

## 基于 DFL 的缺省假设推理研究

黄 晋, 李凡长

(苏州大学 计算机科学与技术学院, 江苏 苏州 215006)

**摘 要:**日常生活中人们可以在信息不完全的情况下进行推理并得出较好的推理结论,而且在推理过程中,很多对象都是具有动态模糊性(Df Character)。因此文中针对研究对象以及它们之间的动态模糊性,提出了基于动态模糊逻辑(DFL)的缺省假设推理,并给出了缺省假设推理的框架描述、动态模糊(DF)知识的表示以及推理算法等。

**关键词:**动态模糊逻辑; 动态模糊推理规则; 缺省假设推理

**中图分类号:**TP181

**文献标识码:**A

**文章编号:**1673-629X(2006)11-0047-03

## Default Assumption Reasoning Based on DFL

HUANG Jin, LI Fan-zhang

(Computer Science and Technology School, Soochow University, Suzhou 215006, China)

**Abstract:** Mankind can use uncertain and incomplete information to infer. However, most of objects in the reasoning processing have dynamic fuzzy character. Due to this, propose default assumption reasoning based on DFL. Here, introduce the reasoning frame, dynamic fuzzy knowledge representation, reasoning algorithms and so on.

**Key words:** dynamic fuzzy logic; dynamic fuzzy reasoning rules; default assumption reasoning

## 0 引言

近些年来人工智能不断发展,很多具有实际用途、效率颇高的专家系统已被开发并被应用。人们也越来越关注动态的问题。事实上,在人们所生活的世界,客观事物都是不断运动发展的,它们所具有的一些属性也都具有模糊的和动态变化的特性。L. A. Zadeh 在 1965 年提出了模糊概念。几十年来,随着模糊理论的发展,模糊问题在某种程度上得到很好的解决,已开发出一些很好的模糊控制系统。对于动态系统的研究也有一定的发展,比如进程代数、动态时序逻辑等<sup>[1]</sup>。

人类的推理决策过程也是一个动态的不确定的过程,而且在信息模糊和缺乏的情况下也可以推出合乎常理的结论。因此文中主要是研究如何使这种不确定的、动态的过程在计算机上得以应用。考虑到如前所述客观研究对象大多数也是模糊的、动态的,因此采用动态模糊逻辑(DFL)作为研究的理论工具。动态模糊逻辑把动态性和模糊性紧密结合起来作为一个整体特性,并且比文献[2, 3]中更系统和完善。

## 1 动态模糊逻辑的基本理论

本节主要给出一些动态模糊逻辑的基本理论,详细的理论可参见文献[4]。动态模糊逻辑是以动态模糊集为理论基础的,相关理论见文献[4]。

## 1.1 动态模糊逻辑命题的逻辑演算

**定义 1** 一个具有动态模糊性的陈述句称为 DF 命题。用大写字母  $A, B, C \dots$  表示。

**定义 2** 度量一个 DF 命题真假度用 DF 数  $(\vec{a}, \vec{a}) \in [0, 1] \times [\leftarrow, \rightarrow]$  来表示,称为该命题的真假,常用小写字母  $(\vec{a}, \vec{a}), (\vec{b}, \vec{b}), (\vec{c}, \vec{c}), \dots$  表示。

**定义 3** 一个 DF 命题可以看成在闭区间  $[0, 1]$  上取值的变量,称为 DF 命题变量。

对于 DF 变量  $(\vec{x}, \vec{x}), (\vec{y}, \vec{y}) \in [0, 1]$  规定如下运算:

(1) 否定“ $\neg$ ”,例如:  $(\vec{x}, \vec{x})$  的否定(Negation)记为  $(\overleftarrow{\vec{x}}, \overleftarrow{\vec{x}})$ , 且  $(\overleftarrow{\vec{x}}, \overleftarrow{\vec{x}}) \triangleq ((1 - \vec{x}), (1 - \vec{x}))$ ;

(2) 析取“ $\vee$ ”,例如:  
 $(\vec{x}, \vec{x}) \vee (\vec{y}, \vec{y}) \triangleq \max((\vec{x}, \vec{x}), (\vec{y}, \vec{y}))$ ;

(3) 合取“ $\wedge$ ”,例如:  
 $(\vec{x}, \vec{x}) \wedge (\vec{y}, \vec{y}) \triangleq \min((\vec{x}, \vec{x}), (\vec{y}, \vec{y}))$ ;

(4) 条件“ $\rightarrow$ ”,例如:  
 $(\vec{x}, \vec{x}) \rightarrow (\vec{y}, \vec{y}) \triangleq (\overleftarrow{\vec{x}}, \overleftarrow{\vec{x}}) \vee (\vec{y}, \vec{y}) \triangleq \max((\overleftarrow{\vec{x}}, \overleftarrow{\vec{x}}), (\vec{y}, \vec{y}))$ ;

(5) 双条件“ $\leftrightarrow$ ”,例如:  
 $(\vec{x}, \vec{x}) \leftrightarrow (\vec{y}, \vec{y}) \triangleq \min(\max((\overleftarrow{\vec{x}}, \overleftarrow{\vec{x}}), (\vec{y}, \vec{y})), \max((\overleftarrow{\vec{y}}, \overleftarrow{\vec{y}}), (\vec{x}, \vec{x})))$

收稿日期: 2006-02-14

基金项目: 国家“八六三”高技术项目(2002AA881030)

作者简介: 黄 晋(1980-), 女, 安徽芜湖人, 硕士研究生, 研究方向为 DFL 的推理、Agent 等; 李凡长, 教授, 研究方向为动态模糊逻辑(DFL)、Agent、自主计算、李群等。

定义 4 DF 命题运算公式可定义为:

(1) 单个 DF 命题变元本身是一个合式公式;

(2) 如果  $(\vec{x}, \vec{x})P$  是一个合式公式, 那么  $(\vec{x}, \vec{x})\neg P$  也是合式公式;

(3) 若  $(\vec{x}, \vec{x})P$  和  $(\vec{y}, \vec{y})Q$  是合式公式, 那么  $(\vec{x}, \vec{x})P \vee (\vec{y}, \vec{y})Q$ ,  $(\vec{x}, \vec{x})P \wedge (\vec{y}, \vec{y})Q$ ,  $(\vec{x}, \vec{x})P \rightarrow (\vec{y}, \vec{y})Q$ ,  $(\vec{x}, \vec{x})P \leftrightarrow (\vec{y}, \vec{y})Q$  都是合式公式;

(4) 当且仅当有限次地应用(1), (2), (3)所得到的命题变元联结词和括号的符号串是合式公式。

## 1.2 DFL 的谓词演算

定义 5 DFL 谓词公式递归定义:

(1) 原子(一阶谓词符号)是公式;

(2) 若  $G, H$  是公式,  $T$  是 DF 真值指派值,  $(\vec{x}, \vec{x})$  是 DFL 中的自由变量, 那么  $G \wedge H$ ,  $G \vee H$ ,  $G \rightarrow H$ ,  $G \leftrightarrow H$ ,  $(\vec{x}, \vec{x})G$ ,  $(\forall (\vec{x}, \vec{x})G)$ ,  $(\exists (\vec{x}, \vec{x})G)$  是公式;

(3) DFL 中所有公式有限次使用(1), (2)后产生的符号串。

定义 6 DFL 中公式  $G$  的一个解释  $I$  由非空域和如下规则组成:

(1) 对于  $G$  中每个变量符号指定  $U$  中一个 DF 元素;

(2) 对  $G$  中每个  $n$  元函数符号指定映射  $U \xrightarrow{T} D$ ;

(3) 对  $G$  中每个  $n$  元谓词符号指定映射  $D \xrightarrow{T} B$ 。

## 2 缺省假设推理

缺省推理是由 Reiter 提出的, 在文献[5]中他给出了缺省逻辑。他从逻辑的角度给出了解决信息不完全时的推理方法。由于客观事物的属性大多是有模糊性的, 从而产生了可以处理表示和模糊推理的缺省推理的研究<sup>[6]</sup>。这些都是在逻辑的基础上通过约束和修改而成的。

文中通过对人类缺省思维过程的研究, 如图 1 所示, 发现在信息缺省的情况下, 人们总是事先进行假设, 再考察假设的结果。如果结果令人们满意就接受该假设, 否则否定假设, 再重新考虑。

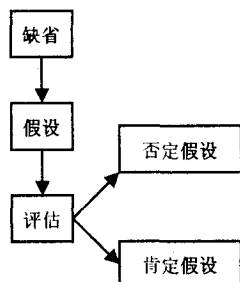


图 1 缺省假设推理过程

根据此过程, 从框架结构上考虑基于 DFL 的缺省假设推理, 并对相应的结构描述、推理规则、算法等方面进行描述。

### 2.1 缺省假设推理的框架结构描述

在基于动态模糊逻辑(DFL)的推理系统中, 为了提高推理效率, 把规则库分为一般的规则库和缺省规则库。推

理的框架结构如图 2 所示。

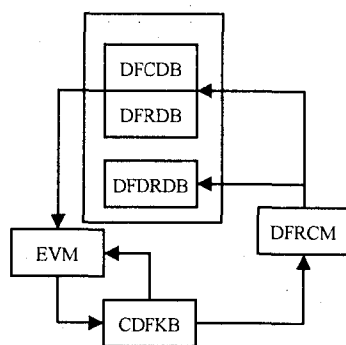


图 2 缺省假设推理的框架

\* DFCDB: 动态模糊(DF)概念库, 存储 DF 概念。

\* DFRDB: DF 规则库, 存储 DF 规则。

\* DFDRDB: DF 缺省规则库, 存储 DF 缺省规则。

\* CDFKB: 当前 DF 知识库, 存储了当前推理过程中的信息。

\* EVM: 评估机制。它需要触发条件。

\* DFRCM: DF 推理控制机制。控制推理的过程。

基本思想和过程:

(1) 当前知识库 CDFKB 向 DF 推理控制机制 DFRCM 提交知识。

(2) DFRCM 查找 DFCDB 和 DFRDB, 若没有规则或概念可匹配, 则检查 DFDRDB。在进行规则匹配时调用动态模糊匹配(DFMA)算法。

(3) 若是通过 DFDRDB 得出的结论, 需要经过评估机制 EVM 进行评估, 若发现它与当前知识库产生矛盾则认为该结论不合理。

(4) 对“X is A”的知识, 若 DFCDB 和 DFRDB 中没有可以判断“X is not A”的概念或规则, 但是 DFRDB 中有规则是以 A 作为前提的, 则先假设“X is A”, 得出结论后同时激活 EVM 进行评估。

### 2.2 DF 知识的描述

定义 7 设  $M$  为 DF 命题。在系统中  $M$  表示为:  $A(x)$  is  $D, (\vec{a}, \vec{a})$ 。记为  $M'$ 。其中  $A$  表示 DF 属性,  $x$  表示 DF 对象,  $D$  为属性的状态,  $(\vec{a}, \vec{a})$  为属性状态的 DF 度且  $(\vec{0}, \vec{0}) \leq (\vec{a}, \vec{a}) \leq (\vec{1}, \vec{1})$ 。

定义 8 设动态模糊谓词  $P$  表示的是对象之间的关系。 $P$  可表示为:  $B(x, y)$  is  $F, (\vec{b}, \vec{b})$ 。记为  $P'$ 。其中  $B$  表示对象  $x$  和  $y$  的关系,  $F$  是 DF 关系的状态,  $(\vec{b}, \vec{b})$  为关系状态的 DF 度且  $(\vec{0}, \vec{0}) \leq (\vec{b}, \vec{b}) \leq (\vec{1}, \vec{1})$ 。

对应与产生式系统 IF  $M_1$  and  $M_2, \dots, M_n$  THEN  $Q_1, Q_2, \dots, Q_n, CF$ 。其中  $CF$  为规则强度。则 DF 产生式系统为: IF  $M_1'$  and  $M_2', \dots, M_n'$  THEN  $Q_1, Q_2, \dots, Q_n, (\vec{CF}, \vec{CF})$ 。其中  $(\vec{CF}, \vec{CF})$  为 DF 规则的 DF 信任测度, 且  $(\vec{CF}, \vec{CF}) \in [(\vec{0}, \vec{0}), (\vec{1}, \vec{1})]$ 。

对于出现的知识和 DF 规则前件的 DF 匹配度有如下

定义:

定义 9 设新出现的事实  $E$  的 DF 状态为  $(\vec{t}, \vec{t})$ , 相应的 DF 规则前件的 DF 状态为  $(\vec{a}, \vec{a})$ 。那么它们的 DF 匹配度记为 MDF, 且

$$\text{MDF} = \begin{cases} (\vec{1}, \vec{1}) & (\vec{t}, \vec{t}) \geq (\vec{a}, \vec{a}) \\ (\vec{t}, \vec{t}) / (\vec{a}, \vec{a}) & (\vec{t}, \vec{t}) < (\vec{a}, \vec{a}) \end{cases}$$

定理 1  $\text{MDF} \in [(\vec{0}, \vec{0}), (\vec{1}, \vec{1})]$

证明: 设  $(\vec{t}, \vec{t}) \geq (\vec{a}, \vec{a})$ , 则由定义 9 可得  $\text{MDF} = (\vec{1}, \vec{1})$ 。反之, 设  $(\vec{t}, \vec{t}) < (\vec{a}, \vec{a})$ , 则  $\text{MDF} = (\vec{t}, \vec{t}) / (\vec{a}, \vec{a}) \leq (\vec{1}, \vec{1})$

当  $(\vec{t}, \vec{t}) = (\vec{0}, \vec{0})$  时,  $\text{MDF} = (\vec{t}, \vec{t}) / (\vec{a}, \vec{a}) = (\vec{0}, \vec{0})$ , 所以  $\text{MDF} \in [(\vec{0}, \vec{0}), (\vec{1}, \vec{1})]$ 。

说明: 当  $(\vec{t}, \vec{t})$  为 DF 关系状态 DF 度时,  $(\vec{a}, \vec{a})$  也就是相应的 DF 关系状态 DF 度。反之, 当  $(\vec{t}, \vec{t})$  仅为 DF 对象属性的 DF 度时,  $(\vec{a}, \vec{a})$  也就是相应的 DF 度。因为由  $M'$  与  $P'$  的定义很容易区分  $(\vec{t}, \vec{t})$  是关系 DF 度还是属性的 DF 度。

对于多个对象之间的关系, 可以对定义 9 进行扩展, 即  $B(x_1, x_2, \dots, x_n) \text{ is } F, (\vec{b}, \vec{b})$ 。

对于命题或谓词间的真值计算, 即算子  $\wedge$ 、 $\vee$ 、 $\neg$  (非) 的运算以及量词  $\exists$  和  $\forall$  的变换遵循动态模糊逻辑的定义, 见上文。

### 2.3 相关算法描述

本节针对前面所描述的框架结构给出相应的算法。

#### (1) 算法 DFMA。

该算法是求出规则匹配时的 DF 匹配度。

Procedure DFMA ( $E, (\vec{t}, \vec{t}), n$ )

//  $n$  is the number of rules in knowledge base //

begin

for  $i = 1$  to  $n$  do

if  $E$  matching  $i$ . rule

{if  $i$ . rule is default rule in DFDRDB

then flag = 1 else flag = 0

end if

if  $(\vec{t}, \vec{t}) < i. (\vec{a}, \vec{a})$  then

$\text{MDF} = (\vec{t}, \vec{t}) / i. (\vec{a}, \vec{a})$  return MDF

else  $\text{MDF} = (\vec{1}, \vec{1})$

return MDF

End if

}

Next  $i = i + 1$

End begin

End procedure

DFMA 算法的时间复杂度和知识库中的规则个数有关, 即为  $O(n)$ 。

#### (2) DF 推理控制机制 DFRCM 的算法描述。

Procedure DFRCM ( $E, (\vec{t}, \vec{t}), n$ )

begin

If  $E$  exists in DFCDDB then return  $E$ ; add to knowledge base

Else

If  $\text{DFMA}(E, (\vec{t}, \vec{t}), n) > S$

If flag = 1 then  $\text{EVM}(E)$

else add to knowledge Base.

End if

End if

End if

End Begin

End procedure

算法分析: 该算法调用了 DFMA 算法, 而且由以下 EVM 算法可以知其时间复杂度为  $O(2n)$ , 即  $O(n)$ 。

#### (3) 评估机制 EVM 的算法描述。

Procedure EVM( $E$ )

begin

Check  $E$  in knowledge base

If don't conflict then add it to knowledgebase

else give up the result

End if

End begin

End procedure

算法分析: 该算法主要是逐个查找知识库中的知识是否矛盾, 所以最坏情况为  $O(n)$ 。

### 2.4 一致性讨论

为了保证知识库的一致性, 采用了评估机制 EVM。但是仅采用上述的算法 EVM 还不能完全保证知识库的一致性。因此对于由缺省规则或由 2.1 中的 (4) 得出来的结论在算法 DFMA 和 DFRCM 中制相应的标记, 在 EVM 评估中发现新得出的结论和相应标记的结论矛盾时, 删除相应标记的结论。因此由相应标记的知识得出的新的知识也要做标记, 这样若存在矛盾则可以根据标记加以回溯。

### 3 结 论

文中在 DFL 的基础上给出了缺省假设推理的结构描述, 把系统分为了动态模糊推理规则库和动态模糊缺省规则库, 可以进行单调推理和非单调推理, 使推理效率更高。通过设置评估机制 EVM 对缺省假设的推理结果进行了评估, 从一定程度上保持了知识库的一致性, 并且为动态模糊知识的表示提供了一种方法。

#### 参考文献:

- [1] Jonker C M, Treur J. Modelling the dynamic of reasoning process: Reasoning by assumption[J]. Cognitive System Research, 2003(4): 119 - 136.
- [2] Peréz - Silva J L, Lara - Rosano F. Dynamic fuzzy logic [C]//In: Mexican International Conference on Artificial Intelligence Acapulco. Mexico: [s. n.], 2000: 611 - 670.

(下转第 53 页)

事务协调 DIAgent、子事务 DIAgent 等的设计。

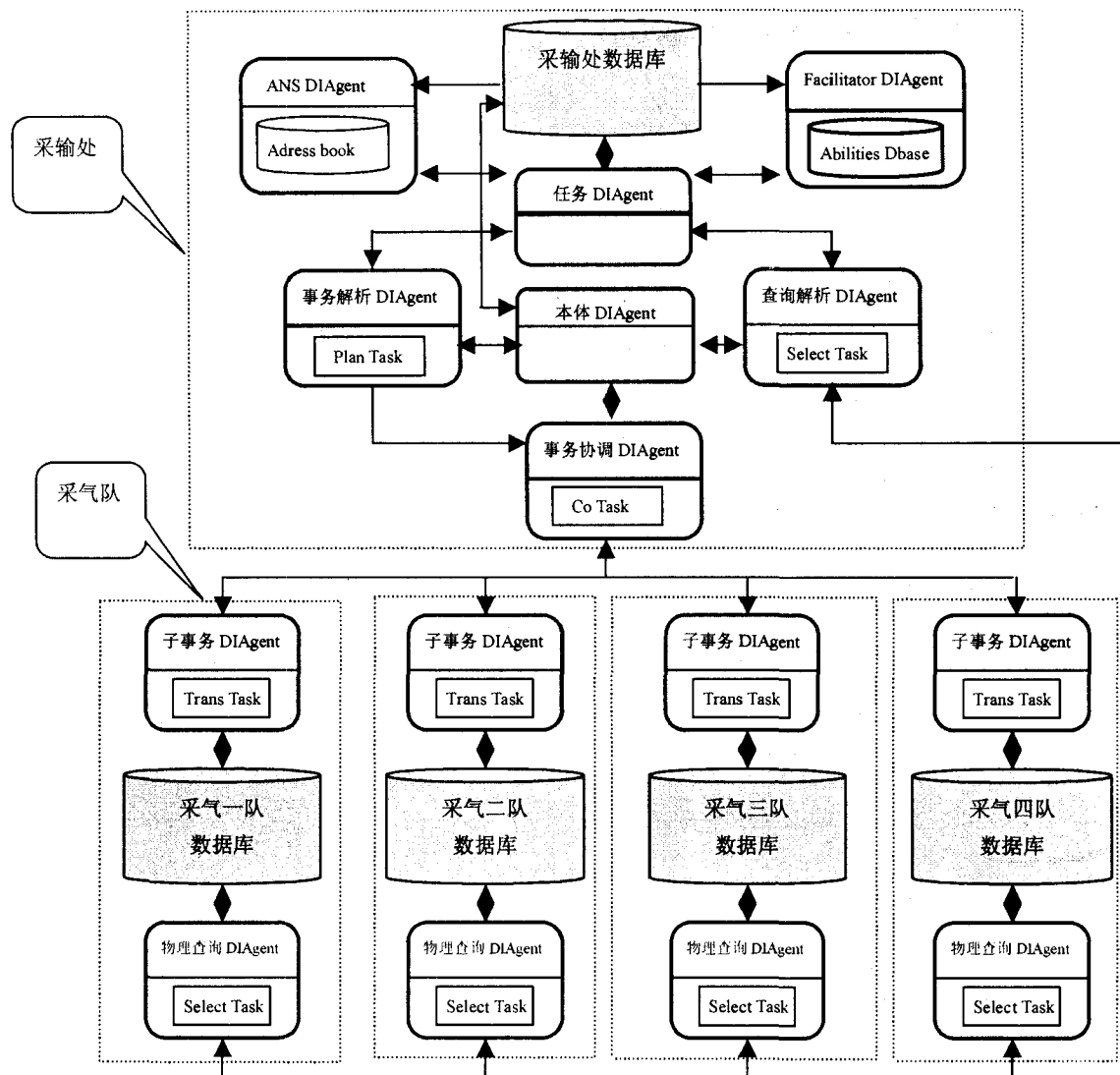


图 3 多 Agent 系统结构

## 5 结 论

文中笔者的创新观点是:通过研究多 Agent 技术与分布式数据库技术的特点,结合课题,提出了用多 Agent 技术集成现有数据库为一个具体全局模式的分布式数据库,为企业提供了新的数据库应用平台。通过文中的分析可以看出:所提出的集成方法为数据库系统集成提出了一个新思路,虽然目前还没有得到普及,但其本身所具有的优越性和强大生命力决定了它的发展前景必然十分广阔。

### 参考文献:

[1] Carver Z, Cvetanovic Z, Lesser V. Sophisticated cooperation in

FA/C distributed problem solving systems: Proceedings of the 9th national conference on Artificial Intelligence[C]. Anaheim, CA: [s. n.], 1991: 191-197.

[2] Bond A H, Gasser L. An analysis of problem and research in Distributed Artificial Intelligence[C]//Bond A H, Gasser L. Reading in Distributed Artificial Intelligence. San Mateo, CA: Morgan Kaufmann, 1995.

[3] 李 英. 多 Agent 系统及其在预测与智能交通系统中的应用[M]. 上海: 华东理工大学出版社, 2004: 31-33.

[4] 萨师煊, 王 珊. 数据库系统概论[M]. 第 3 版. 北京: 高等教育出版社, 2000.

[5] 毛国君. 高级数据库原理与技术[M]. 北京: 人民邮电出版社, 2004.

(上接第 49 页)

[3] Yoshida Y. A continuous-time dynamic fuzzy systems, (1): A limit theorem[J]. Fuzzy sets and systems, 2000, 113(3): 453-460.

[4] 李凡长, 刘贵全, 余玉梅. 动态模糊逻辑引论[M]. 昆明:

云南科技出版社, 2005.

[5] Reiter R. A logic for default reasoning[J]. Artificial Intelligence, 1980(13): 81-132.

[6] 张继负, 刘 静. 表示和处理缺省知识的多维模糊推理方法和算法[J]. 小型微型系统, 2001, 22(7): 893-896.