

# 一种基于定理证明的 Web 服务合成方法研究

余强, 梁丽

(西华大学数学与计算机科学技术学院, 成都 610039)

**摘要:** 随着 Internet 中 Web 服务的不断增长, 如何通过对现存的服务进行合成, 以满足用户的个性化需求, 成为目前的研究热点。通过引入线性逻辑工具, 提出了一种新的 Web 服务合成解决方案, 通过定理证明形成对应的自动服务合成流程。示例证明了该方法的有效性。  
**关键词:** 线性逻辑; Web 服务合成; 定理证明

## Research on Theorem Proof Based Web Services Composition Method

YU Qiang, LIANG Li

(School of Math & Computer Science Technology, Xihua University, Chengdu 610039)

**【Abstract】** With the increase of Web services on the Internet, how to compose the existing services to satisfy the individual requirements becomes a hot research focus. A new solution for composing Web services through theorem proving is put forward by importing linear logic tools. The automatic composing flow of Web services is obtained according to the proof process. In the end, the validity of the new method is also proved by an example.

**【Key words】** Linear logic; Web services composition; Theorem proof

Web 服务是一种 Internet 上的组件, 它们是自包含、自描述、模块化、松耦合的应用, 可以发布、定位以及通过 Web 进行调用<sup>[1]</sup>。作为新一代的开放系统技术, Web 服务与 CORBA、Java 等技术相比, 在保证组件的互操作上更为成功。从经济角度上说, Web 服务的价值在于服务的重用, 重用的目的在于使服务增值。除了重用单个服务本身以外, 将多个服务按照一定的业务规则整合在一起, 形成新的服务, 这也是服务重用的另一种重要的方式, 而对应的过程就是 Web 服务的合成。

为使 Web 服务的合成进入真正实用的阶段, 使得 Web 服务实现跨组织、跨管理域的系统集成和自动交互, 工业界和学术界都进行了深入的研究, 总体说来有两种思路: (1) 引入工作流的思想, 通过定义流程来实现服务合成, 比如 BPEL4WS, 总的说来是一种静态的解决方案; (2) 借助 AI 的智能规划的方法。在该类方法中, 将 Web 服务合成形式化表示为五元组  $\langle S, S_0, G, A, \Gamma \rangle$ , 其中,  $S$  为状态空间,  $S_0$  和  $G (S_0 \subset S, G \subset S)$  分别表示初始和目标状态,  $A$  是可用的 Web 服务集合,  $\Gamma$  代表 Web 服务的功能集合 ( $\Gamma \subseteq S \times A \times S$ )。因此, Web 服务的合成就转化为状态空间的智能搜索问题。本文提出的方法, 属于后一种思路。研究重点是将线性逻辑<sup>[2]</sup> (Linear Logic, LL) 引入到 Web 服务的合成中, 通过定理证明的方式实现 Web 服务的合成。

### 1 背景知识

线性逻辑与经典逻辑不同的是, 它是一种可计算的、面向资源的逻辑。比如, 在线性逻辑的公式中,  $A \otimes A$  与经典逻辑中的  $A \wedge A \rightarrow A$  是不同的。尽管线性逻辑并不是第 1 种面向资源的逻辑 (其他的如 Relevance logic 和 Lambek 演算), 但是由于它对可计算语义方面有很好的支持, 因此目前作为一种逻辑语言和推理系统在诸如定理证明等领域得到广泛应用。

线性逻辑面向可计算资源的特性为 Web 服务的表示与合成提供了更加有力的支持。

#### 1.1 线性逻辑语法

本文用到的线性逻辑语法片段如下:

$$A ::= P \mid A \rightarrow A \mid A \otimes A \mid A \oplus A \mid !A \mid I$$

其中,  $A$  表示公式,  $P$  代表命题变元,  $I$  表示不需要任何资源就可完成的任务。本文用到的逻辑算子有  $\otimes$ 、 $\oplus$ 、 $\rightarrow$  和  $!$ , 它们分别代表并操作、或操作、线性逻辑蕴涵和无限制算子。

例如, 复合公式  $A \otimes B \rightarrow C \otimes D$ , 从资源的角度看, 表达出来的语义为: 当资源  $A$  和  $B$  得到满足后, 任务  $C$  和  $D$  才能执行。

#### 1.2 线性逻辑表达式

一个线性逻辑表达式由符号  $\pi$  分为两部分, 左边部分被看作可以提供消费的资源, 右边部分为将达到目标或要完成的任务。不严格地说, 线性逻辑表达式可以解释为: 当需要的资源得到满足后, 任务就能得到执行。

与经典逻辑相比较, 线性逻辑具有更强的表达能力, 能表示更多的系统特征。特别在 Web 服务能力的表示上, 相对于其他逻辑表示方法, 它具有更强的针对性。总结起来有:

(1) 线性逻辑支持 Web 服务中需消耗资源的定量表示, 如价格、时间等;

(2) 线性逻辑与 Web 服务合成建模的基础  $\pi$  演算模型, 有密切的联系。Abramsky 等对线性逻辑中的定理证明过程向  $\pi$  演算的转化作了深入的研究<sup>[3]</sup>。例如, 并操作  $\otimes$ 、或操作  $\oplus$ 、无限制算子  $!$  分别等价于  $\pi$  演算中的组合、选择求

**基金项目:** 四川省重点基金资助项目(01GG012-01)

**作者简介:** 余强 (1973-), 男, 讲师、博士生, 主研方向: 软件工程; 梁丽, 硕士、讲师

**收稿日期:** 2005-11-03 **E-mail:** yuqiang1973@163.com

和复制操作。

## 2 研究思路

### 2.1 基本步骤

本文介绍一种基于线性逻辑中的定理证明来实现 Web 服务的自动合成的思路<sup>[4]</sup>, 具体见图 1。

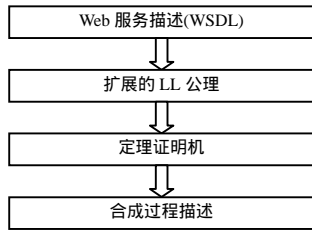


图 1 Web 服务自动合成过程

步骤 1 现存的以 WSDL 描述的原子 Web 服务(Atom Web Services)转化为基于线性逻辑的扩展逻辑公理, 服务合成的请求被表示为待证明的线性逻辑公式。

步骤 2 使用 MILL 定理证明机, 来验证上步得到的线性逻辑表达式能否在现存原子 Web 服务(扩展逻辑公理)和推理规则的基础上得到证明。由于 MILL 定理证明机自身的完备性, 因此它可以找到所有的合成方案。

步骤 3 由步骤 2 得到的证明过程中获得基于流程的合成过程描述。

本文的重点放在步骤 2 上。

### 2.2 基于线性逻辑的 Web 服务表示

本文中, Web 服务合成的需求描述表示为如下形式:  $(\Gamma_v, \Gamma_c); \Delta \quad I \rightarrow O$ 。  $\Gamma_v$  和  $\Gamma_c$  分别表示现存的增值 Web 服务和核心 Web 服务, 以扩展逻辑公理的形式进行表示。  $\Delta$  为非功能约束条件集合, 用于对定量和定性约束进行描述。  $I \rightarrow O$  是对合成 Web 服务的功能描述。其中,  $I$  表示的输入参数集合,  $O$  表示执行该合成 Web 服务后得到的输出参数集合。上式的语义可理解为: 给定可用的原子 Web 服务集合和相应的约束条件, 找到一个满足从输入到输出功能性要求的 Web 服务合成方案。需要注意的是, 对于每个原子 Web 服务而言, 其基本形式表示为:  $\Delta \quad I \rightarrow O$ 。在后文的示例中, 为了简化起见, 不考虑非功能性约束条件。

### 2.3 推理规则集

下面是在示例中将用到的线性逻辑片段的推理规则集:

$$\begin{array}{l}
 \text{Logical axiom and Cut rule:} \\
 A \vdash A \quad (Id) \quad \frac{\Gamma \vdash A \quad \Gamma', A \vdash \Delta'}{\Gamma, \Gamma' \vdash \Delta} \quad (Cut) \\
 \\
 \text{Rules for propositional constants:} \\
 \vdash 1 \quad \frac{\Gamma \vdash A}{\Gamma, 1 \vdash A} \\
 \frac{\Gamma, A, B \vdash C}{\Gamma, A \otimes B \vdash C} \quad (L\otimes) \quad \frac{\Gamma \vdash A \quad \Gamma' \vdash B}{\Gamma, \Gamma' \vdash A \otimes B} \quad (R\otimes) \\
 \frac{\Gamma \vdash A \rightarrow B}{\Gamma, A \vdash B} \quad (Shift) \quad \frac{\Gamma, A \vdash B}{\Gamma \vdash A \rightarrow B} \quad (R \rightarrow) \\
 \frac{\Gamma, A \vdash \Delta \quad \Gamma, B \vdash \Delta}{\Gamma, A \oplus B \vdash \Delta} \quad (L\oplus) \quad \frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A \oplus B} \quad (R\oplus)(a) \quad \frac{\Gamma \vdash B}{\Gamma \vdash A \oplus B} \quad (R\oplus)(b) \\
 \\
 \text{Rules for exponential !:} \\
 \frac{\Gamma \vdash \Delta}{\Gamma, !A \vdash \Delta} \quad (W!) \quad \frac{\Gamma, A \vdash \Delta}{\Gamma, !A \vdash \Delta} \quad (L!) \quad \frac{\Gamma, !A, !A \vdash \Delta}{\Gamma, !A \vdash \Delta} \quad (C!)
 \end{array}$$

## 3 应用实例

### 3.1 需求分析

假设有一个用户需要在线购车, 他需要采取分期付款的方式。当前存在的 Web 服务分别表示如下:

(1)GetAutoPrice(Brand<sup>i</sup>, Model<sup>i</sup>, TotalPrice<sup>o</sup>), 输入参数为汽车品牌 and 型号, 输出汽车价格, 是核心的 Web 服务;

(2)PaymentCompute(TotalPrice<sup>i</sup>, PayperMonthLimit<sup>i</sup>, LoanTimeLimit<sup>i</sup>, PayFirstTime<sup>o</sup>, PayperMonth<sup>o</sup>), 输入参数分别表示汽车价格、每月最大还款额、贷款期限, 输出结果为首付金额和每月还款额;

(3)SelectAuto(LengthLimit<sup>i</sup>, WidthLimit<sup>i</sup>, ExhaustLimit<sup>i</sup>, Brand<sup>o</sup>, Model<sup>o</sup>), 输入为汽车长度、宽度和排量, 输出汽车品牌和型号。

用户需求可表示为 (LenthLimit<sup>i</sup>, WidthLimit<sup>i</sup>, ExhaustLimit<sup>i</sup>, PayFirstTime<sup>o</sup>, PayperMonth<sup>o</sup>), 各个参数的含义同上。

可见, 该用户的需求是现存的任何一个 Web 服务都无法完全满足的。所以, 必须利用现有的 Web 服务进行合成。

3.2 线性逻辑表示  
用线性逻辑表达式对上述几个 Web 服务进行描述如下:

$$F_C = B \otimes M \rightarrow \text{GetAutoPrice } P ;$$

$$F_{v1} = P \otimes Ppml \otimes Ltl \rightarrow \text{PaymentCompute } Pft \otimes Ppm ;$$

$$F_{v2} = LI \otimes WI \otimes EI \rightarrow \text{SelectAuto } B \otimes M .$$

用户需求可表示为待证明的公式:

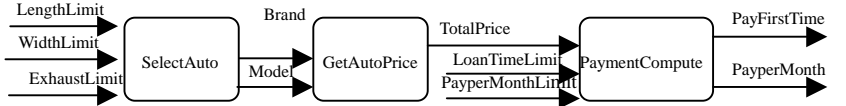
$LI \otimes WI \otimes EI \otimes Ppml \otimes Ltl \rightarrow Pft \otimes Ppm$ 。上述各个表达式中的符号缩写分别对应了相应的输入、输出参数。

3.3 证明过程  
证明过程如下:

$$\frac{
 \frac{
 \frac{
 \frac{
 B \otimes M \rightarrow \text{GetAutoPrice } P \quad LI \otimes WI \otimes EI \rightarrow \text{SelectAuto } B \otimes M
 }{
 B \otimes M \quad \text{GetAutoPrice } P \quad (Shift), \quad LI \otimes WI \otimes EI \quad \text{SelectAuto } B \otimes M \quad (Shift)
 }{
 LI \otimes WI \otimes EI \quad \text{SelectAuto}(\text{GetAutoPrice } P) \quad (Cut), \quad P \otimes Ppml \otimes Ltl \rightarrow \text{PaymentCompute } Pft \otimes Ppm
 }{
 P \otimes Ppml \otimes Ltl \quad \text{PaymentCompute } Pft \otimes Ppm \quad (Shift)
 }{
 LI \otimes WI \otimes EI \quad \otimes Ppml \otimes Ltl \quad \text{SelectAuto}(\text{GetAutoPrice}(\text{PaymentCompute } Pft \otimes Ppm)) \quad (Cut)
 }{
 LI \otimes WI \otimes EI \quad \otimes Ppml \otimes Ltl \rightarrow \text{SelectAuto}(\text{GetAutoPrice}(\text{PaymentCompute } Pft \otimes Ppm)) \quad (R^0)
 }$$

由上述证明过程得到合成的 Web 服务 SelectAuto、GetAutoPrice 和 PaymentCompute 的依次序线性组合, 见图 2。

图 2 合成的 Web 服务



## 4 结束语

与经典逻辑所不同的资源可计算性特点, 使得线性逻辑在 Web 服务的自动合成中扮演了重要的角色。文中运用定理证明的办法, 从状态组合空间中搜索到可用的自动 Web 服务合成方案。下一步的研究重点是把语义本体引入到逻辑表达和推理过程中。

### 参考文献

- W3CWorking Group. Web Services Architecture[EB/OL]. Http://www.w3.org/tr/2004/Note-ws-arch-20040211, 2004-02-11.
- Girard J Y. Linear Logic[J]. Theoretical Computer Science, 1987, 50 (1): 102.
- Abramsky S. Proofs as Processes[J]. Theoretical Computer Science, 1994, 135(1): 5-9.
- Rao Jinghai, Kungas P. Application of Linear Logic to Web Services Composition[C]. Proceedings of the 1<sup>st</sup> International Conference on Web Services, Lasvegas, USA, 2003-06.
- Lincoln P. Deciding Provability of Linear Logic Formulas[M]. London: Cambridge University Press, 1995: 109-122.