

普适计算环境中异构资源互操作的代数方法研究

姜丽芬 赵子平

(天津师范大学计算机与信息工程学院 天津 300387)

摘 要 分析了普适计算对异构资源互操作和集成的需求,以及目前异构资源互操作、集成技术中存在的问题。针对异构信息源互操作技术存在的语义层的问题,提出了一种本体代数方法,建立了一套较完整的本体代数理论体系。本研究完善和发展了已有的本体代数理论,为普适计算环境异构数据的集成和互操作、异构本体资源的管理、查询等提供了理论基础。

关键词 本体,本体代数,普适计算,异构资源,互操作

中图分类号 TP301 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2014.10.054

Research on Algebraic Method of Interoperation of Heterogeneous Resource in Pervasive Computing Environment

JIANG Li-fen ZHAO Zi-ping

(College of Computer and Information Engineering, Tianjin Normal University, Tianjin 300387, China)

Abstract This paper analyzed the requirements and the existing problems of interoperation and integration of heterogeneous resource in pervasive computing environment. In view of the existing semantic problems in heterogeneous resource, an algebraic method for ontology was proposed and a relatively complete algebra theory for ontology was established. It has improved and developed the existing algebra theory for ontology, and has provided a theoretical guidance for interoperation, integration, management and query of heterogeneous resource in pervasive computing environment.

Keywords Ontology, Ontology algebra, Pervasive computing, Heterogeneous resource, Interoperation

1 引言

在普适计算环境中,存在着大量分布的、异构的动态数据资源。如何共享这些资源,以随时随地的方式提供对数据透明的访问是普适计算研究中的一个重要课题。

目前,对于分布式异构数据源的互操作和集成研究面临很多问题。文献[1]从技术的角度区分了信息集成的 3 个层次:技术集成、语法集成和语义集成。技术集成、语法集成的问题主要是由软硬件环境、数据结构、概念模型等的差别引起的,对于这些问题可以通过统一数据结构标准、开放接口标准和服务功能模块化等来解决。而语义层的集成相对复杂得多,如何有效地建立不同领域的概念之间的语义关系,一直是信息系统集成和互操作研究领域的一个不可回避的问题。为了解决跨信息源的信息语义异构的问题,必须理解与信息相关的语义。尽管科技在不断地进步,但让机器精确地理解与信息相关的语义仍然还是一件很困难的事情。随着 XML 的出现,对结构上的异构问题的解决已经取得了很大的进展。然而,普适计算环境下系统的分布性、异构性、数据的多样性和动态性等特点决定了单一的结构化方法不再适应新的信息集成和互操作的需求。因此,需要提供全新的理论和有效的方法,动态地共享、集成和检索这些数据信息。

本文通过引入本体^[2-4]上的代数方法来解决上述问题。当本体改变时,引入的代数机制使本体的管理和维护变得更容易。代数的方法是一种公理化的方法,本体代数方法是本体集成和异构本体间互操作和数据检索的理想技术之一。目前,本体代数已引起国内外学者的关注,但尚处于起步阶段,这一研究领域还面临很多挑战。

在下面的讨论中,我们先从本体概念的描述性定义开始,然后通过严密的逻辑推理,建立一套本体代数理论体系,并给出本体代数的应用。

2 本体代数

2.1 本体概念之间的基本关系

本体表现为一组概念以及概念之间的关系及规则的集合。在本体表示中,从语义上讲,概念之间最基本的关系有 4 种^[5]:Part-of, Kind-of, Instance-of 和 Attribute-of。关系 Part-of 的作用是表达概念之间部分与整体的关系,例如,(CPU, 电脑)表示 CPU 是电脑的一部分;关系 Kind-of 表示概念之间的继承关系,类似于面向对象技术中父类和子类之间的继承关系,例如,(电脑,办公用品)表示电脑是办公用品的子类;关系 Instance-of 的作用是表达概念的实例与概念之间的关系,类似于面向对象中的对象和类之间的关系,在语义学上可

到稿日期:2014-12-24 返修日期:2014-04-11 本文受国家自然科学基金项目(61103074),天津市自然科学基金(11JCYBJC00600),天津师范大学博士基金项目(52XB1101)资助。

姜丽芬 女,博士,教授,主要研究方向为普适计算、人工智能,E-mail:wxjlf@sina.com;赵子平 男,博士,副教授,主要研究方向为人工智能。

以用 Hyponymy(Is-a)表示,例如,(我的电脑,电脑)表示“我的电脑”是“电脑”类的实例。关系 Attribute-of 的作用是表达某个概念是另一个概念的属性,例如,(价格,电脑)表示“价格”是“电脑”的一个属性。图1给出了本体概念之间关系的一个实例。

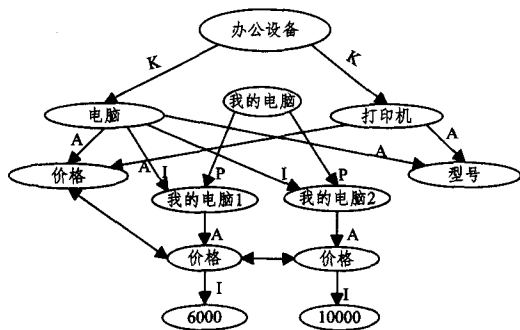


图1 本体概念的关系图

其中,双箭头表示等价概念,P表示 Part-of 关系,K表示 Kind-of 关系,I表示 Instance-of 关系,A表示 Attribute-of 关系。

在下面本体的讨论中,我们以本体概念之间的语义基本关系为基础,给出本体的定义。

2.2 本体

在引入本文的本体描述性定义之前,我们首先定义几个概念。

定义1 概念用一个二元组 $c = (s, a)$ 表示,其中 s 是一个字符串(概念定义的字符串), a 是概念属性的集合。

定义2 设 S 是一个非空的概念集合,定义 $S \times S$ 上的二元关系集合 $\Sigma = \{\text{Synonymy, Part-of, Is-a, Kind-of, } \emptyset\}$ 。其中 Synonymy 表示两个概念的同义关系; $\emptyset$ 表示两个概念之间不存在任何关系。其他关系如:Part-of、Kind-of 和 Is-a 的含义是本文 2.1 节中给出的概念之间的基本关系。

下面给出本文中本体的描述性定义。

定义3 一个本体 O 表示为一个关系 $O = \langle S, \psi \rangle$,其中, S 是定义1中给出的概念集合, ψ 为从概念集合 S 到 Σ 的映射(Σ 是定义2中给出的 S 上的二元关系集合), $\psi: S \times S \rightarrow \Sigma$ 。注:当 $i \neq j$ 时,在一般情况下 $\psi(x_i, x_j) \neq \psi(x_j, x_i)$ 。

例如,概念集 $S = \{\text{办公用品, 电脑, 我的电脑, 我的计算机, CPU}\}$,我们可以定义如下映射关系:

$\psi(\text{电脑, 办公用品}) = \text{Kind-of}$

$\psi(\text{我的电脑, 电脑}) = \text{Is-a}$

$\psi(\text{CPU, 电脑}) = \text{Part-of}$

$\psi(\text{我的电脑, 我的计算机}) = \text{Synonymy}$ 等。

定义4(本体相关的图) 假设本体 $O = \langle S, \psi \rangle$,与本体 O 相关的图表示如下:

一图顶点的集合是 $\{a | a \in S\}$ 。

一图边的集合是 $\{(a, b) | a, b \in S \wedge \psi(a, b) = \leq \wedge \leq \in \Sigma \wedge \leq \neq \emptyset\}$,即图的边表示概念之间的关系。

定义5 设 S 是一个概念集合, S 上的二元关系是 R ,如果满足自反性、反对称性和传递性,则称 R 是 S 上的偏序关系,记为 $\leq$,序偶 $\langle S, \leq \rangle$ 称为偏序集。

定理1 $\forall \leq \in \Sigma$,且 $\leq \neq \emptyset$,则 S 上的二元关系 $\leq$ 是一个偏序关系,记为: $(S, \leq)$ 。

证明:下面仅以 Part-of 关系为例来证明。我们约定:每

个概念都是其自身的一部分。

设 $\leq \in \Sigma$, $\leq$ 为 Part-of 关系。

1)对任意的概念 $a \in S$,显然 $a \leq a$ 成立,故“ $\leq$ ”关系是自反的。

2)对任意的概念 $a, b \in S$,如果 $a \leq b$ 且 $b \leq a$,则必有 $a = b$,因此,关系“ $\leq$ ”是反对称的。

3)对任意的概念 $a, b, c \in S$,如果 $a \leq b$ 且 $b \leq c$,那么必有 $a \leq c$,因此,关系“ $\leq$ ”是传递的。

因此,Part-of 关系是一个偏序关系。其他的关系如 Is-a, Kind-of 关系的证明完全类似,此处略。

由定理1可知, $\forall \leq \in \Sigma$,且 $\leq \neq \emptyset$, S 上的二元关系“ $\leq$ ”是一个偏序关系。由数学理论可知,一个偏序关系的盖住关系是唯一的。因此,可以用盖住关系作为二元关系 $\leq$ 的最小集。因为偏序关系的哈斯图(盖住关系图)正好是集合 S 上的一个层次关系图。

定义6(盖住) 设 $\leq \in \Sigma$,且 $\leq \neq \emptyset$, $(S, \leq)$ 是一偏序关系,对 $a, b \in S$,若 $a \leq b$, $a \neq b$ 且不存在 $c \in S$,使得 $a \leq c$ 且 $c \leq b$,则称 b 盖住 a ,记为 $a < b$ 。并且记 $\text{COV}(S) = \{ \langle a, b \rangle | a, b \in S; \text{且 } a < b \}$ 。

举例:假设概念集合 $S = \{\text{主板(motherboard), 主机(mainframe), 计算机(computer)}\}$ 。以偏序关系 Part-of 为例,其偏序关系 Part-of 可以定义为主板是主机的组成部分,主机是计算机的组成部分,主板也可以说是主机的组成部分。另外,每一个概念都是自身的组成部分。因此,偏序关系可以由如下集合给出:

$\{ \langle \text{motherboard, motherboard} \rangle, \langle \text{mainframe, mainframe} \rangle, \langle \text{computer, computer} \rangle, \langle \text{motherboard, mainframe} \rangle, \langle \text{mainframe, computer} \rangle, \langle \text{motherboard, computer} \rangle \}$ 。

它的盖住关系为 $\text{COV}(S) = \{ \langle \text{motherboard, mainframe} \rangle, \langle \text{mainframe, computer} \rangle \}$ 。盖住关系的本体图层次结构如图2所示。

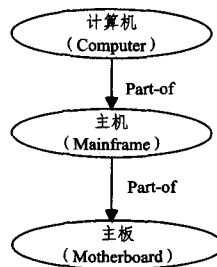


图2 本体层次图

由集合论的知识可知,盖住关系是唯一的,并有很好的性质,可以用盖住的性质画出哈斯图。不失一般性,在下面的讨论中,对概念集 S 上的二元关系“ $\leq$ ”,在默认的情况下只考虑盖住关系。

3 本体上的代数运算及应用

3.1 本体上的代数运算

定义7(本体并运算) 设有两个本体 $O_1 = \langle S_1, \psi_1 \rangle$, $O_2 = \langle S_2, \psi_2 \rangle$,假设两个本体概念之间的关系规则集为 R ,本体的“并”操作 Θ 是两个本体之间的映射 $O_1 \Theta O_2 = O_3$,记作: $O_1 \cup O_2 = O_3$ 。这里 $O_3 = \langle S_3, \psi_3 \rangle$ 。其中, $S_3 = S_1 \cup S_2$, ψ_3 的定义如下:

- 1) $\forall s_1, s_2 \in S_1 \wedge s_1, s_2 \notin S_2$ 时, $\psi_3(s_1, s_2) = \psi_1(s_1, s_2)$
- 2) $\forall s_1, s_2 \notin S_1 \wedge s_1, s_2 \in S_2$ 时, $\psi_3(s_1, s_2) = \psi_2(s_1, s_2)$
- 3) $\forall s_1 \in S_1 \wedge s_2 \in S_2, s_1, s_2 \notin S_1 \cap S_2, \psi_3(s_1, s_2) = \psi(s_1, s_2)$

s_2)

其中, ψ 是由本体 O_1 和本体 O_2 概念之间的关系规则 R 确定的关系映射。

4) $\forall s_1 \in S_1 \wedge s_2 \in S_2, s_1, s_2 \in S_1 \cap S_2$, 如果 $\psi_1(s_1, s_2) = \psi_2(s_1, s_2)$, 则 $\psi_3(s_1, s_2) = \psi_1(s_1, s_2)$, 否则, $\psi_3(s_1, s_2) = f_{merge}(\psi_1(s_1, s_2), \psi_2(s_1, s_2))$ 。

两个本体概念之间关系的规则集 R 的定义, 参考文献[6]中的推理规则算法。我们也可以自定义规则, 例如: 在图3中描述了两个交通工具本体 O_1 和本体 O_2 。

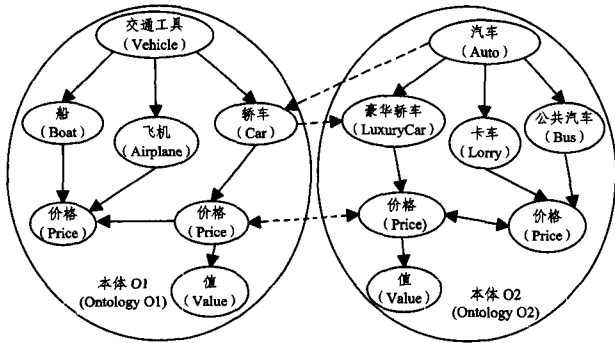


图3 交通工具本体 O_1 和 O_2

我们可以定义如下两本体之间的连接规则:

$(O_2, \text{LuxuryCar Kind-of } O_1, \text{Car})$

$(O_1, \text{Car Kind-of } O_2, \text{Auto})$

$(O_1, \text{Price Equals } O_2, \text{Price})$

$(X \text{ Instance-of } O_1, \text{Car}), (X \text{ hasA } X, \text{Price}), (Y \text{ Instance-of } X, \text{Price}),$

$(Y \text{ hasA } Y, \text{Value}), (Z \text{ Instance-of } Y, \text{Value}), (Z > 500000)$

$\Rightarrow (X \text{ Instance-of } O_2, \text{LuxuryCar})$ 等。

在普适计算环境下本体之间的并运算中, 可以将分布的异构系统中的多个资源本体统一到一个集中的模式下, 实现自下而上的系统集成。同时本体的并运算可以应用于多个本体数据检索、查询等工作。

定义8(本体交运算) 设有两个本体 $O_1 = \langle S_1, \psi_1 \rangle, O_2 = \langle S_2, \psi_2 \rangle$, 两个本体概念之间的关系规则集为 R , 本体的“交”操作 Θ 是两个本体之间的映射 $O_1 \Theta O_2 = O_3$, 记作: $O_1 \cap O_2 = O_3$ 。这里 $O_3 = \langle S_3, \psi_3 \rangle$ 。其中, $S_3 = \{s_1, s_2 \mid \forall s_1 \in S_1 \wedge s_2 \in S_2, \text{使得 } \psi_3(s_1, s_2) \neq \emptyset\}$, 即两个本体间有语义关系的概念结点集合。 ψ_3 是由本体 O_1 和本体 O_2 概念之间的关系规则 R 确定的关系映射。

本体交运算决定了两个本体相关的概念部分。

定义9(本体的差) 设有两个本体 $O_1 = \langle S_1, \psi_1 \rangle, O_2 = \langle S_2, \psi_2 \rangle$, 两个本体概念之间的关系规则集为 R , 本体的“差”操作 $\ominus$ 是两个本体之间的映射 $O_1 \ominus O_2 = O_3$, 记作: $O_1 - O_2 = O_3$ 。这里 $O_3 = \langle S_3, \psi_3 \rangle$, 其中, $S_3 = S_1 - S_2, \forall s_1, s_2 \in S_3, \psi_3(s_1, s_2) = \psi_1(s_1, s_2)$ 。

本体的差集可以用于普适计算环境中动态资源本体维护和修正^[7]。例如, 在物流系统中, 货物需要经常流动。因此, 相应的本体会不断地发生变化。当某一个本体中新增加了一

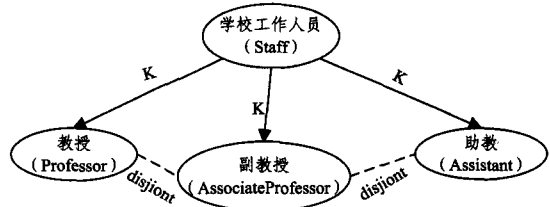
些对象, 在把它们加入到本体中的同时, 相应的本体关系也需要修改。如果新增的概念结点出现在该本体与另一个本体的差集中, 它和其他领域本体间的映射关系就是独立的, 则该本体与另一个本体之间的联系就不用修改。反之, 系统就需要修改该本体与其另一个本体之间的关系以反映本体的动态变化。

3.2 本体描述语言

我们使用本体描述语言 OWL^[8] 来描述本体。OWL 文档继承了 RDF 文档的特点, 采用 Class、Property 和 Instances 三元组的标签形式表示。利用本体的 RDF 图, 可以很容易地构造出本体的 OWL 文档。例如, 在下面例子中, 定义了类——教授 (Professor)、副教授 (AssociateProfessor) 和助教 (assistant) 类, 他们都是学校工作人员 (Staff) 的子类, 其中, 教授、副教授和助教是 Subject, 子类关系为 Predicate, Object 是学校工作人员。最后的定义表示各类 (副教授、教授和助教类) 都是不相交的。

```
<owl:Class rdf:ID="professor">
  <rdfs:subClassOf rdf:resource="# Staff"/>
</owl:Class>
<owl:Class rdf:ID="associateProfessor">
  <rdfs:subClassOf rdf:resource="# Staff"/>
</owl:Class>
<owl:Class rdf:ID="assistant">
  <rdfs:subClassOf rdf:resource="# Staff"/>
</owl:Class>
<owl:Class rdf:about="# Professor">
  <owl:disjointWith rdf:resource="# professor"/>
  <owl:disjointWith rdf:resource="# assistant"/>
</owl:Class>
```

同样, 由 OWL 文档也能很直观地构造出 RDF 图。根据上述文档类的定义可以构造出 RDF 图, 如图4所示。



其中, K-表示子类 (Kind-of 关系)

图4 由 OWL 文档构造出的 RDF 图

可以直观地由本体的 RDF 图构造出 OWL 文档, 反之亦然。因此, 本体上的代数运算可以很容易地使用 OWL 语言进行描述。

3.3 本体及本体代数的应用

(1) 本体层的查询应用

在很多情况下, 单一的信息源是不能满足用户需求的。如果应用跨多个领域, 应用必须使用来自多个领域的术语。从另一个角度看, 即使克服了所有技术上的难题, 完成了对异构数据源的集成并对整合后的资源能够提供一致的访问方法, 但由于缺乏有效的语义处理手段, 集成系统的数据资源的访问、数据检索等还存在着语义上的问题。语义异构不仅要考虑其内容, 还要考虑其相互关系, 也就是我们通常所说的上下文。

目前,各种信息检索技术基本都是建立在关键词的基础之上,大多采用字符串匹配的方式进行检索。而对计算机而言,关键词几乎没有任何语义。尽管目前对检索技术做了许多改进,但是,各种改进方法也只是从算法的角度进行改进,并没有添加处理语义方面的机制。因此,检索的性能还是难以从本质性上提高。不失一般性,以目前主流信息检索技术——倒排文件为例。在倒排文件中,文档用一系列关键词来表示,这些关键词描述了文档相关的内容。只要找到了文档,就能找到文档中的关键词。反之,如果按关键词建立到文档的索引,便可以根据关键词检索到相关的文档。倒排文件的方法几乎被当前所有的商用 IR (information retrieval, IR) 系统所采用^[9]。表 1 是一个倒排文件的例子。

表 1 文件倒排表

关键词	文档编号	位置
交通工具 (Vehicle)	25	68
汽车 (Automobile)	25	32
轿车 (Car)	10	12
豪华轿车 (LuxuryCar)	25	23
船 (Boat)	11	30
轮船 (steamboat)	11	16

例 1 用户提出查询请求:“查找所有交通工具信息”。按照表 1 查询,可能查找到文档 25,却有可能把文档 10,11 漏掉,而文档 10 和文档 11 显然也涉及到了交通工具的信息。

应用本体及本体代数方法可以很方便地解决上述问题。在例 1 中,交通工具 (Vehicle) 在最顶层,下面有许多子概念,这种层次构成了一个本体。用本体图的结构表示如图 5 所示。

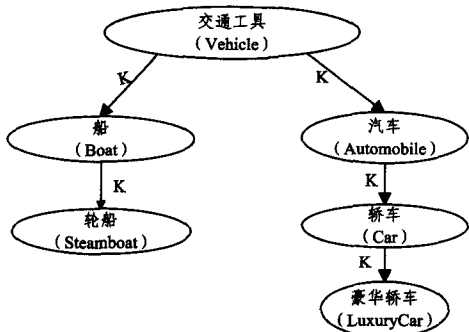


图 5 交通工具本体图

其中,交通工具 (Vehicle) 本体包括的概念集为: $O = \{Vehicle, Boat, Steamboat, Automobile, Car, LuxuryCar\}$, 在图 5 中, K 表示 Kind-of 关系。为了“查找所有交通工具 (Vehicle) 的信息”, 可以从顶层 Vehicle 出发, 利用本体概念的 Kind-of 关系进行处理: $Vehicle \rightarrow Boat \rightarrow Steamboat$; $Vehicle \rightarrow Automobile \rightarrow Car \rightarrow LuxuryCar$ 。这样, 就可以把包括交通工具的所有文档检索出来。

对于更复杂的跨多本体的查询可以使用本体上的代数运算, 如本体并、交和差等运算。

(2) 本体互操作应用

信息源之间的互操作和信息源的集成是不同的。信息互操作是指信息源维持各自独立的信息, 而信息集成则是指来自多个信息源的信息合并成为一个新的整合后的信息。

随着科技的发展, 有大量的本体可以用到, 而且本体本身可能相当大。当信息源之间的矛盾不能彻底解决时, 这些资

源之间的信息完全集成是不可行的。我们提出的本体代数方法, 通过本体的交运算, 选择用户感兴趣的部分建立本体之间的关系来实现本体间的互操作, 并不需要把两个本体的交集完全独立成为一个新的本体, 这种方法便于实际应用。

我们举一个简单的例子: 假设 S 是裤业商场, 主要销售各种时尚裤子。S 本体涉及到很多的概念, 如顾客、裤子的尺码 (颜色、样式等)、售货员、供应商等。服装工厂域 F 涉及的概念如成裤、布料、缝纫机、线、原料供应商、雇员等。为了使用这些资源的信息, 没有必要把裤店和服装厂的本体信息完全合并。例如, 对于顾客来讲, 只是选择他们感兴趣的必需信息, 如裤子的颜色、尺码等建立连接, 进行相互操作即可, 如图 6 所示。

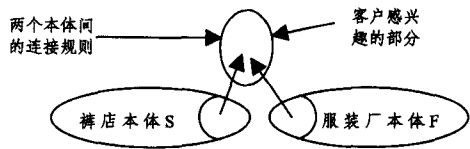


图 6 两个本体交的应用

代数的方法对本体应用具有理论的指导意义。因此, 我们忽略有关工程方面的问题, 只集中在数学理论方面的探讨。我们力图构建一套完整的本体代数理论体系来指导本体的应用。

结束语 斯坦福大学的 Wiederhold G. 在文献[2]中提出了本体代数的思想, 其目标是提供一种能用来操作那些存在语义差异的知识资源。文献[10]讨论了本体整合操作, 其两个本体整合后, 维持他们各自的标识, 通过增加桥或跨两个本体的连接来识别类似的元素。文献[11]通过名字匹配来探索等价和相似算法, 本体上的操作是受限的原始元素的拷贝, 如浅拷贝、深拷贝等。Chimaera^[12], Ontomorph^[13], Prompt^[14]等案例中一般是采用专家人工输入半自动的方法对待本体合并或修正^[7], 这些研究的重点都是针对某一领域的实际应用来开发相应的软件工具。与之相比, 我们的方法是构建一套完整的本体代数理论体系。

在我们的方法中, 定义了更标准的代数操作, 建立了一个较完备的代数理论, 并提供了大部分代数操作的需求。由于篇幅有限, 我们对提供的本体的同态、本体格的理论、本体冗余定理等不再赘述。

参 考 文 献

- [1] Stuckenschmidt H, Wache H. Context modeling and transformation for semantic interoperability[C]// Processings of Knowledge Representation meets Databases (KRDB). Berlin, Germany, 2000; 115-126
- [2] Wiederhold G. Interoperation, mediation, and ontologies[C]// Proceedings International Symposium on Fifth Generation Computer Systems (FGCS), Workshop on Heterogeneous Cooperative Knowledge-Bases. Tokyo, Japan, 1994(3): 33-48
- [3] Hauer I, Muntean C. An overview of ontologies and knowledge management. Universitatea de Stinte Agricole si Medicina Veterinaria a Banatului Timisoara[J]. Lucrari Stiintifice. Seria I. Management Agricol, 2010, 7(2): 1453-1410
- [4] Jonas R, Arenhart B. Ontological Frameworks for Scientific Theories[J]. Foundations of Science, 2012, 17(4): 339-356

(下转第 269 页)

接下来 s_3 进行转发,因为 $Adj[2][3]=1, Adj[5][3]=1$,所以 s_3 向状态 s_2 和状态 s_5 进行转发,根据信息传递规则 2 和 4,不做修改。

同理,接下来由 s_4, s_5 进行信息转发,得到的 Adj 为:

$$Adj = \begin{bmatrix} 1 & 0 & T_3+T_2 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ T_1 & 0 & 1 & 0 & 0 \\ T_1 & 0 & T_2 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

这样找出了不确定转移系统中所有确定可达的状态关系,接下来需要对 Adj 中元素值为 0 的状态对进行判定。由于 $Adj[3][4]=0, Adj[3][1]=T_1, Adj[1][4]=1$,因此 $Adj[3][4]=T_1$ 。

最后得到的 Adj 矩阵为:

$$Adj = \begin{bmatrix} 1 & 0 & T_3+T_2 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ T_1 & 0 & 1 & T_1 & 0 \\ T_1 & 0 & T_2 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

以下为采用矩阵相乘方法和信息传递方法求状态可达关系的实验结果比较。实验环境均为 Windows7+Core(TM)i3-3220 3.3GHz+4.0GB 内存+VC6。两个算法使用的数据输入输出过程相同,故其运行时间没有包括在内。

表 1 运行时间比较

状态数/ms	矩阵相乘/ms	信息传递/ms
5	0.0018	0.0021
10	0.0069	0.0070
20	0.0366	0.0269
30	0.1403	0.0765
50	1.0374	0.2049
70	4.8399	0.4231
100	30.2210	0.8885
130	90.3567	1.5140
150	283.5321	1.8891
200	917.8564	3.0163
300	9061.065	6.1892

从表 1 中可以发现,当状态数较少时,由于计算量较少,矩阵相乘法速度较快;当状态数较多时,矩阵相乘法的计算量

也急剧增加,信息传递法的优势就体现出来了。

结束语 本文设计了用信息传递方法求解不确定系统中可达关系的算法。充分利用不确定系统中每个状态所拥有的可达信息,通过信息收集、信息传递、信息更新,可以快速求出确定系统中状态之间的可达关系。本文的进一步工作有:

1. 优化信息传递策略,加快求解可达关系的速度。
2. 运用信息传递的思想求解不确定规划的强、弱、强循环规划解。

参考文献

- [1] Weld D S. Recent advances in AI planning [J]. AI Magazine, 1999, 20(2): 93-123
- [2] Cimatti A, Roved M, Traverso P. Strong planning in nondeterministic domains via model checking [C] // Proceedings of the 4th International Conference on AI Planning Systems. 1998: 36-43
- [3] Cimatti A, Roveri M. Conformant planning via symbolic model checking [J]. Journal of Artificial Intelligence Research, 2000, 13(3): 305-338
- [4] Wen Zhong-hua, Huang Wei, Liu Ren-ren. Method of hierarchical states in planning based on model checking [J]. Journal of Software, 2009, 20(4): 858-869
- [5] Cimatti A, Pistore M, Roveri M, et al. Weak, strong, and strong cyclic planning via symbolic model checking [J]. Artificial Intelligence, 2003, 47(1): 35-84
- [6] 黄丽芳, 文中华, 胡雨隆, 等. 不确定规划中状态循环可达关系的求解方法 [J]. 计算机应用研究, 2013, 30(9): 2689-2693
- [7] 胡雨隆, 文中华, 常青, 等. 不确定规划中状态非循环可达关系的求解方法 [J]. 计算机仿真, 2012, 29(5): 114-117
- [8] Huang Wei, Wen Zhong-hua, Jiang Yun-fei, et al. Observation reduction for strong plans [C] // Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI-07). 2007: 1930-1935
- [9] 文中华, 黄巍, 刘任任, 等. 模型检测规划中的状态之间的可达关系研究 [J]. 计算机学报, 2012, 35(8): 1634-1643
- [10] 谢希仁. 计算机网络(第五版) [M]. 北京: 电子工业出版社, 2007: 147-152

(上接第 260 页)

- [5] 邓志鸿, 唐世渭, 张铭, 等. Ontology 研究综述 [J]. 北京大学学报: 自然科学版, 2002, 38(5): 730-738
- [6] 崔巍. 用本体实现地理信息系统语义集成和互操作 [D]. 武汉: 武汉大学, 2004
- [7] 孙瑜. 本体修正 [D]. 北京: 中国科学院计算技术研究所, 2006
- [8] Severance C, Lan Horrocks. Standardizing OWL [J]. Computer, 2013, 46(11): 8-9
- [9] Repplinger, Chowdhury J G G. Introduction to Modern Information Retrieval (3rd ed) [M]. College and Research Libraries, 2011
- [10] Mitra P, Wiederhold G, Kersten M L. A Graph-oriented model for articulation of ontology interdependencies [C] // Proceedings of the 7th International Conference on Extending Database Technology, Advances in Database Technology. Konstanz, Ger-

many, 2000: 86-100

- [11] Noy N F, Musen M A. Smart: Automated support for ontology merging and alignment [C] // 21th Workshop on Knowledge Acquisition, Modeling, and Management. Banff, Canada, 1999
- [12] McGuinness D L, Fikes R, Rice J, et al. An environment for merging and testing large ontologies [C] // Proceedings of the 7th International Conference on Principles of Knowledge Representation and Reasoning. Colorado, USA, 2000: 483-493
- [13] Chalupsky H. Ontomorph: A translation system of symbolic logic [C] // Proceeding of the 7th International Conference (KR2000). San Francisco, CA, 2000: 471-482
- [14] Noy N F, Musen M A. Prompt: algorithm and tool for automated ontology merging and alignment [C] // Proceedings of the 2000 National Conference on Artificial Intelligence. Texas, 2000: 450-455