Parametrized Automata Simulation and Application to Service Composition

Walid Belkhir[*], and Yannick Chevalier[†], and Michael Rusinowitch [‡]

June 2014

Abstract

The service composition problem asks whether, given a client and a community of available services, there exists an agent (called the mediator) that suitably delegates the actions requested by the client to the available community of services. We address this problem in a general setting where the agents communication actions are parametrized by data from an infinite domain and possibly subject to constraints. For this purpose, we define *parametrized automata* (PAs), where transitions are guarded by conjunction of equalities and disequalities. We solve the service composition problem by showing that the simulation preorder of PAs is decidable and devising a procedure to synthesise a mediator out of a simulation preorder. We also show that the nonemptiness problem of PAs is PSPACE-complete. For details see [Belkhir et al.(2014)].

Motivations

Service Oriented Architectures (SOA) consider services as self-contained components that can be published, invoked over a network and combined with other services through standardized protocols in order to dynamically build complex applications [Alonso et al.(2004)]. Service composition is required when none of the existing services can fulfill some client needs but a suitable coordination of them would satisfy the client requests. How to find the right combination and how to orchestrate this combination are among the key issues for service architecture development.

Service composition has been studied in many works e.g. [Hull and Su(2005), Martín et al.(2012), Berardi et al.(2008)]. The related problem of system synthesis from libraries of reusable components has been thoroughly investigated too [Lustig and Vardi(2009)].

In this paper we address the composition synthesis problem for web services in which the agents are *parametrized*, i.e. the client and the available services exchange data ranging over an infinite domain and they are possibly subject to some *data constraints*. More precisely, the composition synthesis problem we consider can be stated as follows: (e.g. [Nourine and Toumani(2012), Cheikh(2009)]): given a client and a community of available services, compute a mediator which will enable communication between the client and the available services in such a way that each client request is forwarded to an appropriate service.

This problem was reduced to show that there exists a simulation relation between the target service (specifying an expected service behaviour for satisfying the client requests) and the asynchronous product of the available services. If such a simulation relation exists then it can

[*]LORIA
[†]IRIT
[‡]Corresponding author, LORIA

be easily used to generate a mediator, that is a function that selects at each step an available service for executing an action requested by the client.

One of the most successful approaches to composition amounts to abstract services as finite-state automata (FA) and apply available tools from automata theory to synthesize a new service satisfying the given client requests from an existing community of services. However it is not obvious whether the automata-based approach to service composition can still be applied with infinite alphabets since simulation often gets undecidable in extended models like Colombo ([Akroun et al.(2013)]). Starting from the approach initiated in [Belkhir et al.(2013)] our objective is to define expressive classes of automata on infinite alphabets which are well-adapted to the specification and composition of services and enjoy nice closure properties and decidable simulation preorder. Compared to our previous work [Belkhir et al.(2013)] we introduce a strictly more expressive service specification formalism thanks to the use of guarded transitions.

Contributions

In this paper we rely on automata-based techniques to tackle the problem of composition synthesis of parametrized services. We introduce an extension of automata called *parametrized automata* or PAs, that allows a natural specification and decidable synthesis of parametrized services. In PAs, the transitions are labeled by letters or variables ranging over an infinite alphabets and guarded by conjunction of equalities and disequalities. Besides, some variables can be refreshed in some states, that is, these variables can be released so that new letters can be bound to them. Refreshing mechanism is useful when computations start new sessions.

We introduce a simulation preorder for PAs and show its decidability. The proof relies on a game-theoretic characterization of simulation. We show how this result can be applied to the synthesis of a mediator for web services. Although not detailed here, the simulation decision procedure can help to solve language containment problems which are important ones in formal verification. The potential applicability of our model in verification also follows from the fact that PAs are closed under intersection, union, concatenation and Kleene operator. An advantage of PAs with respect to alternative automata models is the succinct service representation they permit, thanks to the use of variable and guards (e.g. with negative conditions). This benefit is formally supported in the detailed paper [Belkhir et al.(2014)]. Finally we have shown that for PAs the nonemptiness problem is PSPACE-complete, and the membership problem is NP-complete.

References

[Akroun et al.(2013)] Akroun, L., Benatallah, B., Nourine, L., Toumani, F., 2013. On decidability of simulation in data-centric business protocols. In: BPMW'13. Vol. 132. pp. 352–363.

[Alonso et al.(2004)] Alonso, G., Casati, F., Kuno, H., Machiraju, V., 2004. Web services: Concepts, Architectures and Applications. Springer-Verlag.

[Belkhir et al.(2014)] Belkhir, W., Chevalier, Y., Rusinowitch, M., 2013. Guarded Variable Automata over Infinite Alphabets. CoRR abs/1304.6297 (2013). `http://arxiv.org/abs/1304.6297`

[Belkhir et al.(2013)] Belkhir, W., Chevalier, Y., Rusinowitch, M., 2013. Fresh-variable automata: Application to service composition. In: 15th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, SYNASC. IEEE Computer Society, pp. 153–160.

[Berardi et al.(2005)] Berardi, D., Calvanese, D., De Giacomo, G., Hull, R., Mecella, M., 2005. Automatic Composition of Transition-based semantic Web Services with Messaging. In: Proc. 31st Int. Conf. Very Large Data Bases, VLDB 2005. pp. 613–624.

[Berardi et al.(2003)] Berardi, D., Calvanese, D., Giacomo, G. D., Lenzerini, M., Mecella, M., 2003. Automatic composition of E-services that export their behavior. In: ICSOC. pp. 43–58.

[Berardi et al.(2008)] Berardi, D., Cheikh, F., Giacomo, G. D., Patrizi, F., 2008. Automatic service composition via simulation. Int. J. Found. Comput. Sci. 19 (2), 429–451.

[Cheikh(2009)] Cheikh, F., 2009. Composition de services: algorithmes et complexité. Ph.D. thesis, Université Paul Sabatier, Thèse de doctorat.

[De Giacomo et al.(2013)] De Giacomo, G., Patrizi, F., Sardiña, S., Mar. 2013. Automatic behavior composition synthesis. Artif. Intell. 196, 106–142.

[Degano et al.(2012)] Degano, P., Ferrari, G. L., Mezzetti, G., 2012. Nominal automata for resource usage control. In: CIAA. pp. 125–137.

[Fearnley and Jurdzinski(2013)] Fearnley, J., Jurdzinski, M., 2013. Reachability in Two-Clock Timed Automata Is PSPACE-Complete. In: Fomin, F. V., Freivalds, R., Kwiatkowska, M. Z., Peleg, D. (Eds.), ICALP (2). Vol. 7966 of Lecture Notes in Computer Science. Springer, pp. 212–223.

[Feuillade and Pinchinat(2007)] Feuillade, G., Pinchinat, S., 2007. Modal specifications for the control theory of discrete event systems. Discrete Event Dynamic Systems 17 (2), 211–232.

[Hull and Su(2005)] Hull, R., Su, J., 2005. Tools for composite web services: a short overview. SIGMOD Record 34 (2), 86–95.

[Kaminski and Zeitlin(2010)] Kaminski, M., Zeitlin, D., 2010. Finite-memory automata with non-deterministic reassignment. Int. J. Found. Comput. Sci. 21 (5), 741–760.

[Lustig and Vardi(2009)] Lustig, Y., Vardi, M. Y., 2009. Synthesis from component libraries. In: de Alfaro, L. (Ed.), FOSSACS. Vol. 5504 of Lecture Notes in Computer Science. Springer, pp. 395–409.

[Martín et al.(2012)] Martín, J. A., Martinelli, F., Pimentel, E., 2012. Synthesis of secure adaptors. J. Log. Algebr. Program. 81 (2), 99–126.

[Nourine and Toumani(2012)] Nourine, L., Toumani, F., 2012. Formal approaches for synthesis of web service business protocols. In: Web Services and Formal Methods - 9th International Workshop, WS-FM 2012, Tallinn, Estonia, September 6-7, 2012, Revised Selected Papers. Vol. 7843 of Lecture Notes in Computer Science. Springer, pp. 1–15.