

基于推理信息的本体模块化方法^{*}

方 俊 郭 雷 王晓东

(西北工业大学自动化学院 西安 710072)

摘 要 本体模块化在本体推理和复用等应用中有着极为重要的作用。怎么样将本体划分成小的模块是最基本的问题,目前本体模块化的工作主要集中在本体复用的目的上。在这篇文章中,我们提出了一种基于推理信息的本体模块化方法,该方法以提高推理的性能为目的。在基于同一最小推理集合内的公理之间内聚性将会增强的合理假设下,我们的模块化方法通过分析每次推理的过程,得到推理的最小推理集合,然后增强最小推理集合内公理¹之间的内聚度,最后根据内聚度将本体划分成模块。在评估阶段,我们首先使用训练公理划分本体,然后通过测试公理来调查本体推理性能提高的程度。根据训练公理和测试公理所属范围的不同,我们使用了三组实验:训练公理和测试公理限定在同一较小范围,训练公理和测试公理不限定范围,训练公理和测试公理的范围实现三次不同的改变。实验的结果尤其是符合实际应用情况的实验三的结果证明了基于推理信息的本体模块化方法的有效性。

关键词 本体,模块化,推理,公理

Reasoning Information Based Ontology Modularization

FANG Jun GUO Lei WANG Xiao-dong

(College of Automation, Northwestern Polytechnical University, Xi'an 710072, China)

Abstract There is a growing need for applying the principle of modularity to representations of ontological knowledge in order to facilitate the scalability and re-use of ontology. How to partition ontology automatically is the primary issue, current research work has focused on re-use of ontology. In the paper, a reasoning information based ontology modularization method is presented for the purpose of enhancing the reasoning performance in ontology. According to the assumption that axioms which are in the same Minimal Reasoning Set have high cohesion, our method firstly calculates Minimal Reasoning Set of every reasoning task by analyzing the process of reasoning, then cohesion between axioms in the same Minimal Reasoning Set is increased; finally ontology is partitioned into modules according to cohesion. In the evaluation stage, training axioms are used to modularize ontology firstly, then some test axioms are carried to investigate the enhancement of reasoning performance. Three experiments are used to evaluate our ontology modularization approach, the results, especially the result of experiment three which is similar to practical situation prove the effectiveness of the reasoning information based ontology modularization method.

Keywords Ontology, Modularization, Reasoning, Axioms

1 引言

本体在信息处理领域有着非常重要的作用,特别是随着语义网^[1]的发展,越来越多的本体被创建。在某些复杂的领域,本体的体积非常巨大,它包含了成千上万的概念和关系。这些大体积的本体给推理的效率和本体的复用带来了很严重的问题,很多的研究表明本体模块化是很好的解决方法:

1)推理效率:推理的效率在本体研究领域是一个非常重要的问题,因为本体的验证和信息的检索都是以它为基础的。随着本体的增大,推理的效率下降得非常快。如果我们能够将本体划分为很多小的模块,使每一次推理都限制在一个或少数几个模块内,那么推理的效率将会有很大的提升。

2)本体复用:本体复用能够节省资源,提高本体建立的效率,通常的情况是只需要复用本体的一部份。使用本体模块化,我们能够将一个很大的本体分成很多小的模块,这样就只

用根据需要复用部分的模块,从而提高了复用的效率和灵活性。

在文献[2]中,本体模块被定义:一个整体中有意义的一部份。在本体模块的研究领域,根据什么样的标准来划分本体是最为首要的问题。在文献[3]中,为了方便可视化和浏览本体,一种基于本体内部结构的模块化方法被提出来。该方法通过使用本体的结构信息计算概念之间的相关度,然后来进行模块化的划分。文献[4-6]都是为了复用的目的,给定概念和关系的集合,基于本体的逻辑结构,从一个大的本体中提取出相关的模块。所不同的是文献[6]考虑了语义上的特性,并保证了模块的完整性和正确性。

以上的工作都是基于浏览或复用的目的来模块化本体,跟这些工作不同,我们的模块化方法基于本体的推理信息,其目的是提高推理的效率。基于推理信息的模块化方法是基于下面的假设。

^{*}国家自然科学基金资助项目(60675015)资助。方 俊 博士生,主要从事语义网和本体的研究;郭 雷 博士生导师,主要从事神经网络、模式识别和知识管理等;王晓东 博士生,主要从事语义网和智能检索。

¹本体由公理组成(axioms),公理定义了概念(concepts)和作用(roles)关系的描述。

假设:在同一最小推理集合内公理间内聚性将增强。

这里,最小的推理集合是为了进行一次推理必须要涉及到的公理集合。在同一最小推理集合内的公理内聚性越强,则它们就越有可能被划分在同一个模块内,这样就有可能将推理限制在一个或少数几个模块内,从而提高推理的效率。相对于现在的模块化方法,基于推理信息的模块化方法的另一个特点就是考虑到模块划分的动态性,即模块的组织是动态的,是随着用户对本体的使用不停地变化的。这个特点使得基于推理信息的模块化方法有很好的适应性和灵活性,有着更好的应用价值。

本文的工作是要设计一种有效的本体模块化方法。本文第2节简要介绍这篇文章本体表示语言的基础描述逻辑;第3节详细描述基于推理信息的本体模块化方法;第4节介绍了证明我们算法的有效性的实验;最后是结论和未来的工作。

2 基于推理信息的本体模块化方法

基于本体的大部分应用都建立在推理的基础之上,在对本体进行模块化的划分时,也应该充分地利用这些信息。我们用基于推理信息的本体模块化方法分析每一次推理的过程,求解出最小的推理集合,增加该集合中公理的内聚性,最后根据内聚性来模块化本体。本文处理的对象是OWL的基础描述逻辑(Description Logics)^[7]。

2.1 模块化的步骤

本体是由公理组成的。每一对公理之间有内聚度,我们采用函数 $f(\alpha, \beta)$ 来表示公理 α 和 β 之间的内聚度的值。

步骤一(创建无向图) 首先将本体转变成为一个图, $O = \langle C, D, w \rangle$, 这里, C 是图的顶点,表示本体中公理的集合, $C = \{\alpha, \beta, \gamma, \dots\}$; D 是顶点之间的边,在图中,任意两个顶点之间都有边的相连; w 是边的权重,表示公理之间内聚度的值, $w(\alpha, \beta) = f(\alpha, \beta)$, $\alpha, \beta \in O, \alpha \neq \beta$ 。

步骤二(计算内聚度的值) 根据本体的推理信息来计算得到每一对公理之间的内聚度的值。

步骤三(划分模块) 我们将采取下面的算法^[8]来模块化本体。

定义1(模块化的标准) 顶点集合 $I \subseteq C, G = (C, D, w)$ 是一个模块的充分必要条件是:

- 由 I 集中顶点构成的图 (V_I, E_I, W_I) 是 G 的子图;
- I 中顶点和不属于 I 中的顶点的最大权值小于 I 中顶点之间权重的最小值;

$$\max_{\{(v, v') \in D | (v \in I \cap v' \notin I) \cup (v \in I \cap v' \in I)\}} w(v, v') < \min_{(u, u') \in E_I} w(u, u')$$

- I 是满足以上两个条件 G 中最大的顶点集合。

这个模块化划分的标准跟我们对模块本质的认识相符合,因为该方法将内聚性大的公理划分到了一起。使用该方法,我们就能划分出最优的模块,模块中公理的内聚性最大,模块之间的公理内聚性最小。

2.2 内聚度的计算

内聚度表示公理之间相互依赖的程度,这篇文章将会基于推理信息来计算公理之间内聚度的值。在 DL 中,所有的推理问题都可转化成公理相对于本体一致性判断的问题^[7],每一次推理会涉及到本体中的若干公理,我们把这些公理组成的集合叫做推理集合,记为 $R(\alpha)$,定义如下:

定义2(推理集合, Reasoning Set) 公理集合 R 是公理 α 在本体 O 上的推理集合的充分必要条件是:

R 是本体中公理集合的子集;

α 相对于 R 的一致性 $\Leftrightarrow \alpha$ 相对于 O 的一致性。

在众多的推理集合中,存在最小的推理集合,它由一次基于本体的推理必须要涉及到的公理组成,记为 $mR(\alpha)$ 。最小推理集合是推理集合的子集, $mR(\alpha) \subseteq R(\alpha)$ 。

定义3(最小推理集合, Minimal Reasoning Set) 公理集合 mR 是公理 α 在本体 O 上的最小推理集合的充分必要条件是:

mR 是公理 α 在本体 O 上的推理集合;

mR 为空集,或者除去集合 mR 中任何一条公理,将导致 α 相对于公理集合 mR 和相对于本体 O 的一致性不相同。

很明显,相对于最小推理集合的推理效率是最高的。如果我们能将推理限制在接近于最小的推理集合内,那么推理的效率将会大大提高,这也是我们算法假设的依据。在同一最小推理集合中的公理的内聚度增强,这样这些公理在模块化的划分时,就越有可能被划分到同一模块。

在使用 Tableaux 算法^[7]来解决推理问题时,我们通过计算推理的有效路径来取得最小的推理集合。假设求解公理 α 对于本体 O 的一致性的最小推理集合, O 的公理集合为 A ,具体的算法如下所示:

算法1 求解最小推理集合的算法

```

输入:公理  $\alpha$  和公理集合  $A$ 
输出:最小推理集合  $mR$ 
1: 将公理  $\alpha$  转换为概念  $C_0$ 
2: if  $C_0$  和  $A$  中的公理  $\beta$  转换的概念存在着冲突 then
3:    $mR \leftarrow \beta$ ; return  $mR$ ; //  $\alpha$  对于本体  $O$  不一致 (inconsistent)
4: else 使用表1所示的转化规则展开  $C_0$  一次; // 使用转化规则,  $\alpha$  将会以树的形式增长
5:   if  $C_0$  展开的树的分支存在着和  $A$  中公理对应的概念存在着冲突 (clash) then
6:     将  $A$  发生的冲突的所有公理添加到  $mR$  中;
7:     if  $C_0$  中所有的分支都发生了冲突 then
8:       return  $mR$ ; //  $\alpha$  对于本体  $O$  不一致
9:   else if 已经不能再展开  $C_0$  构成的树 then
10:     使用转化规则展开和  $\alpha$  相关的  $A$  中的公理;
11:     if 这些公理对应的概念展开的树和  $C_0$  中分支发生冲突 then
12:       将发生的冲突的树所代表的公理添加到  $mR$  中;
13:   else if 所有节点已不能再展开或者被阻塞 (blocked) 且  $C_0$  的分支没全部发生了冲突 then
14:      $mR \leftarrow \Phi$ ; return; // 将  $mR$  设为空集,  $\alpha$  对于本体  $O$  一致
15:   else 跳转到第5行,继续使用转化规则展开未发生冲突的分支;
16: return  $mR$ ;

```

在求解最小推理集合的算法中,如果 α 对于 O 一致,则 O 中所有的公理和 C_0 展开的树的分支不全发生冲突,最小推理集合将是空集;如果 α 对于 O 不一致,则 C_0 展开的树所有的分支会和 O 的公理集合 A 中的某些公理发生冲突,这些发生冲突的公理是判断 α 要使用的公理, α 相对于该公理集合和相对于本体 O 的一致性是一致的,根据定义2,该公理集合是公理 α 在本体 O 上的推理集合,又因为除去该推理集合内任意一个公理,都会导致 α 相对于该公理集合和相对于本体 O 的一致性不相同,根据定义3,可以知道该推理集合是最小推理集合。这样通过上面的算法可以求出公理 α 对于本体 O 的最小推理集合 mR 。然后,给集合 mR 中公理间的内聚度的值加 ϵ , ϵ 是一个正实数:

$$f(\alpha, \beta) = f(\alpha, \beta) + \epsilon, \alpha, \beta \in mR, \alpha \neq \beta \quad (1)$$

²更多的转化规则请参考文献^[9]。

表1 描述逻辑语言的基本转化规则²

$\sqcap$ 的转化规则:	
条件:	O 包含 $(C_1 \sqcap C_2)(x)$, 但是不包含 $C_1(x)$ 和 $C_2(x)$
动作:	$O' = O \cup \{C_1(X)C_2(X)\}$
$\sqcup$ 的转化规则:	
条件:	O 包含 $(C_1 \sqcup C_2)(X)$, 但是既不包含 $C_1(x)$ 也不包含 $C_2(x)$
动作:	$O' = O \cup \{C_1(X)\}, O' = O \cup \{C_2(X)\}$
$\exists$ 的转化规则:	
条件:	O 包含 $(\exists R.C)(x)$, 但是 O 中不存在个体 $z, C(z)$ 和 $R(x, z)$ 属于 O 中
动作:	$O' = O \cup \{C(z), R(x, z)\}$
$\forall$ 的转化规则:	
条件:	O 包含 $(\forall R.C)(x)$ 和 $R(x, y)$, 但是 O 不包含 $C(y)$
动作:	$O' = O \cup \{C(y)\}$
$\geq$ 的转化规则:	
条件:	O 包含 $(\geq nR)(x)$, 但是 O 中不存在个体 $z_1, \dots, z_n$ 使得 $R(x, z_i) (1 \leq i \leq n)$ 和 $z_i \neq z_j, 1 \leq i < j \leq n$
动作:	$O' = O \cup \{R(X, z_1) 1 \leq i \leq n\} \cup \{z_i \neq z_j 1 \leq i < j \leq n\}$
$\leq$ 的转化规则:	
条件:	O 包含不同的个体 $y_1, \dots, y_n, (\leq nR)(x)$ 和 $R(x, y_1), \dots, R(x, y_{n+1})$, 一些不存在 O 中
动作:	对于每一对满足 $i > j$ 和不存在 O 中的 y_i 和 y_j , 通过使用 y_j 代替 y_i 得到 $O_{i,j}, O_{ij} = [y_i/y_j]$

每一次进行推理,通过分析推理过程得到最小推理集合,然后使用公式(1)改变公理之间内聚度的值。可以看出,公理之间的内聚度是随着对本体的使用变化的,从而使用步骤三进行划分的模块也是不停更新的。基于推理信息的模块化方法能够很好地反映模块化动态性的特点。

3 实验与评估

基于推理信息的本体模块划方法的目的是要提高推理的性能。评估这个算法的实验过程分为三个步骤,首先,随机生成训练的公理在目标本体中推理;然后使用基于推理信息的算法将本体划分成模块;最后我们再随机地生成一些测试的公理在已被划分的本体和原本体上进行推理,评估模块化后带来的推理性能提高的效果。根据训练公理和测试公理所属的范围的不同,我们进行了三组实验。在第一组实验中,训练公理和测试公理都在同一较小的范围内,也就是将构成它们的概念和作用限定在相同的词汇集合内;在第二组实验中,将不限定训练公理和测试公理的范围;第三组实验,限定训练公理和测试公理的范围,并且该范围经过几次跳跃性的变化。

在实验中,我们设定 $\epsilon=1$ 。所有的实验都是在 2G Hz 的 Intel 处理器、2G 的内存和 Windows X PService Pack 2 的环境下进行的。Pellet³ 被用来进行推理。Pellet 要使用到 Java 工具的支持,我们使用的是 Sun's Java 1.5.0 Updata 5 并且 Java 虚拟机的内存被设为 900MB。

3.1 实验数据

我们使用四个本体来测试基于推理信息的本体模块化方法。这四个本体分别是 GALEN⁴, DOLCE⁵, SUMO⁶ 和 GO⁷, 本体 GALEN 和 DOLCE 只含有 TBox, 对于 SUMO 和 GO,

它们 ABox 的内容将被去除掉。这些本体分为两组,第一组是 GALEN 和 DOLCE, 它们的结构比较复杂,在公理中使用了很多构造式。GALEN 使用的是描述逻辑 *SHORF* 的语法, DOLCE 使用的是描述逻辑 *SHORF* 的语法, DOLCE 因为其复杂性导致现在没有推理机能够处理它,所以它被分为了几个模块。在本实验中,使用的是 DOLCE OWL 版本 397。第二组本体 SUMO 和 GO, 结构比较简单,仅仅使用了描述逻辑语言 *EL* 的语法。

3.2 公理的随机生成和基于模块的推理方法

我们使用限定的概念、作用和构造式的集合,从中随机地抽取元素组成公理,不符合语法的公理和仅仅从本身就可判断出真假的公理被去除掉,并且在生成测试公理时,只选取相对于本体不一致的公理。我们设定公理中使用的概念、作用和构造式的总个数为 5~10 个。在第一组实验和第三组实验中,生成公理的概念、作用的集合被限定在 1/10 的元素范围内,在第二组实验中概念和作用可以选取整个本体范围内任意的元素,在这些实验中,训练公理的个数为 1000 个,测试公理的个数为 100 个。另外一点要注意的是,构造式的选取范围是限定在描述该本体所使用的描述逻辑语言的范围内的。

当本体被划分成小的模块后,我们就要在上面执行推理调查其提高推理性能的效率。在本实验中,我们使用了一种简单的方法来进行基于模块中的推理,根据要执行的公理包含的概念和作用,计算出涉及到的模块,然后在这些模块中进行推理。要注意的是,这种基于模块的推理方法有可能会推导致推理的不正确性,原因在于公理的最小推理集合涉及到的某些元素并不在这些模块内。在后面的实验,可以看出这种简单的基于模块的推理方法,推理的正确性能得到很好的保证,特别是在限定范围的实验中。

3.3 实验一:限定范围的实验

我们采用 4.2 节所示的方法在 1/10 本体元素范围内生成 1000 个训练公理在本体上进行推理,然后使用基于推理信息的方法将本体划分为模块,最后我们使用 100 个测试公理在模块上进行推理,并且和这些公理在原本体上推理进行比较。模块划分的信息和推理比较的信息如图 1、表 1 和表 2 所示。表中上标为 1 的部分是本实验的结果,从中我们可以看出,推理的时间缩短了将近 6 到 7 倍,并且推理的正确性也能得到很好的保证。

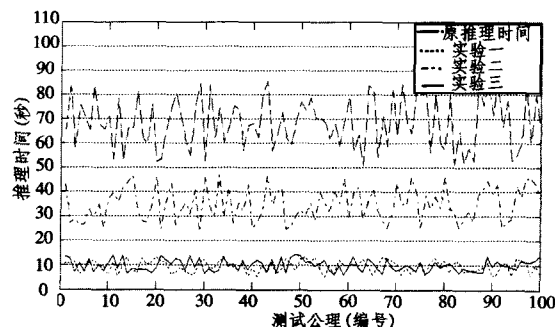


图1 GALEN 本体的推理时间比较图

³更多 Pellet 的信息请参考 <http://pellet.owldl.com/>

⁴<http://www.cs.man.ac.uk/~horrocks/OWL/Ontologies/galen.owl>

⁵<http://www.loa-cnr.it/DOLCE.html>

⁶<http://ontology.teknowledge.com/>

⁷<http://www.geneontology.org>

3.4 实验二:不限定范围的实验

不限定范围的实验的目的主要是调查在推理是无规律的情况下基于推理信息的本体模块化方法对推理性能影响的效果。在这个实验中,方法和限定范围的实验类似,只是生成公理的概念和作用的词汇可以在整个本体内取得。实验的结果如下图和下表所示,表中上标为2的部分是本实验的结果,可以看出,推理的时间只平均缩短了 67.2%,并且正确性也下降很快。其原因主要是当推理公理的不在限定的范围时,无法进行有效的模块划分。

3.5 实验三:范围改变的实验

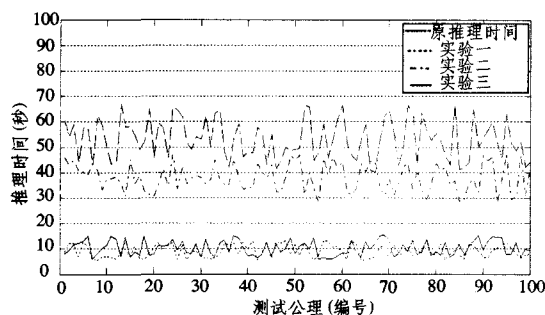


图2 DOLCE 本体的推理时间比较图

从上面的两个实验可以看出,当推理限定在某一范围内,本体推理的正确性能得到很好的保证,并且性能能得到极大的提高。在实际的应用中,在某一定的时期,用户对本体的使用应该是限定在某一范围内的。比如说,在某一段时间,大家都对某个热门话题感兴趣,那么这个小范围内的推理将会非常频繁,并且用户的使用还有一个特征就是动态性,在不同的时期,可能对本体的推理是在不同的范围内。为了调查范围变化对我们的算法的影响,我们做了这个范围改变的实验。

首先采用 4.3 节的方法对本体进行三次前两个阶段的操作,每次都将训练公理的生成范围限定在不同的本体元素内,最后我们在最终一次元素范围生成测试公理来比较推理性能的变化。实验的结果如下图和下表所示。可以看出,在这种情况下,推理的性能和正确性和实验一的结果比较接近。根据前面的分析,实验三这种情况是比较符合实际应用情况的,对本体的推理会在某段时期集中在某个范围内,并且具有动态性。这也证明了基于推理信息本体模块化方法的有效性。

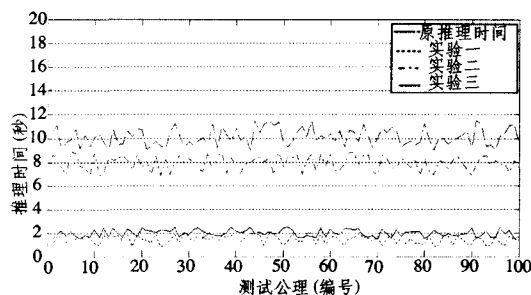


图3 SUMO 本体的推理时间比较图

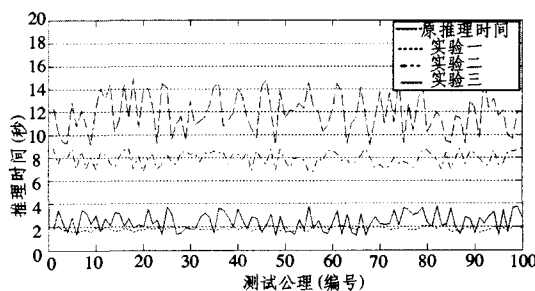


图4 GO 本体的推理时间比较图

表2 基于推理信息的本体模块化方法性能比较

本体	模块数 ¹	模块数 ²	模块数 ³	原平均推理时间(s)	平均推理时间 ¹	正确率 ¹	平均推理时间 ²	正确率 ²	平均推理时间 ³	正确率 ³
GALEN	3	10	6	68.50	9.53	98%	35.00	70%	10.01	96%
DOLCE	3	8	5	53.19	9.01	96%	37.89	71%	10.25	95%
SUMO	2	9	6	10.26	1.63	99%	8.07	79%	2.12	96%
GO	2	11	4	11.78	1.98	100%	8.01	80%	2.53	96%

表3 基于推理信息的本体模块化方法推理时间缩减程度比较

本体	推理时间缩减的程度 ¹	推理时间缩减的程度 ²	推理时间缩减的程度 ³
GALEN	13.9%	51.1%	14.6%
DOLCE	16.9%	71.2%	19.3%
SUMO	15.9%	78.5%	20.7%
GO	16.8%	68.0%	21.5%
总共	15.9%	67.2%	19.0%

结束语 为了提高本体推理的性能,本文提出了一种基于本体推理信息的本体模块化方法。该方法通过分析推理的最小推理集合来动态地改变公理之间的内聚度,然后根据内聚度来划分本体。在评估该算法时,我们主要是考察其提高推理性能和保证推理正确性的程度。实验结果证明了该算法的有效性。未来的工作主要是集中在两个方面。首先我们要将该方法应用到实际的系统中,考查它的性能;另一方面,我们计划做更多的工作来保证其推理的正确性。

参考文献

- [1] Berners-Lee T, Hendler J, Lassila O. The semantic Web. Scientific American, May 2001:28-37
- [2] Spaccapietra S. Report on Modularization of Ontologies, Technical report, The Knowledge Web project, June 2004
- [3] Stuckenschmidt H, Klein M. Structure-based partitioning of large class hierarchies//Proc. ISWC-2004. 2004;289-303
- [4] Seidenberg J, Rector A. Web ontology segmentation: Analysis, classification and use//Proc. WWW-2006. 2006
- [5] Noy N, Musen M. The PROMPT suite: Interactive tools for ontology mapping and merging. Int. Journal of Human-Computer Studies, 2003, 6(59)
- [6] Grau B C, Horrocks I, Kazakov Y, et al. Just the Right Amount, Extracting Modules from Ontologies//Proc. www-2007. 2007
- [7] Baader F, Calvanese D, McGuinness D L, et al. The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press, 2003
- [8] Batagelj V. Analysis of large networks - islands. Presented at Dagstuhl seminar 03361: Algorithmic Aspects of Large and Complex Networks, August/September 2003
- [9] Horrocks I, Sattler U. A tableaux decision procedure for SHOIQ//Proc. of the 19th Int. Joint Conf. on Artificial Intelligence (IJCAI 2005). 2005:448-453