Journal of Symbolic Computation 69 (2015) 40–60

Contents lists available at ScienceDirect

Journal of Symbolic Computation

www.elsevier.com/locate/jsc

Parametrized automata simulation and application to service composition ☆

Walid Belkhir [a], Yannick Chevalier [b], Michael Rusinowitch [a]

[a] *INRIA Nancy–Grand Est & LORIA, 54600 Villers-lès-Nancy, France*
[b] *Université Paul Sabatier & IRIT Toulouse, France*

ARTICLE INFO

Article history:
Received 31 January 2014
Accepted 25 June 2014
Available online 27 September 2014

Keywords:
Parametrized automata
Service
Simulation preorder
Composition synthesis
Infinite alphabet

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 synthesize a mediator out of a simulation preorder. We also show that the Nonemptiness problem of PAs is PSPACE-complete.

1. Introduction

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; Reisig, 2008). Service composition is required when none of the existing services can fulfil 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.

☆ This research is partly supported by FP7 NESSoS project (Grant no. 256980), and extends substantially results presented at SYNASC 2013.

E-mail addresses: walid.belkhir@inria.fr (W. Belkhir), ychevali@irit.fr (Y. Chevalier), rusi@loria.fr (M. Rusinowitch).

http://dx.doi.org/10.1016/j.jsc.2014.09.029
0747-7171/© 2014 Elsevier Ltd. All rights reserved.

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 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.

1.1. 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 labelled 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 will be formally supported in the paper. Finally we have shown that for PAs the Nonemptiness problem is PSPACE-complete, and the membership problem is NP-complete.

1.2. Related work

Logic-based techniques for the synthesis problem (e.g. Feuillade and Pinchinat, 2007; Pathak et al., 2007) have been considered in the literature. Narayanan and McIlraith (2002) rely on the situation calculus and Golog logic programming language for automatic construction of Web services. Waldinger (2001) employs Snark theorem-prover to get software agents distributed over the net to cooperate and answer a query or perform a task. Many works have also addressed service composition by elaborating planning techniques from artificial intelligence (e.g. Pistore et al., 2005; Hoffmann, 2008; Hoffmann et