Model Checking of Non-centralized Automaton Web Service

Zhang Rui-min and Li Xiao-bin,

1 Lanzhou Institute of Technology, Department of Software, Gansu, Lanzhou, 730050, China

Abstract. In view of the web service model checking application, the combination mode of traditional finite state machines cannot guarantee the correctness of Web composite service. A web service model diction algorithm of non-centralized automaton based on satisfiability modulo theories (SMT) is proposed. First, SMT is used for bounded model checking of time automaton, and the time automaton model is directly converted into logical formula that can be identified by SMT, to make solution; secondly, the proposed SMT time automaton theory is used to achieve the employee travel arrangements web services for modeling and verification; finally, through an example analysis, the effectiveness of the algorithm on the termination of the path deadlock and the optimization of network parameters.

Keywords: Satisfiability modulo theories (SMT); Automaton; Web service; Model checking; Big data.

1 Introduction

The available Web services presented in the Internet can make use of integrate distributed mode to achieve the service system structure of distributed application [1]. Usually based on the workflow implementation to the value-added combination of basic Web composite services, establish distribution service invocation engine of different servers. Traditional workflow is a centralized call, which needs to be called and forward to all communication information in the process of implementation, so it needs great communication overhead. Therefore, compared with centralized, non-centralized operation mode has better operating performance [1].

2 Description of Web Composite Service Problem

The key problem of Web service composition optimization in Internet is to reduce the artificial participation. Data in the implementation transfer among multiple basic services, and Web service composition optimization can be expressed based on a group of state transition with the specific process is: to obtain input information from other services. After processing, it transfers the information for broadcasting, which is
similar to a kind of input and output sequence information transmission chain, so it is very suitable for using finite automaton for model checking and expression [8]. The relevant definitions are as follows: [8–9]:

Definition 1: The expression form of the finite automaton five-tuple is: $s A = (S, S^0, I, O, \Delta)$, among which, $S$ is automaton status set, $S^0 \subseteq S$ is the initial state of collection, $I$ and $O$ is automaton’s input and output set respectively, the conversion collection can be expressed as $\Delta \subseteq S \times T \times O \times S$, $\Delta = (s, i, o, s')$ means the initial state of the transformation as $s \in S$, $i \in T$ is input, $o \in O$ is output, $s' \in S$ is the purpose conversion state, can be expressed as: $s \xrightarrow{i, o} s'$, $(s, i, o, s') \in \Delta$.

Figure 1 shows the non-centralized implementation of the finite automaton. Service $s3$ can give a basic service for $r1, r2, r3$, indicating that the non-centralized implementation of automaton regards the input or output of the entire automaton as input or output of a target automaton.

![Non-centralized finite automaton](image)

Fig. 1. Non-centralized finite automaton

### 3 SMT Time Automaton

Given the limited model $M$ and the nature of the model $f$, the model checking problem can be formulated as the model $M$ to satisfy the model property $f$. The traditional model generation method generally gets larger state space, which is
constrained by the hardware conditions. The traditional model checking method is not practical for the large data form of Web composite service. In order to solve this problem, the bounded detection model can effectively solve the problem. The implementation of this model detection is given by Literature [11], that is, for the problem of the common model detection $M = E^f$, an example can be found in the limited implementation steps to meet model $M$ to satisfy the property $f$. For a given $k$, the above model checking can build up the problem to satisfy the problem of the logic formula. If the condition $[M, f]_k$ is satisfied, it is proved that the above mentioned $k$ steps are established to find a satisfying instance, and the model $M$ is synchronously proved to satisfy the model property $f$. Otherwise, the above $k$ steps cannot find satisfying example to meet the conditions, but cannot prove whether to meet the conditions of the case when $k' > k$.

Usually, the value $k$ is gradually increased until it finds an instance to meet the condition, or to achieve the maximum execution step. This detection method does not need to carry out in the whole state space, which can effectively save the computational memory space. Therefore, in the verification of time automaton, it does not need clock preprocessing to get limited space of time automaton. In this way, it is necessary to make solution based on the logic formula, so it is necessary to transform the automaton model into a logical model, and it needs to make variables Boolean formalized, which is not conducive to the simplification of the algorithm. Satisfiability Modulo Theories (SMT) can deal with non-Boolean variables, such as integer or real number. To this issue, this paper proposed a non-variable Boolean bounded time automaton model checking, and applied it to the optimization of Web composite services.

4 Experiment and Result Analysis

The algorithm of Literature [12] was selected to compare with the experiment, which is characterized by the synchronous modeling of the automaton and clock. Its application is quite extensive, such as the system with the clock which can realize the sharing mechanism of multi thread system variables. The computer hardware conditions for the experiment comparison are set as follows: CPU processor is Intel i3 2.4GHz; internal memory is 6G RAM, operating system is Ubuntu, and the test object is selected as shown in Figure 2. The comparison results of running time is seen in Table 1.
Fig. 2. Test object

Table 1. Comparison of running time

<table>
<thead>
<tr>
<th>Synchronization Process</th>
<th>Algorithm of this paper</th>
<th>Algorithm of Literature [10]</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>0m0.324s</td>
<td>0m0.025s</td>
</tr>
<tr>
<td>4</td>
<td>0m0.236s</td>
<td>0m0.053s</td>
</tr>
<tr>
<td>5</td>
<td>0m0.323s</td>
<td>0m0.055s</td>
</tr>
<tr>
<td>6</td>
<td>0m0.442s</td>
<td>0m0.112s</td>
</tr>
<tr>
<td>7</td>
<td>0m0.712s</td>
<td>0m0.298s</td>
</tr>
<tr>
<td>8</td>
<td>0m1.464s</td>
<td>0m1.357s</td>
</tr>
<tr>
<td>9</td>
<td>0m3.367s</td>
<td>0m5.795s</td>
</tr>
<tr>
<td>10</td>
<td>0m8.258s</td>
<td>0m24.653s</td>
</tr>
<tr>
<td>11</td>
<td>0m25.580s</td>
<td>1m37.951s</td>
</tr>
<tr>
<td>12</td>
<td>2m49.638s</td>
<td>6m43.834s</td>
</tr>
<tr>
<td>13</td>
<td>9m14.344s</td>
<td>Time&gt;60m</td>
</tr>
</tbody>
</table>

5 Conclusion

This paper is based on SMT to carry out a bounded model checking method for time automaton; the web service composition problem is optimized, and based on the non-centralized control mode, the finite automaton model is implemented by the non-centralized model. The basic communication process of Web services is realized
through the process of the automaton state migration. Based on the closure of automaton transfer mode to achieve the simulation of each state, and the service, the target and the communication cost are taken as input parameters of the model. By comparing the simulation results, it shows the effectiveness of the proposed algorithm in the barter model application, verifies the performance advantages of the algorithm of this paper. Because there is still the problem of access rights among the services in the practical application, it has certain effect of uncertainty on the actual performance. In the future, the research direction is the composition optimization of a large number of Web services.

References