基于并发事务逻辑的语义Web服务组合

王雪松[1] 钱俊彦[2] 赵岭忠[2] 高荣亮[2]
(桂林电子科技大学电子工程与自动化学院 桂林 541004)[1]
(桂林电子科技大学计算机科学与工程学院 桂林 541004)[2]

摘 要 并发事务逻辑(Concurrent Transaction Logic,CTR)是一种谓词逻辑的扩展,支持语义Web服务自动组合的推理。采用并发事务逻辑作为表示和推理工具,给出了OWL-S功能和行为两个方面的Web服务组合方法。基于并发事务逻辑的执行语义及其Horn子集的过程式语义,提出了一个多项式时间的服务组合算法,从而降低了服务组合推理的复杂性,为解决当前许多主流语义Web服务组合方法不支持并发行为建模的问题提供了新思路。

关键词 语义Web服务,服务组合,并发事务逻辑,霍恩子句
中图法分类号 TP18 **文献标识码** A

Semantic Web Services Composition Based on Concurrent Transaction Logic

WANG Xue-song[1] QIAN Jun-yan[2] ZHAO Ling-zhong[2] GAO Rong-liang[2]
(School of Electronic Engineering and Automation,Guilin University of Electronic Technology,Guilin 541004,China)[1]
(School of Computer Science and Engineering,Guilin University of Electronic Technology,Guilin 541004,China)[2]

Abstract Concurrent Transaction Logic (CTR) is an extension of predicate logic which supports reasoning the automatic composition of semantic Web Services. This paper used CTR as the describing and reasoning tools,and proposed a composition method from the two aspects of OWL-S Web Services: function and behavior. The executing semantics of CTR and the procedural semantics of its Horn clause reduce the reasoning complexity. A polynomial time algorithm was constructed. This paper provided a new method for handling the problem that current mainstream semantic Web service composition methods are not able to model concurrent behaviors.

Keywords Semantic Web service,Service composition,Concurrent transaction logic,Horn clause

1 引言

Web服务是一种面向服务架构(SOA)的分布式计算模型,具有基于公开的标准、平台独立、应用独立以及能够共享数据和资源等特点。Web服务组合是将网络上分布的多个功能单一的Web服务,按某种业务逻辑自动组合起来提供增值服务。语义Web服务组合是探讨Web服务组合的主要趋势,利用本体给Web服务添加软件Agent可处理和理解的语义标记来实现Web服务组合的自动化,是当前的一个研究热点。

当前,形式化描述与自动推理技术相结合是语义Web服务组合研究的主要方法之一[1]。Rao等人在文献[6]中提出了一种基于线性逻辑定理证明的语义Web服务组合方法。该方法使用了外部和内部两种表示Web服务的技术,外部以DAML-S为描述语言,内部则以线性逻辑的公理和证明形式化地描述Web服务。通过线性逻辑语义推理,可直接生成Web服务组合的过程模型。Baader等人在文献[7]中提出用基于描述逻辑ALCQIO的动作机制推理语义Web服务的可执行性和投射。该方法使用ABox断言公理表示服务的前提条件,ABox断言和TBox原子概念联合表示状态迁移,并且证明了可执行和投射在非循环TBox下的推理为co-NP完全问题。鉴于描述逻辑不支持对服务动态行为的建模,史忠植等人在描述逻辑基础上构建了动态描述逻辑[8]SHOIN(D)来支持动作的描述和推理,并在此基础上提出了基于动态描述逻辑的语义Web服务推理研究。该方法从OWL-S Web服务功能和行为描述出发,使用基于SHOIN(D)的动作理论对Web服务建模。应用动态描述逻辑的推理机制,可分别对语义Web服务的可实现性、可执行性、投影、规划等问题进行推理。

然而以上大部分方法,比如线性逻辑和描述逻辑,甚至动态描述逻辑等,都有一个共同的缺陷:采用OWL-S作为服务的语义描述语言,但对OWL-S Web服务的重要特性"并发"却并不支持。CTR提供了并发算符和隔断执行算符,它们可方便地刻画并发以及同步并发性质,从而弥补了这个缺陷。

到稿日期:2011-12-02 返修日期:2012-03-27 本文受国家自然科学基金(61063002)和广西科学基金(2011GXNSFA018166,2011GXNSFA018164)资助。

王雪松(1981—),女,硕士,讲师,主要研究方向为人工智能、形式化方法,E-mail:wangxuesong@guet.edu.cn;钱俊彦(1973—),男,博士,教授,主要研究方向为软件工程、模型检验、嵌入式实时系统;赵岭忠(1977—),男,博士,教授,主要研究方向为人工智能、形式化技术、软件验证;高荣亮(1987—),男,硕士,主要研究方向为形式化技术。

另外，CTR 的执行语义及其 Horn 子集具有过程式语义，利用该性质可降低服务组合推理的复杂性。本文在此基础上，实现了一个基于 CTR 的多项式的服务组合算法。

2 发事务逻辑

并发事务逻辑(CTR)[2] 以一阶谓词逻辑为基础。CTR 的符号表包括：函数符号集合、谓词符号集合和变量集合。CTR 的联结词包括：∧、∨、¬、∀，顺序合取符号(serial conjunction)⊗、并发合取符号(concurrent conjunction)|，及一个隔断执行操作符号⊙。由于支持对顺序和并发行为的描述，CTR 可用于描述状态的迁移过程。

CTR 公式定义如下：

1. 每个原子公式 $p(t_1,\cdots,t_2,t_n)$ 都是一个 CTR 公式，其中 p 为谓词符号，t_i 为项。

2. 如果 ϕ 和 φ 是 CTR 公式，X 为变量，则：$\neg\phi$，$\odot\phi$，$(\forall X)\varphi$，$\phi\wedge\varphi$，$\phi\vee\varphi$，$\phi\otimes\varphi$，$\phi|\varphi$ 也是 CTR 公式。

与一阶逻辑不同的是，CTR 公式语义是在由状态组成的路径上定义的，公式的真值由路径而不是单个状态决定。当路径由一个状态组成时，CTR 就等价于经典逻辑的语义。CTR 公式在某路径上为真，表示该公式可沿着路径的第一个状态开始执行，直到最后一个状态停止。

定义 1(路径和多路) 在状态迁移系统中，如果执行一个动作后，系统由状态 D 变为 D'，由状态组成的序列$\langle DD'\rangle$就是一条路径 p，由多条路径组成的序列$\langle p,p'\rangle$称为多路。

定义 2(多路结构) 多路结构是 CTR 的解释模型，定义为二元组$\langle U,I_F,I_{path}\rangle$，其中 U 是值域，I_F 是函数符号和谓词符号的解释，I_{path} 是多路的解释，它将多路映射成$\langle U,I_F\rangle$语义结构。

多路结构为每一条多路赋予一个经典一阶逻辑语义结构，决定每条多路上哪些原子公式为真。反过来，这些原子公式决定了哪些复合公式在这条多路上为真。

定义 3(可满足性) M 是一个多路结构，π 是一条多路，则 CTR 公式 ϕ 在多路 π 上可满足记作 $M,\pi|=\phi$，定义如下：

1. $M,\pi|=p(t_1,\cdots,t_n)$ 当且仅当 $I_{path}(\langle\pi\rangle)|=^c p(t_1,\cdots,t_n)$，即原子公式 p 在对 $t_1,\cdots,t_n$ 的一组变量赋值下可以沿多路 π 执行；

2. $M,\pi|=\neg\phi$ 当且仅当 $M,\pi|=\phi$ 不成立；

3. $M,\pi|=\odot\phi$ 当且仅当 $M,\pi|=\phi$ 成立，且 π 是一条路径；

4. $M,\pi|=\forall X.\phi$ 当且仅当 $M,\pi|=\phi[X/a]$，即对变量 X 在 U 上的每一个赋值 a 情况下，公式 ϕ 在多路 π 上可满足；

5. $M,\pi|=\phi\wedge\varphi$ 当且仅当 $M,\pi|=\phi$ 和 $M,\pi|=\varphi$ 都成立；

6. $M,\pi|=\phi\vee\varphi$ 当且仅当 $M,\pi|=\phi$ 或 $M,\pi|=\varphi$ 成立；

7. $M,\pi|=\phi\otimes\varphi$ 当且仅当对多路 π_1,π_2 有 $M,\pi_1|=\phi$，$M,\pi_2|=\varphi$ 和 $\pi=\pi_1\cdot\pi_2$；

8. $M,\pi|=\phi|\varphi$ 当且仅当对多路 π_1,π_2 有 $M,\pi_1|=\phi$，$M,\pi_2|=\varphi$ 和 $\pi=\pi_1\parallel\pi_2$。

其中，$|=^c$ 表示经典蕴含，$\pi=\pi_1\cdot\pi_2$ 表示多路 π 由多路 π_2 和 π_1 首尾状态相连而成，即如果路径 $\pi_1=\langle D_1\cdots D_i\rangle$，$\pi_2=\langle D_{i+1}\cdots D_n\rangle$，则路径 $\pi=\langle D_1\cdots D_n\rangle$；$\pi=\pi_1\parallel\pi_2$ 表示多路 π 由多路 π_1 和 π_2 中的路按原顺序互相交错排列组成，如 $\pi_1=\langle D_1\cdots D_4\rangle$，$\pi_2=\langle D_6\cdots D_9\rangle$，则 $\pi=\pi_1\parallel\pi_2$ 的一种可能路径形式为：$\pi=\langle D_1D_2,D_6D_7,D_3D_4,D_8D_9\rangle$。

3 基于 CTR 的 OWL-S Web 服务建模

无论采用何种手段和方法组合 Web 服务，都必须对 Web 服务进行功能和行为描述。OWL-S[3] 是目前应用最广泛的 Web 服务描述的规范。OWL-S Profile 使用 IOPE 对 Web 服务的功能进行定义；OWL-S ProcessModel 使用 9 种控制结构对 Web 服务的行为进行定义。这些控制结构包括：顺序(Sequence)、并发(Split)、同步并发(Split+Join)、选择(Choice)和条件(If-Else)等。使用 CTR 对 Web 服务建模时，分别需要对 OWL-S 服务功能和行为建模。

3.1 OWL-S Web 服务功能的 CTR 建模

对 OWL-S Web 服务功能的建模主要是利用 CTR 描述 OWL-S Profile 中的 IOPE。I,O,P,E 分别代表 Web 服务的输入变量集合 $Input$、输出变量集合 $Output$、前提条件集合 $Precondition$ 和效果集合 $Effect$。前提条件和效果是表示环境的状态的子集。状态通常是断言的集合。为方便描述，不使用断言的形式描述状态，而采用$\{I,O\rangle$表示状态。

在 CTR 中，将每个 Web 服务视为一个动作，进一步将其分为原子动作和组合动作。对每一个动作，使用一个谓词 $p(type,I,O)$表示。其中，p 为动作的名称，$type$ 为表示动作类型的常量；I 代表动作的输入变量集合；O 代表动作的输出变量集合。

原子动作的动作类型 $type$ 包括以下两类：状态更新操作(su)和状态查询操作(sl)。前者通常是提供输入和产生输出的动作，会使系统的当前状态发生变化；执行后者不会使系统的状态发生变化。例如，对当前状态某种条件是否满足的查询操作。

3.2 OWL-S Web 服务行为的 CTR 建模

Web 服务的行为通常是体现服务动态特征的消息传递序列，可分为内部流程和外部交互。功能较复杂的服务常带有内部流程描述。该流程规定了服务内部动作的执行顺序，保证服务在单独执行和参与服务组合时的正确性和安全性。Web 服务的外部交互则是指参与服务组合的不同的原子服务之间的交互。外部行为呈现出的特征主要包括：顺序和并发。其中，并发性会导致组合服务中来自不同原子服务的内部流程在时间上出现交替。因此在描述并发行为时，希望最终的 Web 服务建模工具具有时序特性，从而方便表达不同原子服务内部流程的交替执行情形。CTR 的多路结构为并发执行的描述提供了保证，因此可以很好地刻画 Web 服务的外部交互。

OWL-S ProcessModel 为描述 Web 行为提供了 9 种控制结构。其中，并发和同步并发是两个重要性质。其区别在于：前者的所有子过程一经调度执行，组合过程就算完成了；而后者是参与并发的子过程必须全部执行结束，组合过程才结束。因此，结合并发和同步并发可以定义带子并发过程的并发过程，如图 1 所示。CTR 可以方便地描述并发和同步并发性质，例如对图 1，可以用$(S_2|S_5)$表示服务 S_2 和 S_5 并发，而$\odot(S_3|S_4)$表示服务 S_3 和 S_4 被隔断执行，即在 S_3 和 S_4 都执行结束之前，S_6 不能调度执行，是同步并发关系。

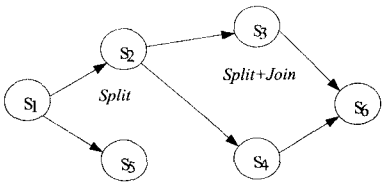

图 1　带同步并发的并发过程

OWL-S 条件结构 If(*cond*)-Then(ϕ)-Else(φ)中 *cond* 可以使用 CTR 中的查询操作来判断其是否满足。由于查询操作不改变状态，因此有 $cond \otimes \phi \equiv cond \wedge \phi$，其表示条件 *cond* 和动作 ϕ 分别在连续的两个相同状态下为真。于是条件结构可用 CTR 公式 $(cond \otimes \phi) \wedge (\neg cond \otimes \varphi)$ 刻画。

迭代结构 Iterate(φ)来可以使用以下两种方式在并发事物逻辑中编码：

1. $\varphi \leftarrow \phi \otimes \varphi$，递归的定义方式；

2. $\otimes^n \varphi$，连续的定义方式。

另外，本文认为用任意顺序(Any-Order)表达服务行为意义不大，而且可以使用其他结构组合构成。因此，对其不予采纳。其他控制结构使用 CTR 公式刻画的结果见表 1。

表 1　OWL-S ProcessModel 控制结构的 CTR 公式刻画

ProcessModel 控制结构	等价的 CTR 公式
Sequence(ϕ,φ)	$\phi \otimes \varphi$
Split(ϕ,φ)	$\phi \mid \varphi$
Split+Join(ϕ,φ)	$\odot(\phi \mid \varphi)$
Choice(ϕ,φ)	$\phi \vee \varphi$
If(cond)-Then(ϕ)-Else(φ)	$(cond \otimes \phi) \vee (\neg cond \otimes \phi)$
Iterate(ϕ)	$\varphi \leftarrow \phi \otimes \varphi$ 或 $\otimes^n \varphi$
Repeat(ϕ)-While(cond)	$\phi \leftarrow (cond \otimes \phi)$
Repeat(ϕ)-Until(cond)	$\phi \leftarrow (\phi \otimes cond)$

3.3　服务行为的约束条件建模

Web 服务行为的约束条件 *C* 通常是指除去输入条件以外，所有服务调度执行顺序上需要满足的特殊条件，是服务行为的某些性质，通常由服务设计者根据领域知识规定，对用户是透明的。约束条件 *C* 在 CTR 中的 3 种编码方式如下：

1. 原子约束：∇e 和 $\neg \nabla e$，*e* 为一个事件；

2. 顺序约束：$\nabla e_1 \otimes \nabla e_2$，$e_1$ 和 e_2 均为事件；

3. 复杂约束：$C_1 \wedge C_2$，$C_1 \vee C_2$，C_1 和 C_2 为原子或顺序约束。

这里采用事件发生的方式表示约束，即如果一个事件 *e* 必须在组合服务的执行路径上某个地方为真，那么可以用 $true \otimes e \otimes true$ 表示这个约束，简记为 ∇e。

本文将 Web 服务的行为约束条件和服务的行为描述通过合取词 $\wedge$ 融合成一个 CTR 公式。根据 CTR 公式的语义解释：$M,\pi \mid = \phi \wedge \varphi$ 当且仅当 $M,\pi \mid = \phi$ 和 $M,\pi \mid = \varphi$ 都成立，可知约束条件和动作在路径上同时为真。

4　语义 Web 服务组合的 CTR 推理

4.1　并发 Horn 公式

CTR 的 Horn 子集使 CTR 具有自动推理能力，因此，为了推理 Web 服务组合，本文在 OWL-S 服务行为的 CTR 刻画结果基础上使用并发 Horn 公式来进一步描述每一个 Web 服务。

定义 4(并发 Horn 目标)　并发事务逻辑的并发 Horn 目标定义如下：

1. 并发事务逻辑原子公式是并发 Horn 目标；

2. 如果公式 ϕ 和 φ 是并发 Horn 目标，那么 $\phi \vee \varphi$，$\phi \otimes \varphi$，$\phi \mid \varphi$ 是并发 Horn 目标；

3. 如果公式 ϕ 是并发 Horn 目标，那么 $\odot \phi$ 是并发 Horn 目标。

定义 5(并发 Horn 规则)　一个并发 Horn 规则是一个具有 $head \leftarrow body$ 形式的并发事务逻辑公式，规则头部 *head* 是一个原子公式，规则体 *body* 是一个并发 Horn 目标。

在并发事务逻辑的多路结构下，蕴含式 $p \leftarrow q$ 表示如果目标公式 *q* 能够沿某条路径执行，那么原子公式 *p* 也能够沿该路径执行。因此，*p* 常被看作是 *q* 的定义或者子程序。

用并发 Horn 公式推理 Web 服务组合时，对每一个 Web 服务，使用一条并发 Horn 规则公式 $p \leftarrow q$ 编码。其中 *p* 同样为一个谓词 $p(type, I, O)$，表示 Web 服务的名称；*q* 为一个并发 Horn 目标公式，表示该控制流图代表的服务的交互行为。例如，一个表示煮咖啡的过程可以用并发 Horn 公式表示为：

$$makecoffee \leftarrow grindCoffee \mid boilWater \otimes fillFilter \otimes pourWater$$

4.2　服务组合推理系统

基于 CTR 的语义 Web 服务推理主要涉及两个方面的内容：判定服务组合可满足性和制定服务组合方案。

定义 6(服务组合可满足性)　服务组合可满足性是指：给定一个表示 Web 服务外部交互的并发 Horn 目标公式 *G*、一个 Web 服务组合约束条件集合 $C=\{\vee_i C_i\}$ 和一个包含各个服务控制流图的并发 Horn 规则公式的集合 *R*，根据服务请求，判定是否存在一条路径 $s_1, s_2, \cdots, s_k$，使得公式 $G \wedge C$ 在 $s_1, s_2, \cdots, s_k$ 上为真。表示如下：

$$R, s_1, s_2, \cdots, s_k \mid = G \wedge (\vee_i C_i) \tag{1}$$

如果服务组合是可满足的，那么就可以构造出一个关于 $\wedge(\vee_i C_i)$ 的证明。在 CTR 中，这个构造性证明从公理开始，顺序使用若干条推理规则后，以 $G \wedge (\vee_i C_i)$ 结束。证明过程的每一步都是针对形式系统当前状态的一次查询操作或是一次改变当前状态的状态转换。因此，得到的证明序列就是满足全局约束 *C* 的服务交互行为 *G* 沿着路径 $s_1, s_2, \cdots, s_k$ 的一次执行。下面讨论式(1)的可满足性的证明。

首先，需要预处理约束和目标融合形成的公式 $G \wedge C_i$，因为其包含了并发 Horn 公式中没有使用的合取词。我们定义了一个简单的方法 $Translate(C_i, G) = G \wedge C_i$ 来消去合取词。

对于 *G* 中形如 $p \leftarrow q$ 的 Web 服务编码公式，有等式：

$$Translate(C_i, p) = Translate(C_i, q)$$

成立。因此，只需要对 *q* 使用 *Translate* 即可。

对 *q* 使用 *Translate*((,*q*)的情况如下：

· 若约束 $\sigma \in C$，且 $\sigma = \nabla e$ 或 $\neg \nabla e$

$Translate\ (\nabla e, e) \equiv e$

$Translate\ (\neg \nabla e, e) \equiv$ false

$Translate\ (\neg \nabla e, e') \equiv e'$

$Translate\ (\nabla e, \alpha \otimes \beta) \equiv Translate\ (\nabla e, \alpha) \otimes \beta \vee \alpha \otimes Translate\ (\nabla e, \beta)$

$Translate\ (\neg \nabla e, \alpha \otimes \beta) \equiv Translate\ (\neg \nabla e, \alpha) \otimes Translate\ (\neg \nabla e, \beta)$

$Translate\ (\nabla e, \alpha \mid \beta) \equiv Translate\ (\nabla e, \alpha) \mid \beta \vee \alpha \mid Translate\ (\neg \nabla e, \beta)$

$Translate(\neg\nabla e,\alpha|\beta) \equiv Translate(\neg\nabla e,\alpha) \mid Translate(\neg\nabla e,\beta)$

$Translate(\sigma,\alpha\vee\beta) \equiv Translate(\sigma,\alpha) \vee Translate(\sigma,\beta)$

$Translate(\sigma,\odot\alpha) \equiv\odot Translate(\sigma,\alpha)$

若约束 $\sigma=\nabla e_1\otimes\nabla e_2$

$Translate(\sigma,q) \equiv Translate(\nabla e_1\otimes send(\omega)) \wedge Translate(receive(\omega)\otimes\nabla e_2,q))$

这里的 *send* 和 *receive* 是系统的两个基本事件，分别用于向管道添加信号 ω 和删除信号 ω。只有在先前使用了事件 *send*，后面的 *receive* 才为真。

• 若约束 $\sigma=C_1\vee C_2$ 或 $\sigma=C_1\wedge C_2$，C_1 和 C_2 为 ∇e

$Translate(C_1\vee C_2,q) \equiv Translate(C_1,q) \vee Translate(C_2,q)$

$Translate(C_1\wedge C_2,q) \equiv Translate(C_1,Translate(C_2,q))$

对公式 $G\wedge(\vee_i C_i)$ 使用 *Translate* 预处理后得到的结果是一个 Horn 目标公式 G' 或者 false。后者表明服务行为和约束相冲突，服务组合不可满足。由于 G' 中的子公式中有可能含有 false，因此，还需要对出现 false 的情况做以下处理：

$false\otimes\alpha=\alpha\otimes false=false$

$false|\alpha= =false$

$false\vee\alpha= =false$

经过上述转换后，对 $G\wedge(\vee_i C_i)$ 的可满足性证明转换成对公式 G' 的可满足性证明。接下来本文对文献[5]中的 CTR 推理系统做适当扩展来证明 G' 的可满足性。CTR 服务组合推理系统包含 1 条公理和 3 条推理规则。公理和推理规则都由序列组成。

公理 $P,D\cdots|-(),D$ 为任意一个状态。

定义 7(序列) 假设 P 为一个并发 Horn 规则公式集合，D 为一个状态，$(\exists)\phi$ 为一个动作，那么称 $P,D\cdots|-(\exists)\phi$ 为一个序列。

序列的含义为：使用 P 中规则定义的动作 $(\exists)\phi$ 可以在一条以状态 D 开始的路径上执行。CTR 推理系统每条推理规则都由上、下两条序列组成，表示如果下面的序列成立，那么上面的序列也成立。

定义 8(可执行动作) 组合动作 φ 的当前可调度执行的原子动作集合称为组合动作 φ 的可执行动作，记为 $hot(\varphi)$。

$$hot(\varphi)=\begin{cases}\varphi,\varphi\text{ 为原子动作}\\ hot(\phi),\varphi=\phi\otimes\varphi\\ hot(\phi)\cup hot(\varphi),\varphi=\phi|\varphi\\ \phi\cup\varphi,\varphi=\phi\vee\varphi\\ \odot\varphi,\varphi=\odot\varphi\end{cases}\tag{2}$$

推理规则：规则 1－3 中，ϕ 和 ϕ' 均为并发 Horn 目标公式，ρ 为替换操作，D 和 D_1，D_2 表示状态，$a\in hot(\phi)$。

1. 使用动作定义

如果公式 $a\leftarrow b$ 是 P 中的规则，式(3)成立，其中，ϕ' 由 ϕ 中 a 替换为 b 得到。

$$\frac{P,D\cdots|-(\exists)\varphi'\rho}{P,D\cdots|-(\exists)\varphi}\tag{3}$$

2. 状态查询操作

如果 a 是一个状态查询操作，x 是 a 的一个变量，且有 $M,\langle D\rangle|=x$，公式(4)成立：

$$\frac{P,D\cdots|-(\exists)\phi'}{P,D\cdots|-(\exists)a(x)\otimes\phi'}\tag{4}$$

3. 状态更新操作

如果 $\odot a\in hot(\phi)$ 是一个状态更新操作，式(5)成立，其中，ϕ' 由 ϕ 删去 $\odot a$ 得到。

$$\frac{P,D_1\cdots|-(\exists)a\otimes\varphi'}{P,D_1\cdots|-(\exists)\varphi},\frac{P,D_2\cdots|-(\exists)\varphi'}{P,D_1\cdots|-(\exists)\varphi}\tag{5}$$

使用规则 1－3 证明 CTR 公式就是执行公式中动作。从初始状态 D 开始调度动作执行，到最后得到一个状态为空集，证明过程就结束了。不过服务组合的推理过程通常要求最终要产生一个输出结果。用上述公理和规则组成的 CTR 推理系统扩展至服务组合 G' 可满足性证明，即执行 G' 时，每次使用的推理规则是根据当前可执行动作 $hot(G')$ 中的动作类型决定。可执行动作(包括约束事件)总是一个状态更新或状态查询操作。为了正确地选择使用规则，做了如下约定：

非原子动作使用规则 1；

状态查询操作对应的原子动作使用规则 2；

其他原子动作使用规则 3。

基于上述约定，便可对 G' 从初始状态调度执行，如果最终到达一个包含所有服务请求输出变量的状态，则 G' 可满足。因此，状态的转换过程即为公式的证明过程，动作的执行序列即为需要的服务组合方案。

4.3 服务组合算法

利用 CTR 的推理规则 1－3 和使用约定，设计了 CTR Web 服务组合算法。该算法思想为基于深度优先搜索的回溯算法。从初始动作开始执行，直到动作执行不下去且目标仍未实现，执行其他子路径上的动作。如果没有别的子路径或所有子路径上动作都已执行过，则回溯至栈顶保存的祖先动作以便从新选取路径执行。

假设 Web 服务内部交互规则库为 R，状态由集合 S 表示，S 初始值为服务请求的输入变量集合，目标状态 *GOAL* 为服务请求输出变量集合。

CTR Web 服务组合算法描述如下。

输入：CTR Horn 目标公式 ϕ。

输出：公式 ϕ 的动作执行序列。

```
1. Begin
2. 将 φ 中所有出现在 R 中形如 a←b 的公式的左边的 a 使用定义 b 替换;
3. φ=φ;
4. do{
5. if(φ! =∅){
6.   计算 hot(φ)集合并从中选择一个未被执行过的原子动作 ω;
7.   if (ω. type==su)
8.     S=S−ω. I∪ω. O; // 删除动作 ω 消耗的输入，加入 ω 的输出结果;
9.   Push(ω);
10.  if(S==GOAL)算法结束;
11.    else {
12.      λ=φ //保存当前环境
13.      φ=φ/ω; //从公式 φ 中删去可执行动作 ω;
14.      }
15. else {
```

(下转第 156 页)

Egypt (Cairo): ACM, 2000:. 237-253

[4] Ramaswamy S, Rastogi R, Shim K. Efficient algorithms for mining outliers from large data sets [C]// Proc of ACM SIGMOD Int'l conf on Management of Data. Texas (Dallas): ACM, 2000: 427-438

[5] Angiulli F, Pizzuti C. Fast outlier detection in high-dimensional spaces [C]// Proc of Principles of Data Mining and Knowledge Discovery. 6th European Conf. Finland (Helsinki): ACM, 2002: 15-26

[6] Breunig M M, Kriegel H P, Ng R T. LOF: Identifying density-based local outliers [C]// Proc of ACM SIGMOD Int'l conf on Management of Data. Texas (Dallas): ACM, 2000: 93-104

[7] Bentley J L. Multidimensional binary search trees used for associative searching [J]. Communications of the ACM, 1975, 18 (9): 509-517

[8] Guttmann R. A dynamic index structure for spatial searching [C]// Proc of ACM SIGMOD int'l conf on Management of Data. New York (New York City): ACM, 1984: 47-57

[9] Berchtold S, Keim D, Kreigel H P. The X-tree: an index structure for high-dimensional data [C]// Proc of the 22nd int'l conf on VLDB. Mumbai (Bombay): Morgan Kaufmann, 1996: 28-39

[10] Knorr E M, Ng R T. Finding intentional knowledge of distance-based outliers [C]// Proc of 25th Int'l conf on VLDB. Scotland (Edinburgh): Morgan Kaufmann, 1999: 211-222.

[11] 张净,孙志挥,宋余庆,等. 基于信息论的高维海量数据离群点挖掘[J]. 计算机科学, 2011, 38(7): 148-151

[12] Bay S D, Schwabacher M. Mining distance-based outliers in near linear time with randomization and a simple pruning rule [C]// Proc of 9th ACM SIGKDD int'l conf on KDD. Washington DC: ACM, 2003: 29-38

[13] Angiulli F, Fassetti F. Very efficient mining of distance-based outliers[C]// Proc of ACM 6th conf on Information and Knowledge Management. Portugal (Lisboa): ACM, 2007: 791-800

[14] Hettich S, Bay S. KDD Cup 1999 Data[OL]. http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html, 2011-09-01

[15] Kaufman L, Rousseeuw P J. Finding Groups in Data: An introduction to Cluster Analysis [M]. New Jersey (Hoboken): John Wiley & Sons, 2005: 32-37

(上接第 142 页)

```
16.    Pop(ω);
17.    如果 γ 不是并发动作且动作 ω∈hot(γ)是状态更新操作
18.      S=S∪ω.I−ω.O; //撤销 ω 的执行
19.    如果 hot(γ)集合中还有未被执行过的原子动作
20.      φ=γ;
21.    else {
22.      如果 hot(γ)集合中原子动作是并发关系, ∀ω∈hot(γ)
23.        S=S∪ω.I−ω.O;
24.      Pop(ω);
25.      如果 λ 不是并发动作且动作 ω∈hot(λ)是状态更新操作
26.        S=S∪ω.I−ω.O;
27.  φ=λ;
28.    }
29.  }
30. }while(栈不为空)
31. End
```

算法时间复杂度分析:基于深度优先的回溯算法时间复杂度不高,通常可在线性时间 $O(n+m)$ 内完成。采用的数据结构不同,算法的时间复杂度也不相同。由于服务组合过程(构造服务调度执行的并发和顺序关系)实际上就是在对应于公式 ϕ 的控制流图上的深度优先搜索过程,因此可以借助控制流图来分析算法的时间复杂度。假设控制流图的深度为 n,宽度为 m,原子动作数为 e,则算法的时间复杂度为多项式级别,最坏情况下为:$m*n+e$。

结束语 本文阐述了基于并发事务逻辑 Horn 子集的语义 Web 服务组合方法。从 OWL-S Web 服务功能和行为两个方面对 Web 服务进行了表示与推理。重点分析了 Web 服务的并发和同步并发行为的建模方法。采用并发 Horn 公式作为组合推理形式,提出将服务组合方案的生成归结为并发 Horn 目标公式可满足性的证明,并给出了对约束的预处理方法。本文最后提出一个多项式时间的服务组合算法,并分析了算法的实践复杂度。仅用服务功能描述的基本输入输出变量作为表示服务行为的动作执行的效果,没有充分利用 OWL-S 提供的丰富语义描述,这是本方法的不足之处。下一步工作将把该算法用于具体网络应用中,对算法的实际运行效率进行实验,并考虑结合描述逻辑推理丰富 Web 服务的功能描述来组合 Web 服务。

参 考 文 献

[1] Rao J, Su X. A survey of automated Web service composition methods[C]// Proceedings of the First International Workshop on Semantic Web Services and Web Process Composition. San Diego, USA, 2004: 43-54

[2] Bonner A J, Kifer M. Concurrency and communication in transaction logic[C]// Proceedings of the Joint internal Conference and Symposium on Logic Programming. Bonn, Germany, MIT Press, 1996: 142-156

[3] OWL-S 1.2 Release [OL]. http://www.ai.sri.com/daml/services/owl-s/1.2/, 2011-11-20

[4] Horrocks I. Ontologies and the semantic Web [J]. Communications of the ACM, 2008, 51(12): 58-67

[5] Bonner A J, Kifer M. Transaction logic programming[R]. Computer Systems Research Institute Technical Report CSRI-323. University of Toronto

[6] Rao Jing-hai, Küngas P, Matskin M. Composition of Semantic Web services using Linear Logic theorem proving [J]. Information Systems, 2006, 31(4/5): 340-360

[7] Baader F, Lutz C, Milicic M, et al. A Description Logic Based Approach to Reasoning about Web Services[C]// Proceedings of the WWW Workshop on Web Service Semantics: Towards Dynamic Business Integration. Chiba, ACM Press, 2005: 636-647

[8] 史忠植. 常亮. 基于动态描述逻辑的语义 Web 服务推理[J]. 计算机学报, 2008, 31(9): 1600-1611