正确性维护系统 知识库维护系统 开放逻辑

计算机科学 1995 Vol. 22 №. 1

10-13

基于 ATMS 的知识库维护系统与开放逻辑的实现*)

方思行 凌卫新 陆子强 TP18

(华南理工大学应用数学系 广州 510641)

摘 要 In this paper, a novel ATMS-based knowledge base maintenance system is proposed. It enhances functions of the tranditional knowledge base maintenance systems as it is a multiple context, open and non-monotonic system, and possesses capacities of describing and characterizing knowledge increment, knowledge update and scientific discoveries. The relation of the open logic and this system is also discussed. Both of them aim at the computer simulation of higher mental processes on theoretical foundation and implementation respectively

关键词 Higher mental processes, Open logic, ATMS, Knowledge base maintenance system

一、引言

传统的形式系统的研究由于缺乏与外部环境进行交互的手段和方法,造成了它的封闭性,因此无法模拟人类的高级思维过程。其基本原因在于这种研究背离了认识论的科学规律,忽视了人类实践的重要环节。

最近,我国学者提出了"开放逻辑"[1]和"思维计算"[2]的新概念;以认识论为指导,以高级思维过程模拟为目标,以逻辑方法为手段,突破了人工智能的现有框架,为高级思维过程的计算机模拟奠定了逻辑基础。

受到上述工作的启发,作者将 ATMS[3-4],推广应用到知识库的管理方面,设计了一种新的知识库维护系统,在多上下文、开放性和非单调性等方面拓广了传统的知识库管理系统的功能,并且提供了描述与刻画知识增长、更新和科学发现的能力。本文将介绍这个系统所采用的数据结构、维护过程和推理机制,最后将它与开放逻辑作一个简单的比较,说明它实际上是开放逻辑的一种具体实现方法。

二、ATMS 概述

人们对现实世界的知识总是不完备的,当人们进行推理的时候,往往要进行种种假设,或者就把已有的一部分知识作为假说(即假设的焦合)。这种基于假设的推理具有非单调性的特点。当系统通过逻辑推理或与外部环境的交互作用获得新的知识之后,假设以及基于它的推理结论可能要被撤消,因而需要修改已有的假设,或者增加新的假设。提出假设与修改假设是人类思维的一个重要特征,作为模拟人的智能的智能系统也应该具有这样的能力。作者认为,ATMS 正是在智能系统中用来描述与刻画知识增长、更新以及假说进化的理想工具。

ATMS 是由 de Kleer 在 1986 年提出来的一种基于假设的正确性维护系统。它通常与问题求解器一起使用,每当收到问题求解器传递的一个新的信息,便增量地计算每个数据所依赖的假设。它是一个多上下文系统,不仅基于证实的控制,而且基于假设集的控制。为了说明可以怎样将 ATMS 推广应用到知识库管理中去,首先介绍几个基本概念。

2.1 ATMS 节点

ATMS 的基本数据结构是节点,由三部分组成,问题求解器的数据、用来表示数据成立的假设条件的标记和支持这个数据的证实,其形式如下:

$\gamma_{\text{数据}}$:(数据,标记,证实)

在 ATMS 中,数据被认为是原子,ATMS 不能访问问题求解过程中的语义。每个证实都是由用于支持数据的推断中的前项所组成,而推断是由问题求解器提供的。标记是一个环境集,其中每个环境代表着允许那个数据被推出的假设集,是由 ATMS 建立的。标记和证实都在析取范式下成立,而每个环境和每个证实则是一个合取,只有它们当中的所有元素均为真时,对应的数据才能被那个环境所支持或者被那个证实所推出。

2.2 节点类型

ATMS 中共有四种不同的节点,即假设(assumption)、前提(premise)、假定节点(assumed node)和导出节点(derived node)。

定义 2.1 如果一个节点的标记环境和证实中只含有假设数据本身,则这个节点称为一个假设。

*)国家自然科学基金资助项目

例如，如果$\neg P$是一个假设，则在ATMS中记为：$\langle \neg P, \{\{\neg P\}\}, \{(\neg P)\}\rangle$。

定义 2.2 前提是指在论域中永远成立的数据，既不依赖于任何假设，也没有任何证实。

例如，如果$P \vee Q$是一个前提，则在ATMS中记为：$\langle P \vee Q, \{\{\}\}, \{()\}\rangle$。即前提的标记是一个空环境(注意：不是空标记!)，同时有一个不含任何前项的证实。

定义 2.3 如果一个节点既不是前提，也不是假设，并且它的证实中含有一个假设，则这个节点称为假定节点。

例如，推断$(P \vee Q) \wedge \neg P \rightarrow Q$给出一个假定节点：$\langle Q, \{\{\neg P\}\}, \{(P \vee Q, \neg P)\}\rangle$。这里，证实表明数据Q是由$P \vee Q$与$\neg P$推导出来的，而标记则表明Q归根结底依赖于假设$\neg P$。

定义 2.4 如果一个数据依赖于某些假设，但是这些假设在该数据的证实表中并不出现，则与这个数据对应的节点称为导出节点。

例如，推断$Q \rightarrow A$给出一个导出节点：$\langle A, \{\{\neg A\}\}, \{(Q)\}\rangle$。这里，A的证实中只含有Q(它不是假设)，但是，我们可以推知，A归根结底也是依赖于假设$\neg P$。

现在假定有一个新的推断：$(A \vee B) \wedge \neg B \rightarrow A$。其中，$A \vee B$是前提，$\neg B$是假设，则原先的导出节点A将变为：$\langle A, \{\{\neg P\}, \{\neg B\}\}, \{(Q), (A \vee B, \neg B)\}\rangle$

2.3 nogoods

系统通过逻辑推理或者与外部环境的交互作用，常常会获知某个数据或某些数据的合取导致矛盾(失败)，因此，我们可以在ATMS中预先给出一个矛盾节点，它实际上是一种特殊的导出节点。例如，如果发现2.2节中的A导致矛盾，即$A \rightarrow \perp$，则有导出节点：

$$\gamma_\perp : (\perp, \{\}, \{(A)\})$$

需要注意的是，本来$\gamma_\perp$的标记应为$\{\{\neg P\}, \{\neg B\}\}$而不是$\{\}$，但是这些导致矛盾的不相容假设集(环境)将会在系统中反复被使用，所以，在ATMS中专门设置了一个*nogoods*数据结构，用来存放系统中所有最小的不相容假设集，它们中的每一个都叫做一个nogood。一个节点标记中的每个环境都不得不含有任何一个nogood，否则该环境应予删除。同时，一旦系统发现有一个nogood是空集，就会给出提示，告诉用户已发现一个不依赖于任何假设集的矛盾，这实际上暗喻系统的前提是不相容的。

2.4 ATMS的基本运算

ATMS的基本运算包括：为问题求解器的数据建立ATMS节点，给节点增加证实以及更新节点标记。为此，我们定义了四个谓词，它们可以作为问题求解器与ATMS的接口：

(1)建立前提

```
premise(premise_name) :-
    put_new_node(premide_name,{{}},{()}).
```

(2)建立假设

```
assumption(assumption) :-
    put_new_node(assumption,{{assumption}},
        {(assumption)}).
```

(3)给节点增加证实

```
justify(antecedents,consequent).
```

(4)发现矛盾(失败)

```
nogood(antecedents).
```

(这个谓词可看作是(3)的一种特殊情况)

因为(3)和(4)的定义涉及比较复杂的运算，限于篇幅，我们没有给出详细定义，下面通过一个例子加以说明。为ATMS设计的基本算法保证在每次基本运算之后，每个节点的标记就目前给出的假设与证实而言都是相容的、健全的、完全的和最小的，有关这些概念的描述，请参阅文[4]。

现在假定ATMS已有如下节点(为节省篇幅，省略了节点中的证实部分)：

$\langle y=1, \{\{A,B\},\{C,D\},\{K\}\}, \{\cdots\}\rangle$
$\langle z=1, \{\{J,K\},\{L,Z\}\}, \{\cdots\}\rangle$
$\langle x=y+z, \{\{J\}\}, \{\cdots\}\rangle$

并且，nogoods$=\{\{A,B,L\}\}$。这时，问题求解器通过谓词justify向ATMS传递一个证实：

$$(y=1) \wedge (z=1) \wedge (x=y+z) \rightarrow x=2$$

因为数据$\gamma_{x=2}$原先在ATMS中未有相应的节点，所以ATMS首先为其建立一个新节点，然后计算它的标记。这个计算过程可以用一个集合运算来描述：

设节点n有m个证实，j_{ik}表示其中第k个证实中第i个节点的标记，则节点n的一个完全的标记为：

$$\bigcup_i \{x \mid x = \bigcup_i x_i, \text{其中 } x_i \in j_{ik}\}$$

当我们从上述标记中将不相容的环境(即所有nogood及其超集)以及包含其他环境的超集环境删除之后，这个标记便是相容的、健全的和最小的了。

回到上述例子中去，可以得到$\gamma_{x=2}$的一个完全的标记为：

$\{\{A,B,J,K\},\{A,B,L,Z,J\},\{C,D,J,K\},\{C,D,L,Z,J\}\{J,K\},\{K,L,Z,J\}\}$

将包含环境$\{J,K\}$的三个超集环境删去之后，便得到一个最小标记：

$\{\{A,B,L,Z,J\},\{C,D,L,Z,J\},\{J,K\}\}$

但它不是相容的。因为环境$\{A,B,L,Z,J\}$包含nogood环境$\{A,B,L\}$。这意味着这个环境会导致矛盾(失败)，所以必须将其删除。于是，我们得到一个相容的、健全的、完全的和最小的标记：

$\{\{C,D,L,Z,J\},\{J,K\}\}$

最后，在ATMS中增加一个新节点：
(x=2,{{C,D,L,Z,J},{J,K}},{(y=1,z=1,x=y+z)})

给一个已有的ATMS节点(例如A)增加新的证实的过程稍微要复杂一点。其原因主要是因为，如果另一个节点(例如B)的证实中含有节点A，则当节点A的标记发生变化时，节点B的标记也必须随之更新，如此类推，这个递归过程将一直进行下去。但是，它必定会收敛，因为问题中假设的个数只有有限个。

2.5 扩展与解释

上下文(context)是指ATMS中的一个数据集，由一个相容环境中的那些假设以及这些假设推出的所有节点所组成。这些假设(标记环境)刻画了上下文的特征，所以，在ATMS中一般并不明确地计算上下文。因为如果一个节点存在于某个上下文，那么它也将存在于该上下文的每个超集中(排除不相容的超集)，所以对于每个节点，ATMS只是记录它所在的最大下界上下文(实际上是标记)。类似地，在ATMS中也只是记录不相容环境的最大下界(nogoods)，通过简单的子集测试，就可以知道一个数据是否在某个环境中成立，或者为了使某个数据成立，需要作什么假设。ATMS的有效性和高效率正是建立在这个原理的基础上。

在问题求解或者科学发现中，一个关键的问题是如何从现有的知识出发，产生所有可能的解或者推出以前未知的新知识。在理由维护系统(RMS)中是通过扩展(extension)这个重要概念来描述的。扩展是RMS数据库中最大的相容数据子集。这里的最大性是指，当该子集中增加任何一个新数据时，都必然会导致不相容。

定义2.5 在ATMS中，扩展是由一个最大的相容假设集所推出的上下文。

定义2.6 刻画扩展的最大相容假设集叫做一个解释(interpretation)。

显然，扩展与解释是一一对应的。因此，扩展就是由其相应的解释所生成的上下文。

在Reiter的缺省逻辑[5]中，定义一个假设A相当于生成一个规范缺省：

$$:MA/A$$

所有假设与证实的集构成一个命题规范缺省理论。Reiter把问题的一个解(即最大的上下文)定义为一个扩展。

但是，严格地说，一个ATMS扩展并不完全等同于一个缺省逻辑扩展，因为缺省逻辑扩展包括所有可能被推出的合式公式，而ATMS扩展只包括其ATMS节点已经存在的所有可推出的原子。因此，缺省逻辑扩展总是它所对应的ATMS扩展的超集。如果问题求解器要获取具体的一个合式公式，那么它就必须在ATMS中明确地为每个缺省建立一个假设节点。

2.6 扩展的生成

计算扩展的目的在于确定相容的最大数据集。为此，首先介绍关于扩展的几个性质。

定理2.1 所有前提和由前提推出一切数据必在任何扩展中。

定理2.2 如果一个假设不属于任何的nogood，则它必定在任何的扩展中。

定理2.3 如果ATMS的nogoods为空，则它只有一个扩展；这时，在ATMS中所表示的理论是相容的，并且可以用所有已经作出的假设来刻画它。

以上三个定理可用来对扩展的生成算法进行优化。

定理2.4 设Γ为一假设集，则它是相容的充分必要条件是Γ中不含任一nogood中的至少一个假设。

根据定理2.4，我们给出生成所有解释的一种算法。为简单计，仅以下面的例子加以说明，并且暂不考虑前提的存在。

假定ATMS所有假设的集为Γ={A,B,C,D,}，以及nogoods={{A,B},{B,C}}。从每个nogood中取一个元素，并作所有可能的组合，得到一个候选人(candidate)集：{A,B},{A,C},{B},{B,C}。然后分别取这些候选人在Γ中的补，得{C,D},{B,D},{A,C,D},{A,D}。根据定理2.4，上述每个假设集均是相容的。因为{C,D},{A,D}都是{A,C,D}的子集，所以将它们删去。于是，得到ATMS的两个最大相容假设集，即解释：{B,D},{A,C,D}。最后，我们只要将每个

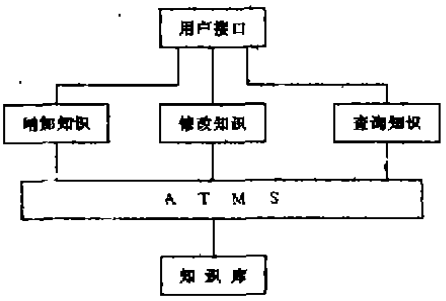

图1 知识库维护系统的结构

数据的标记与上述各个解释分别进行简单的子集测试，就可以生成相应的扩展。

三、一种基于 ATMS 的知识库维护系统

我们已经将 ATMS 推广应用于知识库的维护。下面是这种知识库维护系统的组织结构。

(1)增加知识：这个模块提供了三个谓词，分别用于增加前提(premise)、增加假设(assumption)与增加证实(justify)，其定义见2.4节。根据用户的要求，通常的事实可以作为前提，也可以作为假设；而规则可以作为假设，也可以作为证实。

(2)修改知识：如果通过逻辑推理、科学实验等手段发现某些数据导致矛盾(失败)，则调用此模块的谓词 nogood 来生成 nogoods。该谓词的变元不必限于假设，而可以是任意数据的合取。

当执行(1)和(2)的时候，都必须与 ATMS 进行交互作用，对 ATMS 的节点进行更新。

(3)查询知识：通过一种专门的知识查询语言，用户可以进行下述四种主要的查询：①给定环境，查询有效数据或某数据是否成立；②给定数据，查询它成立的环境；③查询当前的最大相容假设集(解释)；④查询当前的最大相容数据集(扩展)。

与传统的知识库维护系统不同的是，该知识库维护系统并不要求整个知识库是相容的，知识库是多上下文的；同时它是开放式的，通过逻辑推理或与外部的交互作用，每个数据的信念可以不断更新和变换，这种做法能够更好地模拟人的思维，也更加符合实际的需要。

四、开放逻辑与 ATMS

李未的开放逻辑旨在真实地描述客观世界的开放性，对论域不加任何限制，它所作的一切都是根据实践结果来更新假说，我们认为，ATMS 正是体现了这些特点，因此它是开放逻辑的一种理想的实现方法；反过来，开放逻辑将假说生成过程数学理论化，从而也为 ATMS 奠定了逻辑基础。

不难看出，开放逻辑中的几个重要的基本概念，在 ATMS 中都可以找到相应的实现方法。例如，"新假设"对应于"前提"与"假设"；"理想事实反驳"对应于"nogoods"；"可接受假设"对应于"解释"；"R-重构"对应于"扩展"等等。为了说明这一点，我们将文[1]中关于从伽利略的相对性原理发展到爱因斯坦的狭义相对论的描述，利用 ATMS 重新加以描述.

令 B(x)表示 x 是一物体，A(x)表示 x 满足伽利略的速度相加原理，Ω 表示狭义相对论以前积累的力学知识。于是，我们有两个假设：

推论	标记
(1)assumption(Ω)	{{Ω}}
(2)assumption(¬B(x)∨A(x))	{{¬B(x)∨A(x)}}

大量的实验已证实光 c 具有粒子性，因此，我们有一个前提：

(3)premise(B(c))	{{ }}

从(2)与(3)可以形式地推出 A(c)：

(4)justify((¬B(x)∨A(x),B(c)),A(c))	{{¬B(x)∨A(x)}}

但是，科学实验与天文观测都支持 A(c)的反面¬A(c)，更精确地说，是光速不变。所以，我们又有一个前提：

(5)premise(¬A(c))	{{ }}

这时，我们发现 A(c)与¬A(c)是不相容的，即

(6)nogood(A(c),¬A(c))	{¬B(x)∨A(x)}

于是，nogoods 中含有不相容的假设集{¬B(x)∨A(x)}。据此，对 ATMS 中各节点进行更新。受影响的是(2)和(4)，它们的标记变为空集，这暗喻"任何物体都满足速度相加原理"和"光子满足速度相加原理"已无任何假设支持。不难计算，这时只有一个解释{Ω}。考虑到现在有两个前提，于是得到一个扩展{Ω,B(c),¬A(c)}，这就是爱因斯坦的狭义相对论(注意：当上述的"前提"换成"假设"时，得到的解释不同，但扩展是相同的)。

五、结论

模拟人的高级思维过程是智能系统的根本目的。开放逻辑和本文所提出的基于 ATMS 的知识库维护系统都是在这方面所作的努力。这种新型的开放式知识库维护系统推广了传统的知识库管理系统的能力，提供了描述与刻画知识增长、更新和科学发现的一种实现方法，我们相信，它将会在问题求解、专家系统、数据库管理、机器学习、知识获取等方面发挥重要作用。

参考文献

[1] 李 未，一个开放的逻辑系统，中国科学(A辑)，1992，第10期，1103-1113

[2] 洪家荣，知识进化和高级思维过程的逻辑基础，计算机科学，1993，20卷6期，8-16

[3] De Kleer，J.，An assumption-based TMS，Artificial Intelligence，1986，28，127-162

[4] 方思行，理由维护系统的研究，计算机科学，1992，19卷2期，6-14

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