ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.19, No.4, April 2008, pp.967−980
DOI: 10.3724/SP.J.1001.2008.00967
© 2008 by *Journal of Software*. All rights reserved.
E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

用描述逻辑进行语义Web服务组合*

王杰生[1+], 李舟军[2], 李梦君[1]

[1](国防科学技术大学 计算机学院,湖南 长沙 410073)

[2](北京航空航天大学 计算机学院,北京 100083)

Composing Semantic Web Services with Description Logics

WANG Jie-Sheng[1+], LI Zhou-Jun[2], LI Meng-Jun[1]

[1](School of Computer, National University of Defence Technology, Changsha 410073, China)

[2](School of Computer Science and Engineering, BeiHang University, Beijing 100083, China)

+ Corresponding author: Phn: +86-10-82338247, E-mail: lizj@buaa.edu.cn

Wang JS, Li ZJ, Li MJ. Composing semantic Web services with description logics. *Journal of Software*, 2008, 19(4):967−980. http://www.jos.org.cn/1000-9825/19/967.htm

Abstract: Aiming at critical issues in the function-oriented composition of semantic Web services, such as the incapability of classic AI planning methods to handle the dynamically created individuals during Web services' executions and the inadequacy of service matchmaking based methods to fully exploit the semantic connections between types of services' I/O parameters. Following a comparative study of description logics and dynamic logics, Web services' IOPR's (inputs, outputs, preconditions and results) are encoded in axioms of description logics, and AI (artificial intelligence) planning method based on dynamic logic is extended to accommodate the transformation of a service composition problem into reasoning task of description logics. It finally overcomes difficulties in the classic AI planning method and defects in those methods based on service matchmaking.

Key words: semantic Web; Web service; description logics; service composition; automated reasoning

摘 要: 针对面向服务功能的语义Web服务组合问题,特别是经典的人工智能规划方法无法有效地处理Web服务执行过程中动态产生的新个体,以及基于服务匹配的方法则无法充分利用服务 I/O 参数类型之间大量的语义关联等关键问题,通过动态逻辑和描述逻辑之间的对比研究,采用描述逻辑公理来刻画 Web 服务的 IOPR(inputs,outputs, preconditions and results),扩展了基于动态逻辑的人工智能规划方法,提出了把语义 Web 服务组合问题转化为描述逻辑推理问题的方法,克服了经典的人工智能规划方法中的困难和基于服务匹配的服务组合方法的缺点.

关键词: 语义网;Web 服务;描述逻辑;服务组合;自动推理

中图法分类号: TP393 **文献标识码**: A

Web服务是万维网上一类非常重要的资源,随着语义Web各项技术,特别是语义Web本体标记语言OWL (Web ontology language)[1]的逐步发展和成熟,在语义Web中智能化和自动化地整合Web服务资源的需求日益凸

* Supported by the National Natural Science Foundation of China under Grant Nos.60473057, 60573057, 90604007, 60703075 (国家自然科学基金)

Received 2006-07-07; Accepted 2006-12-27

© 中国科学院软件研究所 http://www.jos.org.cn

显,语义Web服务也应运而生.在语义Web中智能化和自动化地整合Web服务资源,离不开基于语义的Web服务检索和面向Web服务的程序综合(program synthesis).面向Web服务的程序综合按其关注点的不同可分为两种类型:第 1 种类型是面向服务功能的服务组合,此类方案[2−4]基本上都采用经典的人工智能规划的方法,它们从服务输入输出参数的类型(type)以及服务执行的前提条件(precondition)和结果(result)的角度来产生满足查询的组合服务(composite service);第 2 种类型是面向服务行为的服务交响(service orchestration),此类方案[5,6]都或多或少地利用了模型检验的思想,它们从服务消息序列和消息参数类型的角度来产生满足查询的组合服务.

语义Web服务组合方法的研究主要集中于两个领域:一个是人工智能领域,另一个是形式化方法和自动推理领域,这两个领域的工作既互相交叉,又互为补充.人工智能领域中的工作主要集中在将规划算法[2−4]应用于服务组合(service composition),形式化方法和自动推理领域的工作则包括使用程序综合[7]和模型检验(model checking)[5,6]等方法设计各类服务组合方案.

目前,上述服务组合方案都不能在语义Web的环境中给出令人满意的、完整的解决方案.在基本层次上的服务组合方案,即基于服务参数类型进行服务组合的方案[7,8],都只利用了本体中的上下位关系(即类型间的包含关系),所以,这些服务组合算法在语义Web的环境中对类型间其他关系的利用是不完全的.对于超越了基本层次的服务组合方案,即进一步利用服务执行的前提条件和结果进行服务组合的方案[2−4],现有的服务组合方法都建立在经典的人工智能规划算法之上,无法有效地处理Web服务执行过程中动态产生的新个体.而且需要指出的是,对于超越了基本层次的服务组合方案,服务组合算法的适用性和可用性受限于描述服务执行的前提条件和结果的形式语言.特别地,一个表达能力很强的语言虽然能够适用于所有的Web服务,但也会导致服务组合算法无法终止.

本文通过动态逻辑和描述逻辑之间的对比研究,采用描述逻辑来刻画 Web 服务的输入输出参数类型以及 Web 服务执行的前提条件和结果,扩展了基于动态逻辑的人工智能规划方法,提出了把语义 Web 服务组合问题转化为描述逻辑推理问题的方法.在基于动态逻辑的人工智能规划方法中,刻画动作的描述逻辑公理比较简单,特别地,这些公理的前提中不会有模态词的出现.但是在语义 Web 服务组合中,由于 Web 服务可以接收多于一个的输入,所以,对于刻画 Web 服务的描述逻辑公理来说,它们的前提部分要复杂得多.通过利用描述逻辑中的传递角色实现 Web 服务输出到输入的绑定,我们屏蔽了这些复杂性.

文中的服务组合方案完全建立于描述逻辑之上,不仅能够避免经典的人工智能规划方法中的困难,而且能够充分利用本体中各种各样的语义关联.本文第 1 节简要介绍描述逻辑(description logics),并讨论它与模态逻辑之间的对应关系.第 2 节阐述采用描述逻辑进行服务组合的基本思路和相关结论.第 3 节进一步提出在现有的语义 Web 技术基础上进行 Web 服务组合的解决方案.第 4 节比较本文提出的服务组合方案和其他服务组合方案.最后是对本文工作的总结及未来工作的讨论.

1 描述逻辑

描述逻辑中最基本的两个术语是概念(concept)和角色(role).概念是一类事物的抽象,通常用 $A,B,C,D,\ldots$来表示;而角色则刻画了事物之间的各种联系,通常用 $P,Q,R,S,\ldots$来表示.描述逻辑中的真假通常用$\top$和$\bot$分别来表示,析取和合取分别用$\sqcup$和$\sqcap$来表示,而蕴含和等价则分别用$\sqsubseteq$和$\equiv$来表示.

1.1 描述逻辑$\mathcal{ALC}$和$\mathcal{SHOIQ}$

$\mathcal{ALC}$是一种简单但却非常重要的描述逻辑.很多更为复杂、表达能力更强的描述逻辑都可以通过扩展$\mathcal{ALC}$得到.$\mathcal{ALC}$的 BNF 语法如下所示(其中,A 和 P 分别表示原子概念和原子角色):

$$C,C'::=A|\top|\bot|\neg C|C\sqcup C'|\exists R.C$$

$$R::=P$$

对于另外一些常见的描述逻辑构造子(例如$\forall$,$\sqcap$,$\circ$,$^+$和*等等)生成的概念,存在一个由上述语法产生的概念与之等价.表 1 列出了这些构造子生成的概念以及它们的等价概念.

© 中国科学院软件研究所 http://www.jos.org.cn

Table 1 Some equivalent concepts in description logics

表 1 一些等价的描述逻辑概念

Constructors	Formed concepts	Equivalent $\mathcal{ALC}$ concepts
$\forall$	$\forall R.C$	$\neg\exists R.\neg C$
$\sqcap$	$C\sqcap C'$	$\neg(\neg C\sqcup\neg C')$
$\sqcup$ (roles)	$\exists R\sqcup R'.C$	$\exists R.C\sqcup\exists R'.C$
$\circ$ (roles)	$\exists R\circ R'.C$	$\exists R.\exists R'.C$
$^+$ (roles)	$\exists R^+.C$	$\exists R.C\sqcup\exists R.\exists R.C\sqcup\exists R.\exists R.\exists R.C\sqcup\ldots$
* (roles)	$\exists R^*.C$	$C\sqcup\exists R.C\sqcup\exists R.\exists R.C\sqcup\exists R.\exists R.\exists R.C\sqcup\ldots$

表 2 给出了$\mathcal{ALC}$的语义.特别地,我们还在表 2 中列出了动态逻辑PDL(propositional dynamic logic)的语义以方便下文对照.如表 2 所示,一个概念C在解释I下的语义为论域Δ^I的一个子集$C^I\subseteq\Delta^I$,一个角色R在解释I下的语义为论域Δ^I上的一个二元关系$R^I\subseteq\Delta^I\times\Delta^I$.注意,$\Delta^I$是一个非空集合.

Table 2 Correspondence between $\mathcal{ALC}$ concepts and PDL formulas

表 2 $\mathcal{ALC}$ 概念和 PDL 公式的对应关系

$\mathcal{ALC}$ concepts	PDL formulas	$\mathcal{ALC}$ semantics	PDL semantics
$\top$	True	Δ^I	W
$\bot$	False	$\varnothing$	$\varnothing$
$\neg C$	$\neg C$	$\Delta^I\backslash C^I$	$W\backslash V(C)$
$C\sqcup C'$	$C\vee C'$	$C^I\cup C'^I$	$V(C)\cup V(C')$
$C\sqcap C'$	$C\wedge C'$	$C^I\cap C'^I$	$V(C)\cap V(C')$
$\exists R.C$	$\langle R\rangle C$	$\{a\mid a\in\Delta^I\wedge\forall b.(a,b)\in R^I\rightarrow b\in C^I\}$	$\{w\mid w\in W\wedge\forall w'.(w,w')\in V(R)\rightarrow w'\in V(C)\}$
$\forall R.C$	$[R]C$	$\{a\mid a\in\Delta^I\wedge\exists b.(a,b)\in R^I\wedge b\in C^I\}$	$\{w\mid w\in W\wedge\exists w'.(w,w')\in V(R)\wedge w'\in V(C)\}$

描述逻辑使用词汇公理(terminological axiom)来刻画描述逻辑中的各个概念之间的包含关系,其最基本的形式是 $A\sqsubseteq B$,直观的解释为概念 B 的外延包含了概念 A 的外延.我们用 $A\equiv B$ 来表示"$A\sqsubseteq B$ 并且 $B\sqsubseteq A$",特别地,如果 A 是原子概念,则称 B 是 A 的定义.

作为OWL(Web ontology language)的理论基础的描述逻辑$\mathcal{SHOIQ}$,它是$\mathcal{ALC}$的一种扩展,$\mathcal{S}$代表$\mathcal{ALC}$和传递角色,$\mathcal{H}$代表角色层次(role hierarchy).角色层次由一组用于刻画角色的词汇公理指定.遵循表 2 中的记号,如果角色R和S满足$R^I\subseteq S^I$,则称角色R包含于角色S中,并用$R\sqsubseteq S$表示此包含关系;一个角色R称为是传递的,如果对于任意的$(a,b)\in R^I,(b,c)\in R^I$都有$(a,c)\in R^I$.

1.2 描述逻辑的推理问题

仍然遵循表 2 中的记号,如果对于概念C,解释I使得C^I不是空集,则称解释I满足概念C,或称概念C是可满足的;令Γ为一个词汇公理集合,如果Γ中任意一个公理$A\sqsubseteq B$在解释I下都满足$A^I\subseteq B^I$,则称解释I为Γ的一个模型;如果存在词汇公理集合Γ的一个模型I使得C^I不是空集,则称概念C是关于Γ可满足的;如果对于词汇公理集合Γ的每一个模型I都有$C^I\subseteq D^I$,则称概念C"Γ包含"于概念D中,记为$\Gamma\models C\sqsubseteq D$.

概念的可满足问题和概念的"Γ包含"问题可相互归约.在给出相关定理之前,先说明几个简记符号.对于一组概念$\{C_1,C_2,\ldots,C_n\}$,$\sqcap_iC_i$表示$C_1\sqcap C_2\sqcap\ldots\sqcap C_n$,而$\sqcup_iC_i$则表示$C_1\sqcup C_2\sqcup\ldots\sqcup C_n$.对于一组角色$\{R_1,R_2,\ldots,R_n\}$,$\sqcap_iR_i$表示$R_1\sqcap R_2\sqcap\ldots\sqcap R_n$,而$\sqcup_iR_i$则表示$R_1\sqcup R_2\sqcup\ldots\sqcup R_n$.

定理 1. 设在词汇公理集合Γ,概念C和D中出现的角色集合为$\{R_1,R_2,\ldots,R_n\}$,则$\Gamma\models C\sqsubseteq D$当且仅当概念$\forall(\sqcup_iR_i)^*.(\sqcap_{A\sqsubseteq B\in\Gamma}\neg A\sqcup B)\sqcap C\sqcap\neg D$是不可满足的.

证明:见文献[9]. □

上述定理称为内化(internalization)定理.所谓内化,就是一种在保持语义的条件下用一个概念对一个公理集合进行编码的技术.在定理 1 中,概念$\forall(\sqcup_iR_i)^*.(\sqcap_{A\sqsubseteq B\in\Gamma}\neg A\sqcup B)\sqcap C\sqcap\neg D$是不可满足的意味着如果$\Gamma$中的每个$A\sqsubseteq B$成立,那么$C\sqsubseteq D$的反面,即$C\sqcap\neg D$是不可能的.描述逻辑$\mathcal{SHOIQ}$没有角色构造子"*",但是,我们仍然可以对公理集合进行内化.

定理 2. 设在词汇公理集合Γ,概念C和D中出现的角色集合为$\{R_1,R_2,\ldots,R_n\}$,角色U是传递的,并且$R_1\sqsubseteq U$,$R_2\sqsubseteq U,\ldots,R_n\sqsubseteq U$,则$\Gamma\models C\sqsubseteq D$当且仅当概念$(\sqcap_{A\sqsubseteq B\in\Gamma}\neg A\sqcup B)\sqcap\forall U.(\sqcap_{A\sqsubseteq B\in\Gamma}\neg A\sqcup B)\sqcap C\sqcap\neg D$是不可满足的[10].

© 中国科学院软件研究所 http://www.jos.org.cn

证明:我们来证明概念$\forall(\sqcup_i R_i)^*.(\sqcap_{A\sqsubseteq B\in\Gamma}\neg A\sqcup B)\sqcap C\sqcap\neg D$可满足当且仅当概念$(\sqcap_{A\sqsubseteq B\in\Gamma}\neg A\sqcup B)\sqcap\forall U.(\sqcap_{A\sqsubseteq B\in\Gamma}\neg A\sqcup B)\sqcap C\sqcap\neg D$可满足.

一方面,如果概念$\forall(\sqcup_i R_i)^*.(\sqcap_{A\sqsubseteq B\in\Gamma}\neg A\sqcup B)\sqcap C\sqcap\neg D$可满足,我们把$U$取为$(\sqcup_i R_i)^*$,则概念$(\sqcap_{A\sqsubseteq B\in\Gamma}\neg A\sqcup B)\sqcap\forall U.(\sqcap_{A\sqsubseteq B\in\Gamma}\neg A\sqcup B)\sqcap C\sqcap\neg D$同样可以满足;另一方面,如果概念$(\sqcap_{A\sqsubseteq B\in\Gamma}\neg A\sqcup B)\sqcap\forall U.(\sqcap_{A\sqsubseteq B\in\Gamma}\neg A\sqcup B)\sqcap C\sqcap\neg D$可满足,设$Id^I=\{(a,a)|a\in\Delta^I\}$,则$(\sqcup_i R_i)^*\sqsubseteq U^*=Id\cup U$.即对于任意解释$I$,如果它能够满足$(\sqcap_{A\sqsubseteq B\in\Gamma}\neg A\sqcup B)\sqcap\forall U.(\sqcap_{A\sqsubseteq B\in\Gamma}\neg A\sqcup B)\sqcap C\sqcap\neg D$,那么,它也能够满足$\forall(\sqcup_i R_i)^*.(\sqcap_{A\sqsubseteq B\in\Gamma}\neg A\sqcup B)\sqcap C\sqcap\neg D$. □

1.3 描述逻辑与动态逻辑的对应关系

描述逻辑大多用于静态事物的描述,而动态逻辑则多见于动态系统的刻画.Web 服务作为构件化的分布式程序,使用动态逻辑来刻画似乎更加合理;但另一方面,在语义 Web 环境中的 Web 服务是采用描述逻辑来描述的,并且和 Web 服务相关的领域本体中的概念都是采用描述逻辑来描述的.因此,为了与语义 Web 环境相适应以充分利用领域本体中的大量信息,我们采用描述逻辑作为服务组合的工作语言.加之描述逻辑和动态逻辑有如下所述的对应关系,因此,用描述逻辑进行语义 Web 服务组合便是名正言顺了.

我们已经在表 2 中列出了$\mathcal{ALC}$与动态命题逻辑PDL的对应关系[11].动态逻辑的公式由一组命题变量和原子程序构成,分别对应于描述逻辑中的概念和角色.在表 2 中,PDL的语义由一个Kripke结构$\langle W,S,V\rangle$来解释,第 1 个分量W是所有可能世界的集合,第 2 个分量$S\subseteq W\times W$是可能世界之间的迁移关系,而第 3 个分量$V:Q\cup P\rightarrow 2^W\cup 2^S$把$Q$中的一个命题变量映射到可能世界集合$W$的一个子集,把$P$中的一个原子程序映射到一个迁移关系$S$的一个子集.可见,在描述逻辑中用于指称Web服务的角色与在动态逻辑中用于指称程序语句的原子程序(atomic program)在逻辑推理中的作用是一样的,只是呈现的形式有所不同;并且,在描述逻辑中用于刻画Web服务参数类型的概念的作用也十分类似于在动态逻辑中用于刻画各个可能世界(possible world)的命题变量(propositional variable).稍有不同的是,描述逻辑中的推理过程反映了Web服务执行前后个体及其类型的变迁,而动态逻辑的推理过程反映了程序执行前后可能世界及其状态的变迁.

事物之间既有静态的联系也有动态的联系.比如从 Web 服务输入到 Web 服务输出的迁移是一种动态的联系,而一个智能主体(intelligent agent)对事物的认识(比如事物的属性)则是一种静态的联系.文献[12]中提出的“动态描述逻辑(dynamic description logic)”便从形式上区分了事物之间的静态联系和动态联系.普通的描述逻辑并不作这样的区分,因为事物之间的静态联系和动态联系不是对立的.例如,Web 服务从输入产生输出本是动态的,但是如果把 Web 服务当作智能主体,那么 Web 服务的输出就构成了它们对 Web 服务输入的认知,而这种认知则是一种静态的联系.由此可见,使用描述逻辑中的角色和概念来刻画 Web 服务和服务输入输出参数的类型,不仅逻辑上是正确的,而且认知上也是合理的.

2 用描述逻辑进行语义 Web 服务组合

服务组合离不开服务描述,本节将首先介绍语义Web服务描述语言OWL-S[13],然后使用描述逻辑中的角色和概念分别对Web服务和Web服务的参数类型进行形式化,最后利用描述逻辑推理进行服务组合.

2.1 OWL-S的服务轮廓(service profile)和服务模型(service model)

OWL-S 是建构于 OWL 之上的用于描述 Web 服务的标记语言.OWL-S 使用 IOPR(inputs,outputs, preconditions and results)来刻画 Web 服务的功能,即 Web 服务输入(I)输出(O)参数的类型、Web 服务的能够正确执行的前提条件(P)以及 Web 服务执行后产生的结果(R).其中,Web 服务输入和输出参数的类型 I/O 对应于一组描述逻辑的概念,而 PR 则是分别对应一组表达式,这组公式可以是描述逻辑中的概念,也可以是其他公式.例如 KIF(knowledge interchange format)公式和 SWRL (semantic Web rule languge)公式等等.当然,如果 PR 中的表达式也是描述逻辑概念,那么实际上就没有区分 I/O 和 PR 的必要,都以 I/O 待之即可.

由于 Web 服务的执行路径通常存在多个分支,所以,Web 服务执行后产生的结果通常由一组条件效应(conditional effect)子句来刻画.每个条件效应子句呈现了 Web 服务的一个视图,即 Web 服务在特定的前提条件

© 中国科学院软件研究所 http://www.jos.org.cn

下执行得到的特定结果.Web 服务具体呈现哪个视图则取决于 Web 服务选取的执行路径分支.

OWL-S 把 Web 服务模型区分为原子进程(atomic process)、组合进程(composite process)和简单进程(simple process),分别对应于 Web 服务、组合服务和抽象服务.原子进程可以通过 Web 服务消息进行直接的交互,而组合进程则可以通过OWL-S提供的一组构造子在原子进程的基础上生成.简单进程是一种抽象,由于Web服务通常具有多个视图,这使得我们难以用描述逻辑将 Web 服务的功能简单地刻画出来,通过使用简单进程对 Web 服务的各个服务视图进行抽象,我们可以把一个 Web 服务分解为几个抽象服务加以刻画,所以,简单进程的抽象作用就显得十分重要.由于这些抽象服务由一个输出参数类型和 1 到多个输入参数类型所刻画,因此非常适合于用描述逻辑来刻画.

清单 1 中描述的Web服务S由两个条件效应($ce1$ 和$ce2$)来刻画.条件效应$ce1$ 指出:如果输入参数in的类型是I_1,那么输出参数out的类型是A;条件效应$ce2$ 指出:如果输入参数in的类型是I_2,那么输出参数out的类型是B.即Web服务S有两个服务视图,分别对应条件效应$ce1$ 和条件效应$ce2$.因此,Web服务S可以被抽象为两个抽象服务S_1和S_2,抽象服务S_1具有类型为I_1的输入参数 in_{S_1} 和类型为A的输出参数 out_{S_1},而抽象服务S_1具有类型为I_2的输入参数 in_{S_2} 和类型为B的输出参数 out_{S_2}.

清单 1. Web 服务 S 的 OWL-S 描述.

```
〈process:AtomicProcess rdf:ID="S"〉
  〈process:hasInput〉
    〈process:Input rdf:ID="in"/〉
  〈/process:hasInput〉
  〈process:hasOutput〉
    〈process:Output rdf:ID="out"/〉
  〈/process:hasOutput〉
  〈process:hasResult〉
〈process:Result rdf:ID="ce1"〉                    (服务视图 1)
      〈process:inCondition〉
        〈owlexpr:Condition〉
          〈expr:expressionObject〉
            〈rdf:Description rdf:about="#in"〉
            〈rdf:type rdf:resource="#I1"/〉
          〈/rdf:Descripton〉
          〈/expr:expressionObject〉
        〈/owlexpr:Condition〉
      〈/process:inCondition〉
      〈process:hasEffect〉
        〈owlexpr:Condition〉
          〈expr:expressionObject〉
            〈rdf:Description rdf:about="#out"〉
            〈rdf:type rdf:resource="#A"/〉
          〈/rdf:Descripton〉
          〈/expr:expressionObject〉
        〈/owlexpr:Condition〉
      〈/process:hasEffect〉
    〈/process:Result〉
```

© 中国科学院软件研究所 http://www.jos.org.cn

```
〈process:Result rdf:ID="ce2"〉          (服务视图 2)
    〈process:inCondition〉
      〈owlexpr:Condition〉
        〈expr:expressionObject〉
        〈rdf:Description rdf:about="#in"〉
          〈rdf:type rdf:resource="#I2"/〉
        〈/rdf:Descripton〉
        〈/expr:expressionObject〉
      〈/owlexpr:Condition〉
    〈/process:inCondition〉
   〈process:hasEffect〉
      〈owlexpr:Condition〉
        〈expr:expressionObject〉
          〈rdf:Description rdf:about="#out"〉
          〈rdf:type rdf:resource="#B"/〉
        〈/rdf:Descripton〉
        〈/expr:expressionObject〉
      〈/owlexpr:Condition〉
    〈/process:hasEffect〉
  〈/process:Result〉
 〈/process:hasResult〉
〈/process:AtomicProcess〉
```

在我们的语义 Web 服务组合方法中,现实的语义 Web 服务总是先被抽象成若干个抽象服务,并用描述逻辑公理对这些抽象服务加以刻画,然后把这些抽象服务当作现实的语义 Web 服务的替身参与到 Web 服务组合中去.以后行文中若无特别说明,Web 服务皆指抽象服务.

2.2 从服务描述到Web服务的描述逻辑编码

动态逻辑能够有效地刻画人工智能规划中的动作[14].假设动作a能够正确执行的前提为p,它执行后产生的效应是q,那么,该动作可以用下面的动态逻辑公式加以刻画:

$$p\rightarrow[a]q \tag{1}$$

式(1)表明:如果 p 成立并且动作 a 被执行,那么,动作 a 执行后 q 成立.下面的动态逻辑公式进一步保证只有在 p 成立的情况下动作 a 才会执行:

$$\langle a\rangle \text{true}\rightarrow p \tag{2}$$

式(1)和式(2)的一种等价表述是[14]

$$\text{true}\rightarrow[a]q \tag{3}$$

$$\langle a\rangle \text{true}\rightarrow p \tag{4}$$

用描述逻辑刻画 Web 服务的方法和上述方法是十分类似的.假设有一个简单的 Web 服务 S,其输出参数的类型是 O,两个输入参数的类型分别是 A 和 B.下面的词汇公理刻画了 Web 服务 S 执行的结果:

$$\top\sqsubseteq\forall S.O \tag{5}$$

式(5)指出,如果执行了 Web 服务 S,那么,Web 服务 S 将产生一个类型为 O 的输出.而下面的词汇公理将保证 Web 服务 S 仅在输入正确的情况下执行,即

$$\exists S.\top\sqsubseteq I_S \tag{6}$$

式(6)想要表达的意思是,如果执行了 Web 服务 S,那么在 Web 服务 S 执行之前的那一刻,必有一个 A 类型的

© 中国科学院软件研究所 http://www.jos.org.cn

对象和一个 B 类型的对象.为此,我们必须弄清 Web 服务 S 输入的来源.

从输入和输出的角度来看,对象的属性有着与Web服务一样的特性.因而,所有Web服务的输入或直接来自查询提供的输入,或是某个Web服务产生的输出,或是来自这些对象的某个属性.令***initial***为一个保留角色(下一节将给出与***initial***相关的词汇公理),用于刻画查询提供的输入.就这个简单的例子而言,角色$(\boldsymbol{initial}\sqcup S)^+$显然足以囊括Web服务$S$所有的输入来源,所以

$$I_S\equiv\exists(\boldsymbol{initial}\sqcup S)^+.A\sqcap\exists(\boldsymbol{initial}\sqcup S)^+.B \tag{7}$$

在人工智能规划中,一个动作可以选择不执行,即使它的前提能够被满足.但是,Web 服务组合更加关注 Web 服务执行的可能性,下面的词汇公理用于保证每个合法输入都能让 Web 服务执行:

$$I_S\sqsubseteq\exists S.\top \tag{8}$$

因此,Web 服务 S 的描述逻辑编码即是式(5)~式(8).这个例子表明,一个 Web 服务可以用一组词汇公理来编码,下面我们给出正式的定义:

定义 1(Web服务的描述逻辑编码). 令$\mathcal{C}(\mathcal{T})$和$\mathcal{R}(\mathcal{T})$分别是本体$\mathcal{T}$中所有原子概念和原子角色的集合,$S\subseteq\mathcal{R}(\mathcal{T})$是所有抽象服务的集合,保留角色$\boldsymbol{initial}\notin S$用于刻画查询提供的输入.函数$itype:S\to 2^{\mathcal{C}(\mathcal{T})}$把一个抽象服务映射到该Web服务的输入参数的类型集合;函数$otype:S\to\mathcal{C}(\mathcal{T})$指定了该Web服务的输出参数的类型.Web服务$service\in S$的描述逻辑编码$\Delta_{service}$为一个描述逻辑词汇公理集合,它由以下词汇公理构成:

$$\top\sqsubseteq\forall service.otype(service) \tag{9}$$

$$I_{service}\equiv\bigsqcap_{I\in itype(service)}\exists(\bigsqcup_{R\in\mathcal{R}(\mathcal{T})}R)^+.I \tag{10}$$

$$\exists service.\top\sqsubseteq I_{service} \tag{11}$$

$$I_{service}\sqsubseteq\exists service.\top \tag{12}$$

我们用$\Gamma_S=\cup_{service\in S}\Delta_{service}$来表示全部Web服务的描述逻辑编码所组成的并集.

2.3 从服务组合到描述逻辑推理问题的归约

对于给定的一组动作$\{a_1,a_2,a_3,\ldots,a_n\}$,初始状态$Initial$和目标状态$Final$,人工智能规划通过检验公式$Initial\wedge\langle a_1\vee a_2\vee a_3\vee\ldots\vee a_n\rangle Final$的可满足性来产生规划.服务组合的方法和上述做法是类似的,只不过组合服务的输入比规划的初始状态稍微复杂了些,因此,服务组合必须完成一系列的可满足性测试.

定义 2(组合服务的可满足性). 服务组合刻画了这样一个组合服务Q,它的n个输入参数的类型分别是$I_1,I_2,\ldots,I_n$,输出参数的类型为O.令$\mathcal{C}(\mathcal{T})$和$\mathcal{R}(\mathcal{T})$分别是本体$\mathcal{T}$中所有原子概念和原子角色的集合,$S\subseteq\mathcal{R}(\mathcal{T})$是所有抽象服务的集合,词汇公理集合$\Gamma_I$由一组关于保留角色$\boldsymbol{initial}\notin S$的词汇公理构成,即

$$\top\sqsubseteq\forall\boldsymbol{initial}.(\bigsqcup_i I_i) \tag{13}$$

$$I_Q\equiv\bigsqcap_i\exists\boldsymbol{initial}.I_i \tag{14}$$

$$\exists\boldsymbol{initial}.\top\sqsubseteq I_Q \tag{15}$$

设所有的词汇公理组成的集合为Γ,则组合服务Q的可满足性是一个"Γ包含"问题,即检验$\Gamma\models I_Q\sqsubseteq\exists(\bigsqcup_{R\in\mathcal{R}(\mathcal{T})}R)^+.O$是否成立.

不难看出,定义 2 的词汇公理集合Γ_I中的词汇公理和定义 1 中的式(9)~式(11)是非常相似的,只不过定义 1 中的词汇公理所刻画的是一个Web服务,而定义 2 的词汇公理所刻画的是保留角色***initial***;并且定义 2 中与定义 1 中的式(12)相对应的部分即是需要我们检验的

$$I_Q\sqsubseteq\exists(\bigsqcup_{R\in\mathcal{R}(\mathcal{T})}R)^+.O \tag{16}$$

注意,本来出现在$I_{service}$定义中的复杂角色$(\bigsqcup_{R\in\mathcal{R}(\mathcal{T})}R)^+$,并不出现在$I_Q$的定义中而出现在式(16)中.因为对于组合服务$Q$,它的输入被***initial***所限定,但是从这些输入产生输出的途径是多种多样的;对于Web服务$service$,它能够产生的输出被$service$所限定,但是获取输入的途径是多种多样的.

在式(16)中,I_Q利用***initial***取遍了所有合法的输入,以方便服务组合检验生成的组合服务是否能够为每一个合法的输入产生一个类型为O的输出.最后,必须指出的是,虽然词汇公理集合Γ_I限定了能够满足I_Q的解释,但是并不是所有的Γ_I模型中都能让我们产生兴趣(比如在某些解释之下,I_Q个体可以具有多于n个的***initial***属性值,也

© 中国科学院软件研究所 http://www.jos.org.cn

可以具有***initial***之外的属性.这些解释对服务组合来说都是无关紧要的,并且我们完全可以用$\boldsymbol{initial}_1$, $\boldsymbol{initial}_2$,…,$\boldsymbol{initial}_n$分别刻画类型为$I_1,I_2,\ldots,I_n$的输入以排除这些解释).

由于支持角色传递闭包的"I包含"问题是可判定的[9],而且上述的归约是有限过程,所以用描述逻辑进行服务组合问题是可判定的.

3 在语义 Web 中实现服务组合

上一节讨论了利用描述逻辑进行服务组合的基本思想和方法;在本节中,我们将在一种特殊的描述逻辑$\mathcal{SHOIQ}$中实现它.

采用描述逻辑$\mathcal{SHOIQ}$来实现语义 Web 服务组合的理由是多方面的:首先,$\mathcal{SHOIQ}$是 OWL 的理论基础,而且用于描述语义 Web 服务的 OWL-S 也是构建在 OWL 上的,采用$\mathcal{SHOIQ}$使得 Web 服务组合能够在一个统一的推理框架下实现;其次,$\mathcal{SHOIQ}$具有足够强大的表达能力,并且能够简洁地表述上一节中相关的词汇公理;最后,$\mathcal{SHOIQ}$有着有效的 Tableaux 推理算法和推理机的支持,利用$\mathcal{SHOIQ}$概念的可满足性判定算法进行服务组合非常方便.

3.1 利用$\mathcal{SHOIQ}$概念的可满足性判定算法进行服务组合

由定理 1 和定义 2 可得:组合服务 Q 是可满足的当且仅当下面的描述逻辑概念是不可满足的:

$$\forall(\sqcup_{R\in\mathcal{R}(\mathcal{T})}R)^*.(\sqcap_{A\sqsubseteq B\in I}\neg A\sqcup B)\sqcap I_Q\sqcap\forall(\sqcup_{R\in\mathcal{R}(\mathcal{T})}R)^+.\neg O \tag{17}$$

令保留角色***universal***是传递的,并且$\mathcal{R}(\mathcal{T})$中的任意一个角色都包含于***universal***中,下面我们把上文的相关定义和结论移植到描述逻辑$\mathcal{SHOIQ}$中.为了方便下面行文,我们把一个概念 C 中 M 的一个指定出现替换为 N 后的结果记为 $C[M\%N]$.

对于式(17),设I中的词汇公理$I_{service}\sqsubseteq\exists service.\top$和$\exists service.\top\sqsubseteq I_{service}$在$\sqcap_{A\sqsubseteq B\in I}\neg A\sqcup B$中的展开式分别为$F_{service}$和$E_{service}$,即

$$F_{service}\equiv\forall(\sqcup_{R\in\mathcal{R}(\mathcal{T})}R)^+.\neg I_1\sqcup\forall(\sqcup_{R\in\mathcal{R}(\mathcal{T})}R)^+.\neg I_2\sqcup\ldots\sqcup\forall(\sqcup_{R\in\mathcal{R}(\mathcal{T})}R)^+.\neg I_n\sqcup\exists service.\top \tag{18}$$

$$E_{service}\equiv\forall service.\bot\sqcup\exists(\sqcup_{R\in\mathcal{R}(\mathcal{T})}R)^+.I_1\sqcap\exists(\sqcup_{R\in\mathcal{R}(\mathcal{T})}R)^+.I_2\sqcap\ldots\sqcap\exists(\sqcup_{R\in\mathcal{R}(\mathcal{T})}R)^+.I_n \tag{19}$$

引理 1. 设$\boldsymbol{universal}\sqsubseteq U$,解释$I$满足$E_{service}[(\sqcup_{R\in\mathcal{R}(\mathcal{T})}R)^+\%U]$却不能满足$E_{service}$,那么,存在解释$I'$,使得解释$I$和$I'$的区别仅在于$service^{I'}\subset service^I$,并且$(E_{service}\sqcap F_{service})^{I'}=(E_{service}[(\sqcup_{R\in\mathcal{R}(\mathcal{T})}R)^+\%U])^I$.

证明:由于解释I不能满足$E_{service}$,所以$(\forall service.\bot)^I=\varnothing$,并且

$$(\exists(\sqcup_{R\in\mathcal{R}(\mathcal{T})}R)^+.I_1\sqcap\exists(\sqcup_{R\in\mathcal{R}(\mathcal{T})}R)^+.I_2\sqcap\ldots\sqcap\exists(\sqcup_{R\in\mathcal{R}(\mathcal{T})}R)^+.I_n)^I=\varnothing.$$

我们通过解释I来构造一个能够满足$\forall service.\bot$的解释I'.直观上,解释I'就是从$service^I$中移除所有满足$a\in(E_{service}[(\sqcup_{R\in\mathcal{R}(\mathcal{T})}R)^+\%U])^I$的元组$(a,b)$之后的$I$.显然,$E_{service}{}^{I'}=(E_{service}[(\sqcup_{R\in\mathcal{R}(\mathcal{T})}R)^+\%U])^I$.

另一方面,由于$(\sqcup_{R\in\mathcal{R}(\mathcal{T})}R)^{I'}\subset(\sqcup_{R\in\mathcal{R}(\mathcal{T})}R)^I$,所以,$(\forall(\sqcup_{R\in\mathcal{R}(\mathcal{T})}R)^+.\neg I_1\sqcup\forall(\sqcup_{R\in\mathcal{R}(\mathcal{T})}R)^+.\neg I_2\sqcup\ldots\sqcup\forall(\sqcup_{R\in\mathcal{R}(\mathcal{T})}R)^+.\neg I_n)^{I'}\supseteq(\forall(\sqcup_{R\in\mathcal{R}(\mathcal{T})}R)^+.\neg I_1\sqcup\forall(\sqcup_{R\in\mathcal{R}(\mathcal{T})}R)^+.\neg I_2\sqcup\ldots\sqcup\forall(\sqcup_{R\in\mathcal{R}(\mathcal{T})}R)^+.\neg I_n)^I=\Delta^I$,即得

$$(E_{service}\sqcap F_{service})^{I'}=(E_{service}[(\sqcup_{R\in\mathcal{R}(\mathcal{T})}R)^+\%U])^I\cap\Delta^I=(E_{service}[(\sqcup_{R\in\mathcal{R}(\mathcal{T})}R)^+\%U])^I.\quad\square$$

引理 1 说明,把定义 1 和定义 2 中的复杂角色$(\sqcup_{R\in\mathcal{R}(\mathcal{T})}R)^+$替换为一个包含***universal***的角色U,不会对服务组合的结果产生影响.

引理 2. $I_Q\sqcap\forall(\sqcup_{R\in\mathcal{R}(\mathcal{T})}R)^*.(\neg I_Q\sqcup\forall(\sqcup_{R\in\mathcal{R}(\mathcal{T})}R)^+.\neg O)\equiv I_Q\sqcap\forall(\sqcup_{R\in\mathcal{R}(\mathcal{T})}R)^+.\neg O$.

证明:因为

$$\forall(\sqcup_{R\in\mathcal{R}(\mathcal{T})}R)^+.\neg O\sqsubseteq\forall(\sqcup_{R\in\mathcal{R}(\mathcal{T})}R)^+.\forall(\sqcup_{R\in\mathcal{R}(\mathcal{T})}R)^+.\neg O,$$

$$\forall(\sqcup_{R\in\mathcal{R}(\mathcal{T})}R)^+.\forall(\sqcup_{R\in\mathcal{R}(\mathcal{T})}R)^+.\neg O\sqsubseteq\forall(\sqcup_{R\in\mathcal{R}(\mathcal{T})}R)^+.(\neg I_Q\sqcup\forall(\sqcup_{R\in\mathcal{R}(\mathcal{T})}R)^+.\neg O),$$

所以,

$$\forall(\sqcup_{R\in\mathcal{R}(\mathcal{T})}R)^+.\neg O\sqcap\forall(\sqcup_{R\in\mathcal{R}(\mathcal{T})}R)^+.(\neg I_Q\sqcup\forall(\sqcup_{R\in\mathcal{R}(\mathcal{T})}R)^+.\neg O)\equiv\forall(\sqcup_{R\in\mathcal{R}(\mathcal{T})}R)^+.\neg O.$$

又因$I_Q\sqcap(\neg I_Q\sqcup\forall(\sqcup_{R\in\mathcal{R}(\mathcal{T})}R)^+.\neg O)\equiv I_Q\sqcap\forall(\sqcup_{R\in\mathcal{R}(\mathcal{T})}R)^+.\neg O$,故引理成立. $\square$

© 中国科学院软件研究所 http://www.jos.org.cn

引理2说明,把定义2中的结论$I_Q \sqsubseteq \exists(\bigsqcup_{R\in\mathcal{R}(\mathcal{T})}R)^+.O$改写为前提$I_Q \sqsubseteq \forall(\bigsqcup_{R\in\mathcal{R}(\mathcal{T})}R)^+.\neg O$,不会对服务组合的结果产生影响.

定理 3. 设$\boldsymbol{universal} \sqsubseteq U$,令$G\equiv\forall(\bigsqcup_{R\in\mathcal{R}(\mathcal{T})}R)^*.(\bigsqcap_{A\sqsubseteq_{\mathcal{T}}B\in I}(\neg A\sqcup B)\sqcap(\neg I_Q\sqcup\forall(\bigsqcup_{R\in\mathcal{R}(\mathcal{T})}R)^+.\neg O))\sqcap I_Q$,则组合服务$Q$是可满足的当且仅当概念$G[(\bigsqcup_{R\in\mathcal{R}(\mathcal{T})}R)^+\%U]$是不可满足的.

证明:由定理 2 的证明我们可以同样得到:如果概念G是可满足的,那么概念$G[(\bigsqcup_{R\in\mathcal{R}(\mathcal{T})}R)^+\%U]$也是可满足的.而且由引理 1 我们知道:如果概念$G[(\bigsqcup_{R\in\mathcal{R}(\mathcal{T})}R)^+\%U]$是可满足的,那么概念$G$也是可满足的.也就是说,概念$G[(\bigsqcup_{R\in\mathcal{R}(\mathcal{T})}R)^+\%U]$是不可满足的当且仅当概念$G$是不可满足的.由引理 2 可知,概念$G$是不可满足的当且仅当式(17)是不可满足,因此定理成立. □

根据定理3,我们有定义3和定义4:

定义 3(Web服务的$\mathcal{SHOIQ}$编码). 令$\mathcal{R}(\mathcal{T})$为领域本体$\mathcal{T}$中所有原子角色的集合,角色$\boldsymbol{universal}\notin\mathcal{R}(\mathcal{T})$. ***universal***是传递的,并且***universal***与$\mathcal{R}(\mathcal{T})$中的任意一个角色都构成包含关系.Web服务S有n个输入参数$inarg_1$,$inarg_2$,$inarg_3$,…,$inarg_n$,它们的类型分别为$I_1,I_2,\ldots,I_n$,Web服务S的输出$outarg$的类型为O.Web服务S的$\mathcal{SHOIQ}$编码Δ_S为一个$\mathcal{SHOIQ}$公理集合,由以下公理构成:

$$outarg \sqsubseteq \boldsymbol{universal};$$
$$\top \sqsubseteq \forall outarg.O;$$
$$\exists outarg.\top \sqsubseteq I_S;$$
$$I_S \sqsubseteq \exists outarg.\top;$$
$$I_S \equiv \bigsqcap_i \exists inarg_i.I_i;$$
$$\boldsymbol{universal} \sqsubseteq inarg_1;$$
$$\boldsymbol{universal} \sqsubseteq inarg_2;$$
$$\ldots$$
$$\ldots$$
$$\boldsymbol{universal} \sqsubseteq inarg_n.$$

我们用Γ_{SH}来表示全部Web服务的$\mathcal{SHOIQ}$编码所组成的并集.

定义 4(组合服务的可满足性在$\mathcal{SHOIQ}$中的归约). 服务组合刻画了这样一个组合服务Q,它的n个输入参数$inarg_1$,$inarg_2$,$inarg_3$,…,$inarg_n$的类型分别是$I_1,I_2,\ldots,I_n$,它的输出$outarg$的类型为O.令$\mathcal{R}(\mathcal{T})$为领域本体$\mathcal{T}$中所有原子角色的集合,角色$\boldsymbol{universal}\notin\mathcal{R}(\mathcal{T})$.***universal***是传递的,并且***universal***与$\mathcal{R}(\mathcal{T})$中的任意一个角色都构成包含关系.$\mathcal{SHOIQ}$公理集合Γ_{DH}由以下公理构成:

$$\top \sqsubseteq \forall inarg_1.I_1;$$
$$\top \sqsubseteq \forall inarg_2.I_2;$$
$$\ldots$$
$$\top \sqsubseteq \forall inarg_n.I_n;$$
$$I_Q \equiv \bigsqcap_i \exists inarg_i.I_i;$$
$$\exists inarg_1.\top \sqsubseteq I_Q;$$
$$\exists inarg_2.\top \sqsubseteq I_Q;$$
$$\ldots$$
$$\exists inarg_n.\top \sqsubseteq I_Q;$$
$$I_Q \sqsubseteq \forall outarg.\neg O;$$
$$inarg_1 \sqsubseteq \boldsymbol{universal};$$
$$inarg_2 \sqsubseteq \boldsymbol{universal};$$
$$\ldots$$
$$inarg_n \sqsubseteq \boldsymbol{universal};$$

© 中国科学院软件研究所 http://www.jos.org.cn

$$\textbf{\textit{universal}} \sqsubseteq outarg.$$

设$\Gamma_{\mathcal{H}}=\Gamma_{\mathcal{RH}}\cup\Gamma_{\mathcal{SH}}$,则组合服务$Q$的可满足性在$\mathcal{SHOIQ}$中的归约是一个$\mathcal{SHOIQ}$概念的可满足性问题,即检验概念$I_Q$是否关于$\Gamma_{\mathcal{H}}$不可满足.

由于$\mathcal{SHOIQ}$概念的可满足性问题是可判定的[10],而且上述的归约是有限过程,所以用描述逻辑进行服务组合问题是可判定的.

3.2 描述逻辑的Tableaux算法

由于描述逻辑推理的演绎系统过于复杂,描述逻辑的推理通常由Tableaux实现.Tableaux算法能够解决描述逻辑概念的可满足性问题.并且,由于描述逻辑中的推理问题可以规约为描述逻辑概念的可满足性问题(定理 1和定理 2),Tableaux实际上给出了一种描述逻辑推理的方法.当然,描述逻辑推理的方法不只有Tableaux,还有其他方法,但Tableaux方法是比较行之有效的,也是众多描述逻辑推理机采用的方法,例如pellet[15]和fact[16]等.

文献[10]提出了一种检验 $\mathcal{SHOIQ}$ 概念可满足性的 Tableaux 算法,该算法通过产生一个 Tableaux 来检验$\mathcal{SHOIQ}$概念的可满足性.如果一个 $\mathcal{SHOIQ}$概念 C 产生的 Tableaux 不包含存在冲突的分支,那么该概念是可满足的.也就是说,该 Tableaux 实际上给出了一个能够满足概念 C 的解释.由于 $\mathcal{SHOIQ}$的 Tableaux 算法使用了一些非常复杂的技巧,我们只介绍它的一个简化版本,即 $\mathcal{SH}$ 的 Tableaux 算法.

设所有概念和所有角色的集合分别为$\mathcal{NC}$和$\mathcal{NR}$,概念C产生的Tableaux是一个符号有向图$G_C=\langle V,E,\mathcal{L}\rangle$,其中,函数$\mathcal{L}:V\cup E\to 2^{\mathcal{NC}\cup\mathcal{NR}}$把图中的一个顶点映射到一组概念,并且把图中的一条边映射到一个角色.图中的一个分支是$B_{x:A}$冲突的,如果该分支中存在这样的一个顶点x,使得$\mathcal{L}(x)$同时包含一个原子概念A及其否定$\neg A$,记为$\langle P_{x:A},P_{x:\neg A}\rangle$,其中,$P_{x:A}$为产生$A$的符号路径,$P_{x:\neg A}$为产生$\neg A$的符号路径.

设$\sqsubseteq^*$是包含关系$\sqsubseteq$的自反传递闭包,记集合$\{e\in E|\mathcal{L}(e)\sqsubseteq^* R\}$为 $E(R)$,概念 A 的否定内置范式(negation normal form)为 $nnf(A)$,下面是构造一个 Tableaux 的规则,其中,每个概念的标注[$path$]记录了产生该概念的符号路径:

展开规则.如果A是原子概念并且$A\sqsubseteq B,A^{[path]}\in\mathcal{L}(x),nnf(B)\notin\mathcal{L}(x)$,则$\mathcal{L}(x)=\mathcal{L}(x)\cup\{nnf(B)^{[path:A]}\}$.

分支规则$\sqcup$.如果$A\sqcup B^{[path]}\in\mathcal{L}(x)$,并且$nnf(A)\notin\mathcal{L}(x),nnf(B)\notin\mathcal{L}(x)$,则产生$G$的一个副本$G'$,$\mathcal{L}(x)=\mathcal{L}(x)\cup\{nnf(A)^{[path]}\},\mathcal{L}(x')=\mathcal{L}(x')\cup\{nnf(B)^{[path]}\},G=G\cup G'$.

扩充规则$\exists$.如果$\exists R.A^{[path]}\in\mathcal{L}(x)$,并且不存在顶点$y\in V$使得$\langle x,y\rangle\in E(R)$并且$A\in\mathcal{L}(y)$,则$V=V\cup\{y\},E=E\cup\{\langle x,y\rangle\},\mathcal{L}(y)=\{A^{[path.R]}\},\mathcal{L}(\langle x,y\rangle)=R$.

约束规则$\sqcap$.如果$A\sqcap B^{[path]}\in\mathcal{L}(x)$,但$nnf(A)\notin\mathcal{L}(x)$或$nnf(B)\notin\mathcal{L}(x)$,则$\mathcal{L}(x)=\mathcal{L}(x)\cup\{nnf(A)^{[path]},nnf(B)^{[path]}\}$.

约束规则$\forall$.如果$\forall R.A^{[path]}\in\mathcal{L}(x)$,并且存在边$\langle x,y\rangle\in E(R)$满足$A\notin\mathcal{L}(y)$,则$\mathcal{L}(y)=\mathcal{L}(y)\cup\{A^{[path.R]}\}$.

约束规则$\forall+$.如果$R\sqsubseteq^* S$,并且R是传递的,$\forall S.A^{[path]}\in\mathcal{L}(x)$,存在边$\langle x,y\rangle\in E(R)$满足$\forall R.A\notin\mathcal{L}(y)$,则$\mathcal{L}(y)=\mathcal{L}(y)\cup\{\forall R.A^{[path.S]}\}$.

显然,展开规则实际上是分支规则$\sqcup$在A是原子概念情况下的一个加速,因为当A是原子概念时,分支规则$\sqcup$将迅速使 G 成为一个冲突分支,从而有 $G=G'$.正是上述原因,展开规则必须在分支规则之前应用.另外,展开规则标明了符号路径 $path$ 产生的原子概念 A.

给定一个概念C,Tableaux算法首先把符号路径$path$置为一个空串ε,并把图G_C初始化为$\langle\{o\},\varnothing,\mathcal{L}\rangle$,其中,$\mathcal{L}(o)=nnf(C)^{[\varepsilon]}$,然后对$G_C$的每个节点应用上述规则.最后检查$G_C$是否存在非冲突的分支,以确定概念$C$是否可满足.

3.3 一个例子

下面我们以清单 1 中的Web服务为基础,举例说明这个服务组合方法的过程和步骤.假设服务组合描述了这样一个组合服务Q,它使用参数$initial$接受一个类型为$I_1\sqcup I_2$的输入,并且通过参数$result$提供一个类型为$A\sqcup B$的输出.根据定义 3,我们可以从清单 1 中直接得到Web服务S_1的编码 Δ_{S_1},即

$$\{S_1\equiv\exists\, in_{S_1}.I_1,\top\sqsubseteq\forall\, out_{S_1}.A,S_1\sqsubseteq\exists\, out_{S_1}.\top,\exists\, out_{S_1}.\top\sqsubseteq S_1,out_{S_1}\sqsubseteq\textbf{\textit{universal}},\textbf{\textit{universal}}\sqsubseteq in_{S_1}\}.$$

© 中国科学院软件研究所 http://www.jos.org.cn

类似地,我们可以得到Web服务S_2的编码Δ_{S_2}.根据定义 4,我们得到一个刻画组合服务Q的词汇公理集合Γ_{IM},即

$\{Q\equiv\exists initial.(I_1\sqcup I_2),\top\sqsubseteq\forall initial.(I_1\sqcup I_2),\exists initial.\top\sqsubseteq Q,Q\sqsubseteq\forall result.(\neg A\sqcap\neg B),initial\sqsubseteq \textbf{\textit{universal}},\textbf{\textit{universal}}\sqsubseteq result\}$.

然后,我们使用 Tableaux 算法来判定概念 Q 是不可满足的.对 Web 服务组合来说,Tableaux 产生的一个冲突记录实际上是一个输入输出参数的绑定.令G_Q为$\Delta_{S_1}\cup\Delta_{S_2}\cup\Gamma_{IM}$的展开式,清单 2 给出了产生用于检验概念$Q$可满足性的 Tableaux 的过程.

清单 2. 概念 Q 的 Tableaux 的产生过程.

1. Tableaux选择G_Q中一个分支$\exists\ out_{S_2}.\top\sqcap\exists\ in_{S_2}.I_2$却导致$S_2$的$out_{S_2}$属性产生的个体与$Q$的$\forall result.(\neg A\sqcap\neg B)$产生冲突,记录冲突$\langle:S_2.out_{S_2},:Q.result\rangle$.
2. Tableaux选择G_Q中一个分支$\exists\ out_{S_1}.\top\sqcap\exists\ in_{S_1}.I_1$却导致$S_1$的$out_{S_1}$属性产生的个体与$Q$的$\forall result.(\neg A\sqcap\neg B)$产生冲突,记录冲突$\langle:S_1.out_{S_1},:Q.result\rangle$.
3. Tableaux选择G_Q中一个分支$\exists initial.I_1$却导致Q的$initial$属性产生的个体与S_1的$\forall\ in_{S_1}.\neg I_1$产生冲突,记录冲突$\langle:Q.initial,:S_1.in_{S_1}\rangle$.
4. Tableaux选择G_Q中一个分支$\exists initial.I_2$却导致Q的$initial$属性产生的个体与S_2的$\forall\ in_{S_2}.\neg I_2$产生冲突,记录冲突$\langle:Q.initial,:S_2.in_{S_2}\rangle$.

4 相关工作

文献[17]根据类型之间的上下位关系提出了基于输入输出参数的服务匹配算法,文献[18]进一步发掘了类型之间的重叠关系,提出了部分匹配的算法;文献[8]针对 Web 服务数量巨大这一特点,在文献[18]的基础上设计了基于表格(table-base)的高效匹配算法以及前向搜索的服务组合算法.文献[7]借鉴了自动化程序综合的思想,在文献[18]的基础上提出了采用线性逻辑推理进行 Web 服务组合的方法.文献[7]中的方案还能够进行 Web 服务参数个数的匹配,但是,该方案并不能保证服务组合算法会终止.

无论是深度搜索的方法还是定理证明的方法,上述服务组合方案所利用类型之间的关系十分有限,即等价、包含和相交 3 种关系.所以,这些服务组合方案通常难以满足用户的要求.比如,你需要一些汽油(oil),但是所有的 Web 服务都只能提供桶装汽油(barreled oil),那么,上述的服务组合方案就会拒绝你的要求.如图 1 所示,因为桶装汽油不是一种汽油,它是一种由汽油和油桶(barrel)构成的复合体(complex).本文的服务组合方案便能够很好地解决这个问题,它会告诉用户把桶装汽油买回来,里面装的就是汽油.

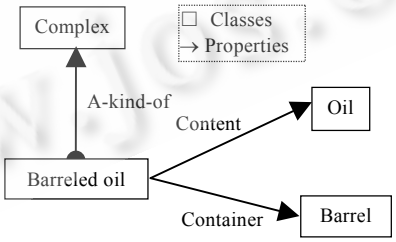

Fig.1 Ontology describing barreled oil

图 1 描述桶装汽油的本体

文献[2−4]利用人工智能规划算法进行服务组合,但这些工作所依赖的情景演算并不十分适合 Web 服务.因为 Web 服务通常要产生输出以返回结果,而情景演算则假设规划中不会产生新个体.所以,这些服务组合方案只能应用于那些执行过程中没有新对象产生的 Web 服务.相比之下,本文的服务组合方案便没有这样的局限性.

由于语义 Web 服务组合的方法依赖于形式化语言,因此,形式化语言的表达能力和计算复杂度就决定了这些语义 Web 服务组合方法的能力和计算复杂度.在特定应用和环境下,某些形式语言要比其他形式语言更为适

© 中国科学院软件研究所 http://www.jos.org.cn

合.然而,没有哪种形式语言具有绝对的优越性,因为形式语言表达能力的增强总意味着计算复杂度的提高.图 2 总结了上述各种语义 Web 服务组合方法的适用性.由于语义 Web 服务组合是在语义 Web 的环境中进行的,而描述逻辑是语义 Web 的基础,所以用描述逻辑进行语义 Web 服务是非常适合的.

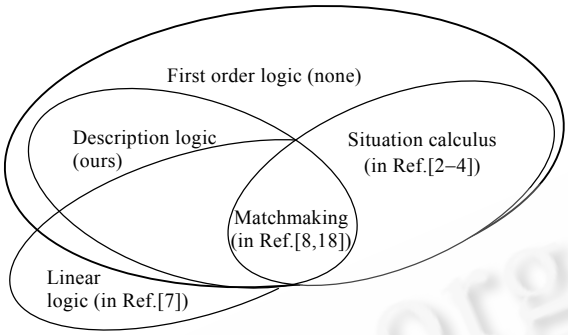

Fig.2 Applicability of approaches to service composition

图 2 服务组合方法的适用性

5 结论和展望

本文通过动态逻辑和描述逻辑之间的对比研究,采用描述逻辑来刻画 Web 服务的输入输出参数类型以及 Web 服务执行的前提条件和结果,扩展了基于动态逻辑的人工智能规划方法,提出了把语义 Web 服务组合问题转化为描述逻辑推理问题的方法.与基于动态逻辑的人工智能规划方法相比,刻画 Web 服务的描述逻辑公理通过在它们的前提中引入角色来解决多个 Web 服务输入参数的问题,通过角色的传递性来解决类型的可获取性问题,并通过角色的包含关系来完成 Web 服务输入输出的绑定.

角色的传递性可以通过角色传递闭包来实现,但是,角色的传递闭包并不为语义 Web 所采用的描述逻辑所支持,因此,我们只能依靠一个传递角色来实现它.通过公理把传递闭包中的角色包含于这个传递角色中,我们同样解决了类型的可获取性问题,但是,这样做的代价是能够满足这些公理的模型增多了,因此,我们不能简单地检验是否存在满足这些公理的模型来回答 Web 服务组合的目标能否达成的问题.只有证明满足这些公理的每个模型都能够在从提供的输入中产生用户需要的输出,我们才能回答 Web 服务组合的目标能够达成的问题.定理 3 告诉我们,在本文的 Web 服务组合方法中,传递角色确实能够代替角色的传递闭包而不损害这些公理的逻辑结果.

在我们的语义 Web 服务组合方法中,现实的语义 Web 服务总是先被抽象成若干个抽象服务,并用描述逻辑公理对这些抽象服务加以刻画,然后把这些抽象服务当作现实的语义 Web 服务的替身参与到 Web 服务组合中去.同时,用户需求的组合服务同样也为一些描述逻辑概念和公理所刻画.这样,我们就能够通过描述逻辑推理来告诉用户需求的组合服务能否达成(如果能够达成,推理的过程当然就是服务组合的过程).

我们的方案能够保证在基本层次上进行服务组合的算法是终止的,并且在超越了基本层次的服务组合方案中保证算法的可终止性,只要描述服务执行的前提条件和结果的形式语言与 OWL 结合后的语言是可判定的.然而,至于使用何种形式语言来描述服务执行的前提条件和结果,目前还没有一个能够获得广泛认可的解决方案,其主要障碍在于如何把本体整合到一个基于规则的演绎系统中去.

任何一种服务组合方法都需要Web服务的描述语言提供相应的支持,同时,服务描述语言的表达能力和描述风格也会对建立于其上的服务组合方法的性能乃至功能产生影响.现有的语义Web服务描述框架如OWL-S+SWRL/KIF[19],WSML(Web service modeling language)+WSMO(Web service modeling ontology)[20,21]和SWSL(semantic Web service language)+SWSO(semantic Web service ontology)[22,23]都不是为本文的服务组合方案所设计的,对现有的语义Web服务描述框架进行改进也是一项非常有意义的工作.

如果存在多个组合服务同时满足查询的需要,那么我们则面临挑选一个最好的组合服务的问题.我们可以

© 中国科学院软件研究所 http://www.jos.org.cn

在文献[10]的 Tableaux 算法中使用一些优选策略来选取我们认为最好的 Web 服务,也可以利用模糊描述逻辑来对 Web 服务进行分级.

文献[6]提出了利用自动机进行面向 Web 服务行为的服务组合(服务交响)方案,而且该方案的实现也同样利用了用于判定 $\mathcal{SHOIQ}$ 概念可满足性的 Tableaux 算法,所以,我们可以把文献[6]中服务交响的方法和本文的服务组合方法整合在一起,为服务组合和服务交响的自动化提供一个统一的平台.

References:

[1] McGuinness DL, van Harmelen F. OWL Web ontology language overview. World Wide Web Consortium (W3C) Recommendation. 2004. http://www.w3.org/TR/owl-features/

[2] McIlraith SA, Son TC. Adapting golog for composition of semantic Web services. In: Fensel D, Giunchiglia F, McGuinness DL, Williams MA, eds. Proc. of the 8th Int'l Conf. on Principles and Knowledge Representation and Reasoning (KR 2002). Toulouse: Morgan Kaufmann Publishers, 2002. 482−496.

[3] Sirin E, Parsia B, Wu D, Hendler JA, Nau DS. HTN planning for Web service composition using SHOP2. Journal of Web Semantics, 2004,1(4):377−396.

[4] Baader F, Lutz C, Milicic M, Sattler U, Wolter F. A description logic based approach to reasoning about Web services. In: Vasiliu L, ed. Proc. of the WWW 2005 Workshop on Web Service Semantics: Towards Dynamic Business Integration. Chiba: ACM Press, 2005. 636−647.

[5] Pistore M, Traverso P, Bertoli P, Marconi A. Automated synthesis of executable Web service compositions from BPEL4WS processes. In: Ellis A, Hagino T, eds. Proc. of the 14th Int'l World Wide Web Conf. (WWW 2005) Special Interest Tracks and Posters. New York: ACM Press, 2005. 1186−1187.

[6] Berardi D, Calvanese D, De Giacomo G, Lenzerini M, Mecella M. Automatic service composition based on behavioral descriptions. Int'l Journal of Cooperative Information Systems, 2005,14(4):333−376.

[7] Rao JH, Küngas P, Matskin M. Logic-Based Web services composition: From service description to process model. In: Zhang LJ, ed. Proc. of the IEEE Int'l Conf. on Web Services (ICWS 2004). San Diego: IEEE Computer Society, 2004. 446−453.

[8] Constantinescu I, Faltings B, Binder W. Large scale, type-compatible service composition. In: Proc. of the IEEE Int'l Conf. on Web Services (ICWS 2004). San Diego: IEEE Computer Society, 2004. 506−513. http://csdl2.computer.org/persagen/DLAbsToc.jsp?resourcePath=/dl/proceedings/icws/&toc=comp/proceedings/icws/2004/2167/00/2167toc.xml&DOI=10.1109/ICWS.2004.1314776

[9] Calvanese D, De Giacomo G, Lenzerini M, Nardi D. Reasoning in expressive description logics. In: Robinson A, Voronkov A, eds. Handbook of Automated Reasoning. Amsterdam: Elsevier Science Publishers, 2001. 1581−1634.

[10] Horrocks I, Sattler U, Tobies S. Practical reasoning for expressive description logics. In: Ganzinger H, McAllester D, Voronkov A, eds. Proc. of the LPAR'99. LNCS 1705, Heidelberg: Springer-Verlag, 1999. 161−180.

[11] De Rijke M. Modal logics and description logics. In: Franconi E, De Giacomo G, MacGregor RM, Nutt W, Welty CA, eds. Proc. of the '98 Int'l Workshop on Description Logics (DL'98). CEUR-WS.org, 1998. 1−3. http://citeseer.ist.psu.edu/derijke98modal.html

[12] Shi ZZ, Dong MK, Jiang YC, Zhang HJ. A logical foundation for the semantic Web. Science in China (Series E), 2005,48(2): 161−178 (in Chinese with English abstract).

[13] Martin D, Burstein M, Denker G, Elenius D, McDermott D, McGuinness D, McIlraith S, Paolucci M, Parsia B, Payne T, Sirin E, Srinivasan N, Sycara K. OWL-S: Semantic markup for Web services. DAML, 2006. http://www.ai.sri.com/daml/services/owl-s/1.2/

[14] Badea L. Planning in description logics: Deduction versus satisfiability testing. In: Prade H, ed. Proc. of the 13th European Conf. on Artificial Intelligence. Chichester: John Wiley and Sons, 1998. 479−483.

[15] Pellet. 2003. http://www.mindswap.org/2003/pellet/

[16] FaCT. 2003. http://www.cs.man.ac.uk/~horrocks/FaCT/

[17] Paolucci M, Kawamura T, Payne TR, Sycara KP. Semantic matching of Web services capabilities. In: Horrocks I, Hendler JA, eds. Proc. of the 1st Int'l Semantic Web Conf. on the Semantic Web. LNCS 2342, Heidelberg: Springer-Verlag, 2002. 333−347.

[18] Li L, Horrocks I. A software framework for matchmaking based on semantic Web technology. Int'l Journal of Electronic Commerce, 2004,8(4):39−60.

[19] Horrocks I, Patel-Schneider PF, Boley H, Tabet S, Grosof B, Dean M. SWRL: A semantic Web rule language combining OWL and RuleML. DAML, 2003. http://www.daml.org/2003/11/swrl/

[20] De Bruijn J, Fensel D, Keller U, Kifer M, Lausen H, Krummenacher R, Polleres A, Predoiu L. Web service modeling language (WSML). 2005. http://www.w3.org/Submission/WSML/

© 中国科学院软件研究所 http://www.jos.org.cn

[21] De Bruijn J, Bussler C, Domingue J, Fensel D, Hepp M, Keller U, Kifer M, Konig-Ries B, Kopecky J, Lara R, Lausen H, Oren E, Polleres A, Roman D, Scicluna J, Stollberg M. Web service modeling ontology (WSMO). 2005. http://www.w3.org/Submission/WSMO/

[22] Battle S, Bernstein A, Boley H, Grosof B, Gruninger M, Hull R, Kifer M, Martin D, McIlraith S, McGuinness D, Su JW, Tabet S. Semantic Web services language. 2005. http://www.w3.org/Submission/SWSF-SWSL/

[23] Battle S, Bernstein A, Boley H, Grosof B, Gruninger M, Hull R, Kifer M, Martin D, McIlraith S, McGuinness D, Su JW, Tabet S. Semantic Web services ontology. 2005. http://www.w3.org/Submission/SWSF-SWSO/

附中文参考文献:

[12] 史忠植,董明楷,蒋运承,张海俊.语义 Web 的逻辑基础.中国科学(E 辑),2004,34(10):1123−1138.

王杰生(1981－),男,广东普宁人,助理工程师,主要研究领域为语义网,形式化方法与技术.

李梦君(1975－),男,博士,讲师,主要研究领域为形式化方法与技术,信息安全技术.

李舟军(1963－),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为高可信软件技术,语义网,Web 服务与 P2P 计算,数据挖掘与生物信息学.

Call for Papers for the 2008 IEEE/IFIP International Conference on Embedded and Ubiquitous Computing (EUC 2008)

The EUC 2008 conference provides a forum for engineers and scientists in academia, industry, and government to address all resulting profound challenges including technical, safety, social, legal, political, and economic issues, and to present and discuss their ideas, results, work in progress and experience on all aspects of embedded and ubiquitous computing. Topics of particular interest include, but are not limited to:

Embedded Computing Track: Embedded System Software & Optimization; HW/SW Co-Design & Design Automation; Cyber-Physical Systems; Real-Time & Operating Systems; Application-Specific Processors and Devices; Power-Aware Computing; Sensor Networks; System/Network-on-Chip; Reconfigurable Computing Applications; Others and emerging new topics.

Ubiquitous Computing Track: Pervasive Computing & Communications; Middleware and Peer-to-Peer Computing; Internet Computing and Applications; Multimedia and Data Management; Human-Computer Interaction; Network Protocols; Wireless Communication & Networks; Mobile Computing; Agents and Distributed Computing; Security and Fault Tolerance Applications.

Submission Information

Submissions should include abstract, 5-10 keywords, and the e-mail address of the corresponding author and be in PDF format. Each submission must not exceed 8 pages in the IEEE 8.5 " × 11" two-column format with 10-12 point font (Template: **DOC** or **LaTeX**), including tables and figures. Each submission should be regarded as an undertaking that, should the submission be accepted, at least one of the authors must attend the conference to present the work in order that the accepted paper can be put into digital library. The final version of an accepted paper will be limited to 8 pages in IEEE proceeding format.

Important Dates

Workshop Proposal: 1 April 2008
Paper submission due: 30 May 2008
Acceptance notification: 18 August 2008
Camera-Ready due: 22 September 2008
Author registration: 22 September 2008
Conference: 17~20 December 2008

Publications

The accepted papers have to be presented orally at the conference and will be published in proceedings of the EUC 2008 conference by IEEE Computer Society. The selected best papers will be published in special issues of journals, including International Journal of Parallel, Emergent and Distributed Systems and Journal of Embedded Computing.

© 中国科学院软件研究所 http://www.jos.org.cn