Complexity of synthesis of composite service with correctness guarantee

DENG Ting\(^{1,2}\)*, HUAI JinPeng\(^{1,3}\) & WO TianYu\(^{1,3}\)

\(^{1}\)State Key Laboratory of Software Development Environment, Beihang University, Beijing 100191, China; 
\(^{2}\)School of Mathematics and System Science, Beihang University, Beijing 100191, China; 
\(^{3}\)School of Computer Science and Engineering, Beihang University, Beijing 100191, China

Received December 13, 2010; accepted May 2, 2011

Abstract How to compose existing web services automatically and to guarantee the correctness of the design (e.g. temporal constraints specified by temporal logic LTL, CTL or CTL*) is an important and challenging problem in web services. Most existing approaches use the process in conventional software development of design, verification, analysis and correction to guarantee the correctness of composite services, which makes the composition process both complex and time-consuming. In this paper, we focus on the synthesis problem of composite service; that is, for a given set of services and correctness constraint specified by CTL or CTL* formula, a composite service is automatically constructed which guarantees that the correctness is ensured. We prove that the synthesis problem for CTL and CTL* are complete for EXPTIME and 2EXPTIME, respectively. Moreover, for the case of synthesis failure, we discuss the problem of how to disable outputs of environment (i.e. users or services) reasonably to make synthesis successful, which are also proved complete for EXPTIME and 2EXPTIME for CTL and CTL*, respectively.

Keywords business protocol, composite service, synthesis, environment, branching temporal logic


1 Introduction

Service-oriented computing has become a promising paradigm for realizing distributed applications, as more and more web services are being developed and published on the Internet based on SOAP\(^{1}\), WSDL\(^{2}\) and BPEL\(^{3}\). These services can serve as the reusable components for building complex applications. Recently, the web service composition issue has emerged as an important and challenging problem in Web service applications, which is concerned with how to combine existing web services when a client request cannot be satisfied by any individual service [1].

There have been many efforts towards automated service composition, and most of them are based on formal methods including automata theory, logical reasoning, planning in AI and theorem proving [1–6]. Most of these approaches require developers to provide a detailed specification of the desired behaviors

\(^{*}\)Corresponding author (email: dengting@act.buaa.edu.cn)

1) http://www.w3.org/TR/2007/REC-soap12-part0-20070427/
2) http://www.w3.org/TR/wsdl

© Science China Press and Springer-Verlag Berlin Heidelberg 2012  
info.scichina.com  www.springerlink.com
of a composite service (e.g., goal service in [2,3] or conversation protocol in [6]) with formal models or a
specification language (e.g., BPEL4WS). To ensure the correctness of the design, developers of a composite
service need to perform formal verification of the correctness constraints such as temporal properties [7,8].
The process of design, verification, analysis and correction makes the composition synthesis a difficult and
time-consuming task. So in this paper, we focus on the problem of synthesizing the composite service
from the library of services such that correctness constraints specified by temporal logic formulas are
guaranteed during the synthesis process.

Synthesis is the automated construction of a system from its specification. In the literature, a variety
of synthesis problem have been studied for closed system [9,10] and open systems [11–14]. A closed
system is a system whose behavior is completely determined by its own state. An open system is one
that interacts with its environment and whose behavior crucially depends on this interaction. Since
services interact with users and other services through message exchanges, they can be treated as open
systems taking users or services as their environments. In [11–14], systems are constructed from scratch
instead of composing from reusable components. However, almost every non-trivial system, especially
the Web-based system, is constructed using libraries of reusable components. In [15], the synthesis from
component library for LTL was studied in which only linear temporal properties are guaranteed.

In light of these, we investigate the synthesis of composite service from library of services for CTL and
CTL*, and establish their respective complexity bounds. CTL and CTL* are two kinds of branching
temporal logic which can support both branching and linear temporal properties. Furthermore, for the
case of synthesis failure; that is, when there exists no composite service over given library of services
which satisfies given CTL/CTL* formula, we observe that we can still construct the desired composite
service by restricting environment’s output behaviors reasonably, which is referred to as the environment-
controllable synthesis. Correspondingly, the former synthesis of composite service is referred to as
the environment-uncontrollable synthesis. We give formal treatments of these two synthesis problem and
establish the complexity bounds for them for CTL and CTL*, respectively.

The paper is organized as follow. Section 2 discusses related work. Section 3 gives some preliminary
definitions. Section 4 and Section 5 give formal investigation of the environment-uncontrollable synthesis
problem and environment-controllable synthesis problem, respectively. Section 6 concludes the work.

2 Related work

Several approaches have been proposed to automated synthesis of composite services. The Roman model
specified the goal service and e-services as finite state machine and reduced the composition synthesis
and world state of local database [3]. Fan et al. [4] modeled the goal service and services as alternating
finite automata and discussed the time complexity of composition synthesis by exploring connections
between composition synthesis and query rewriting using view. Mitra et al. [5] modeled services and goal
service as I/O automata, and reduced the existence problem of choreography to the simulation of I/O
automata. Fu et al. [6] modeled global behavior of e-service composition in asynchronous messages as
conversation based on Mealy machine and discussed the realizability and synchronization of conversations.
Above researches all require developer to provide a priori detailed behavior specification for composite
service. In contrast to researches above, we only require developer to provide the correctness constraint
on composition and automatically synthesize the behavior of composite service.

Huai et al. [16] and Pistore et al. [17] presented composition framework AutoSyn and Astro, and stud-
ied mediator-based synthesis of composite service from CTL formula and EAGLE formula, respectively.
In contrast, the composite scenario in this paper is not mediator-based, which means that composite
service can be invoked as a component by another composite service. Moreover, in order to handle com-
position failure, we study the restriction mechanism to environment behaviors which is not considered in
[16,17].

In addition, there are other papers on automated service composition from semantic or syntactic
description of services which consider a service as function with input and output, and use the classical