclic lossy channel systems (introduced in [7]). Lossy channel systems are an important model of distributed systems where finite-state processes communicate via point-to-point message transmission over an unbounded channel. When the channels are assumed to be reliable, the control state reachability is undecidable [9]. However, when the channels are assumed to be lossy, control state reachability becomes decidable[2]. Even with lossy channels, if the processes are assumed to be pushdown, the control state reachability problem is undecidable[7, 5]. If the communication topology is assumed to be acyclic, this was shown to be decidable in [7], but no elementary upper bound was known.

We establish strong connection between the acyclic fragment of the subword-ordering string constraints and acyclic lossy channel systems [7]. This also allows us to settle the complexity of control state reachability in acyclic lossy channel systems which has been open for more than a decade. We show that this problem is decidable in elementary time, in fact in nexptime, and supplement the result with a matching lower bound.

References

[1] A. P. ABDULLA, F. M. ATIG, Y.-F. CHEN, D. P. BUI, L. HOLÍK, A. REZINE, P. RUMMER, Trau : SMT solver for string constraints. In: *Proceedings of the 18th Conference on Formal Methods in Computer-Aided Design*. FMCAD Inc., 2019, 165–169.

[2] P. ABDULLA, B. JONSSON, Verifying programs with unreliable channels. In: *Proceedings of 8th Annual IEEE Symposium on Logic in Computer Science*. IEEE Computer Society, Los Alamitos, CA, USA, 1993.

[3] P. A. ABDULLA, M. F. ATIG, B. P. DIEP, L. HOLÍK, P. JANKU, Chain-Free String Constraints. In: Y. CHEN, C. CHENG, J. ESPARZA (eds.), *Automated Technology for Verification and Analysis - 17th International Symposium, ATVA 2019, Taipei, Taiwan, October 28-31, 2019, Proceedings*. Lecture Notes in Computer Science 11781, Springer, 2019.

[4] O. C. ABIKOYE, A. ABUBAKAR, A. H. DOKORO, O. N. AKANDE, A. A. KAYODE, A novel technique to prevent SQL injection and cross-site scripting attacks using Knuth-Morris-Pratt string match algorithm. *EURASIP Journal on Information Security* **2020** (2020) 1.

[5] C. AISWARYA, On Network Topologies and the Decidability of Reachability Problem. In: C. GEORGIOU, R. MAJUMDAR (eds.), *Networked Systems - 8th International Conference, NETYS 2020, Marrakech, Morocco, June 3-5, 2020, Proceedings*. Lecture Notes in Computer Science 12129, Springer, 2020, 3–10.

[6] C. AISWARYA, S. MAL, P. SAIVASAN, On the Satisfiability of Context-free String Constraints with Subword-Ordering. In: C. BAIER, D. FISMAN (eds.), *LICS '22: 37th Annual ACM/IEEE Symposium on Logic in Computer Science, Haifa, Israel, August 2 - 5, 2022*. ACM, 2022, 6:1–6:13.
`https://doi.org/10.1145/3531130.3533329`

[7] M. F. ATIG, A. BOUAJJANI, T. TOUILI, On the Reachability Analysis of Acyclic Networks of Pushdown Systems. In: F. VAN BREUGEL, M. CHECHIK (eds.), *Concurrency Theory, 19th International Conference, CONCUR 2008, Toronto, Canada, August 19-22, 2008. Proceedings*. Lecture Notes in Computer Science 5201, Springer, 2008.

[8] M. BERZISH, V. GANESH, Y. ZHENG, Z3str3: A String Solver with Theory-Aware Heuristics. In: *Proceedings of the 17th Conference on Formal Methods in Computer-Aided Design*. FMCAD '17, FMCAD Inc, Austin, Texas, 2017.

[9] D. BRAND, P. ZAFIROPULO, On Communicating Finite-State Machines. *J. ACM* **30** (1983) 2.

[10] C. CADAR, V. GANESH, P. M. PAWLOWSKI, D. L. DILL, D. R. ENGLER, EXE: Automatically Generating Inputs of Death. *ACM Trans. Inf. Syst. Secur.* **12** (2008) 2.

[11] T. CHEN, Y. CHEN, M. HAGUE, A. W. LIN, Z. WU, What is decidable about string constraints with the ReplaceAll function. *Proc. ACM Program. Lang.* **2** (2018) POPL, 3:1–3:29.

[12] T. CHEN, M. HAGUE, A. W. LIN, P. RÜMMER, Z. WU, Decision procedures for path feasibility of string-manipulating programs with complex operations. *Proc. ACM Program. Lang.* **3** (2019) POPL.

[13] V. GANESH, M. MINNES, A. SOLAR-LEZAMA, M. C. RINARD, Word Equations with Length Constraints: What's Decidable? In: A. BIERE, A. NAHIR, T. E. J. VOS (eds.), *Hardware and Software: Verification and Testing - 8th International Haifa Verification Conference, HVC 2012, Haifa, Israel, November 6-8, 2012. Revised Selected Papers*. Lecture Notes in Computer Science 7857, Springer, 2012.

[14] L. HOLÍK, P. JANKU, A. W. LIN, P. RÜMMER, T. VOJNAR, String constraints with concatenation and transducers solved efficiently. *Proc. ACM Program. Lang.* **2** (2018) POPL.

[15] S. KAN, A. W. LIN, P. RÜMMER, M. SCHRADER, CertiStr: a certified string solver. In: A. POPESCU, S. ZDANCEWIC (eds.), *CPP '22: 11th ACM SIGPLAN International Conference on Certified Programs and Proofs, Philadelphia, PA, USA, January 17 - 18, 2022*. ACM, 2022.

[16] A. KIEZUN, V. GANESH, S. ARTZI, P. J. GUO, P. HOOIMEIJER, M. D. ERNST, HAMPI: A Solver for Word Equations over Strings, Regular Expressions, and Context-Free Grammars. *ACM Trans. Softw. Eng. Methodol.* **21** (2013) 4.

[17] T. LIANG, A. REYNOLDS, N. TSISKARIDZE, C. TINELLI, C. BARRETT, M. DETERS, An Efficient SMT Solver for String Constraints. *Form. Methods Syst. Des.* **48** (2016) 3.

[18] A. W. LIN, P. BARCELÓ, String solving with word equations and transducers: towards a logic for analysing mutation XSS. In: R. BODÍK, R. MAJUMDAR (eds.), *Proceedings of the 43rd Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2016, St. Petersburg, FL, USA, January 20 - 22, 2016*. ACM, 2016, 123–136.

[19] G. S. MAKANIN, The problem of solvability of equations in a free smigroup. *Mathematics of the USSR-Sbornik* **32(2)** (1977).

[20] Y. V. MATIYASEVICH, A connection between systems of words-and-lengths equations and Hilbert's tenth problem. *Studies in constructive mathematics and mathematical logic. Part II,Zap. Nauchn. Sem. LOMI* **8** (1968).

[21] W. PLANDOWSKI, Satisfiability of Word Equations with Constants is in PSPACE. In: *40th Annual Symposium on Foundations of Computer Science, FOCS '99, 17-18 October, 1999, New York, NY, USA*. IEEE Computer Society, 1999, 495–500.

[22] W. PLANDOWSKI, An efficient algorithm for solving word equations. In: J. M. KLEINBERG (ed.), *Proceedings of the 38th Annual ACM Symposium on Theory of Computing, Seattle, WA, USA, May 21-23, 2006*. ACM, 2006.

[23] S. SON, K. S. MCKINLEY, V. SHMATIKOV, Diglossia: Detecting Code Injection Attacks with Precision and Efficiency. In: *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security*. CCS '13, Association for Computing Machinery, New York, NY, USA, 2013.

[24] M. TRINH, D. CHU, J. JAFFAR, S3: A Symbolic String Solver for Vulnerability Detection in Web Applications. In: G. AHN, M. YUNG, N. LI (eds.), *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, Scottsdale, AZ, USA, November 3-7, 2014*. ACM, 2014.

[25] M. TRINH, D. CHU, J. JAFFAR, Progressive Reasoning over Recursively-Defined Strings. In: S. CHAUDHURI, A. FARZAN (eds.), *Computer Aided Verification - 28th International Conference, CAV 2016, Toronto, ON, Canada, July 17-23, 2016, Proceedings, Part I*. Lecture Notes in Computer Science 9779, Springer, 2016.

[26] T.-Y. WU, J.-S. PAN, C.-M. CHEN, C.-W. LIN, Towards SQL Injection Attacks Detection Mechanism Using Parse Tree. In: H. SUN, C.-Y. YANG, C.-W. LIN, J.-S. PAN, V. SNASEL, A. ABRAHAM (eds.), *Genetic and Evolutionary Computing*. Springer International Publishing, 2015.