缺言推理系统GKD-DIS

杨莉 胡守仁

(长沙国防科技大学计算机系)

DEFAULT INFERENCE SYSTEM GKD-DIS

Yang Li and Hu Shouren

(*Department of Computer Science, National University of Defence Technology, Changsha*)

ABSTRACT

That how to enable knowledge base to have the function of inference and maintenance automatically has been a key problem, which existed in the design of the knowledge base that wants to be practical and effective. In this paper, we propose an automatical knowledge base inference and maintenance method based on default inference at first, and then we introduce a knowledge base inference and maintenance system, which has been implemented on VAX-11/780 machine in GKD-Prolog.

摘 要

如何使知识库具有自动推理和维护的功能，已成为知识库能够达到有效和实用所面临的一个关键问题。本文首先提出了一种基于缺言推理的知识库自动推理和维护的方法，然后介绍了根据此方法，我们在VAX-11/780 上使用GKD-Prolog 作为工具实现的一个知识库推理和维护系统。

§1. 引 言

目前，几乎所有的人工智能系统都具有一个庞大的知识库，如何保证知识库具有无冗余性、一致性，如何使知识库具有自动推理和维护能力已成为人工智能系统达到高效和实用所面临的一个关键问题。

传统的知识库系统是建立在谓词逻辑基础上的单调系统，即知识库中已知为真的知识数目随时间而严格增加。然而，由于人对客观世界的认识不仅是不确定的，而且往往也是不完全的，因此，对于基于人类知识的智能化知识库系统来说，它所拥有的知识都应是暂时过渡的，还需要不断进行改进和完善。根据上述观点，本文给出一种基于缺言理论来实现知识库自动推理和维护的方法。

1989 年11 月收到，1990 年2 月21 日定稿。

§2. 基本概念及定义

一个缺言理论定义为二元组$\Delta = (D, Q)$，其中D 是应用于某个模型化世界的缺言规则集，Q 是该世界的知识集，亦即一阶公式集。缺言规则的一般形式表示为:

$$\frac{\alpha(x) : MB_1(x), \cdots, MB_n(x)}{w(x)} \tag{1}$$

其中$\alpha(x), \beta_1(x), \cdots, \beta_n(x)$ 和$w(x)$ 都是一阶公式，其自变量$x = x_1, \cdots, x_m$。(1) 式的含义为: "对所有个体$x_1, \cdots, x_m$，如果$\alpha(x)$ 被相信，且$\beta_1(x), \cdots, \beta_n(x)$ 与知识系统Q 相容，则$w(x)$ 可以被相信。这里，我们称被Q 所相信的知识为Q 的信条。例如: bird (x): fly (x)/fly (x) 是一个规则，则第二个fly (x) 是一个信条。在(1) 式中$\alpha(x)$ 称为前提，$\beta_1(x), \cdots, \beta_n(x)$ 称为合理性条件，而$w(x)$ 称为结论。这里合理性条件只是一种假设，也就是说，$\beta(x)$ 成立并不导致当前的Q 出现矛盾。在这里，我们说"当前"是有意义的，因为在用合理性条件进行推理时，以后得出的结论可能会使Q 出现矛盾。为了在我们的系统中能用缺言理论来实现知识库的自动推理和自动维护，我们给出几个基本定义和约定:

定义1　事实和规则统称为信条(即知识)。每一信条又对应一个结点，用n_i 表示$i = 1, 2, \cdots$，用N 表示系统中所有结点的集合。

定义2　每一结点都有一状态status 以标记该结点当前是否为可信，status=In, 相信为真，status=Out, 不相信为真。此外，每个结点都对应一个证实表JL，其中每一证实J 形式为: (Inlist, Outlist)，它为有效的，仅当Inlist 表为空或当Inlist 中每一个结点的状态均为In, 且Outlist 表为空或者Outlist 表中的每一个结点均为Out。In 结点是指那些当前至少有一个有效证实的结点，反之为Out 结点。

定义3　如果一个结点的Inlist 表或Outlist 表不为空，则此结点表示一个假设，否则此结点表示一个前提。如果一个假设h 的Outlist 表为空，则称此假设为单调假设，否则称为非单调假设。

定义4　系统中信条的当前可信集定义为:

$S_{cr} = \{n_i | status(n_i) = In, n_i \in N\}$

当前不可信集定义为:

$S_{ncr} = \{n_i | status(n_i) = Out, n_i \in N\}$

定义5　对某一结点n, 设其状态status (n)=In, 则必$\exists J_m \in JL_n$, 使得status (n)=In, 称J_m 为结点n 的支持证实。

定义6　结点n_i 的支持结点集$S_{sn}(n_i)$ 定义如下:

如果$n_i \in S_{cr}$, 且n_i 的支持证实为P_j, 则

$S_{sn}(n_i) = \{n_k | n_k \in Inlist\ P_j$ 或$n_k \in Outlist\ P_j\}$

如果$n_i \in S_{ncr}$, 且$JL_{n_i} = \{P_1, P_2, \cdots, P_k\}$, 则$S_{sn}(n_i) = \{n_j | P_j \in JL_{ni}, n_j \in Inlist\ P_j$ 且 status $(n_j) = Out$, 或$n_j \in Outlist\ P_j$且 $status(n_j) = In, j = 1, 2, \cdots, k\}$

定义7　结点n 的支持结点关系为:

$R_{sn}(n) = \{< n, n_j > | n_j \in S_{sn}(n)\}$

如果$N = \{n_1, n_2, \cdots n_k\}$, 则系统支持结点关系为: $R_{ssn}(N) = \bigcup_{i=1}^{k} R_{sn}(n_i)$, 系统支持结点关系的传递闭包为: $R^+_{ssn}(N) = \bigcup_{i=1}^{k} R^i_{ssn}(N)$。我们定义结点$n$ 的祖先集为:

$S_{ancent}(n)=\{n_j|<n,n_j>\in R^+_{isn}(N)\}$

定义8 结点n的推论集定义为:

$S_{con}(n)=\{n_k|\exists P_i\in JL_{nk}$,且$n\in Inlist\ P_j$或$n\in Outlist\ P_j\}$

其推论关系为: $R_{con}(n)=\{<n,\ n_j>|n_j\in S_{con}(n)\}$

设$N=\{n_1,n_2\cdots,n_k\}$, 则系统的推论关系为: $R_{con}(N)=\bigcup_{i=1}^{k}R_{con}(n_i)$

定义9 设$N=\{n_1,n_2,\cdots,n_k\}$, 系统推论关系的传递闭包定义为: $R^+_{con}(N)=\bigcup_{i=1}^{k}R^i_{con}(N)$, 我们定义结点$n$的反应集为$S_{reply}(n)=\{n_j|<n,n_j>\in R^+_{con}(N)\}$。

约定1 冗余的知识有下面两种情况:

I: 两条一模一样的事实或两条一模一样的规则。

II: 两条规则之间有包含性关系如:

$R_1:e_1\wedge e_2\wedge\cdots\wedge e_m\rightarrow e$

$R_2:e'_1\wedge e'_2\wedge\cdots\wedge e'_n\rightarrow e$, 当$e_i=e'_i$且$m\neq n$时

如果$m<n$, 则R_2为一条冗余的规则。

约定2 矛盾的规则有下述两种情况:

I: 条件相同, 但结论矛盾, 如:

$R_1:a_1\wedge a_2\wedge\cdots\wedge a_n\rightarrow a$ 与

$R_2:a_1\wedge a_2\wedge\cdots\wedge a_n\rightarrow\neg a$ 相矛盾

II: 条件部分除开含有互为矛盾的条件子句外的其余部分以及结论部分均相同, 如:

$R_1:a_1\wedge a_2\wedge\cdots\wedge a_r\wedge a\rightarrow b$ 与

$R_2:a_1\wedge a_2\wedge\cdots\wedge a_r\wedge\neg a\rightarrow b$ 相矛盾

约定3 环路规则: 规则中的事实可循环导出, 如:

$R_1:a_1\wedge a_2\wedge\cdots\wedge a_m\rightarrow a'$

$R_2:a'\wedge a'_2\wedge\cdots\wedge a'_m\rightarrow a''$

$R_3:a''\wedge a''_2\wedge\cdots\wedge a''_m\rightarrow a_1$

§3. 系统主要设计思想及运行实例

在我们所研制的知识库系统GKD-DIS 中, 对应结点只有简单命题[事实表示为$P(x_1,\cdots,x_n)$, 规则$P\leftarrow Q_1,\cdots,Q_m$, 转换为giantoa $(P,Q_1,\cdots,Q_m)$], 且每一结点都有一状态来标记该命题当前是否为可信。这样不同于任何单调系统, 在任何时候系统中既存在可信命题也存在不可信命题。系统中命题的状态是随时间而不断变化的, 新的事实、规则的加入可能使以前为可信的命题变为不可信, 那么一切基于这个已变化命题的命题都要受到影响, 需要得到修正。

系统所实现的自动推理和维护过程的主要步骤如下:

1. 把命题A 连同它的证实J_A加入知识库时, 首先检查知识库中是否有此命题。如有, 则转1a, 否则, 把命题A加入知识库中。

1a. 判断证实J_A的Inlist_A和Outlist_A表中的任一结点β是否被结点A直接或间接证实, 即如果$A\in Sancent(\beta)$, 或$\beta\in Sreply(A)$, 则表示结点A和结点β构成了一个循环证实回路, 因而在本系统中只简单地把β从J_A中删除以避免循环回路的出现, 设通过上述检查后的J_A变为J'_A。

1b. 把A的证实J'_A加入知识库中A的证实表JL_A中。

2. 判断命题A 在当前是否为可信命题，如不可信，则本次调用结束。

3a. 当命题A 为可信时，则在知识库中进行推理，看是否引起矛盾，如引起矛盾，则调用4 的相关性回溯，通过跟踪矛盾结点理由充足的证实，去掉引起该矛盾的非单调假设之一以消除矛盾并同时生成一个记录来避免以后出现类似的矛盾。

3b. 由命题A 和知识库中的其它命题$\beta_1,\cdots,\beta_m$ 进行推理，如果能推出其它的命题C，则把命题A 和命题$\beta_1,\cdots,\beta_m$ 组成命题C 的一个证实连同命题C 一起转第1 步，否则结束。

4a. 从产生矛盾的结点开始，回溯跟踪该矛盾的理由充足的支持以便寻找导致矛盾的非单调假设集。具体做法是从矛盾起通过支持矛盾的状态为I_n 的结点向后标记，直到找到一个非单调假设并返回所找到的非单调假设里的所有新假设(所谓新假设是不通过中间非单调假设直接支持矛盾的非单调假设)，只修改新假设以避免不必要的在知识库中做很大的变动。

4b. 设返回的新假设表$S=(A_1,\cdots,A_n)$，从S 中选取一假设A_j，把此假设的Out 证实之一变为In 证实。具体做法是：假设A_j，其证实J_{A_j} 为：$(Inlist,(\beta_1,\cdots,\beta_j,\cdots,\beta_m))$，再寻找一个新证实$J_D$(可能为空)，它由所有这样的命题组成，这些命题最终支持矛盾，但它们既不是S 中的元素，也不为S 中的元素所证实，设J_D 为：$((A'_1,\cdots,A'_i),(\beta'_1,\cdots,\beta'_j))$，给命题$\beta_j$ 一个新的证实J_{β_j}：$((A'_1,\cdots,A'_i,A_1,\cdots,A_{j-1},A_{j+1},\cdots,A_n),(\beta_1,\cdots,\beta_{j-1},\beta_{j+1},\beta_m))$，从而使$A_j$ 的状态变为Out 来消除矛盾，并把命题β_j 和其证实J_{β_j} 一起转第一步，导致知识库的进一步和谐和完善。

下面举几个例子来看看系统是如何进行推理维护的。

设系统中知识库所存的知识如图2 所示：图中"–" 表示对应的知识当前为不可信的，"+" 表示对应的知识当前为可信的。

图1 的含义为：知识M 或由知识$A_{11},\cdots,A_{1n_1}$ 推出，或由知识$A_{21},\cdots,A_{2n_2}$ 推出，或由……，知识$A_{n1},\cdots,A_{nn_n}$ 推出。

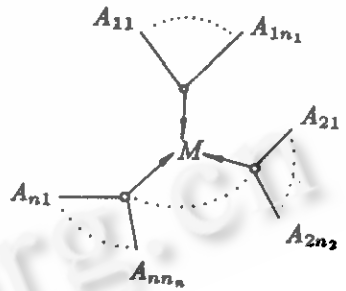

图 1

hair (yang, yellow), eye (yang, blue),
hair (zh, yellow), eye (zh, brown),
contradiction ←is (X, one), is (X, two).

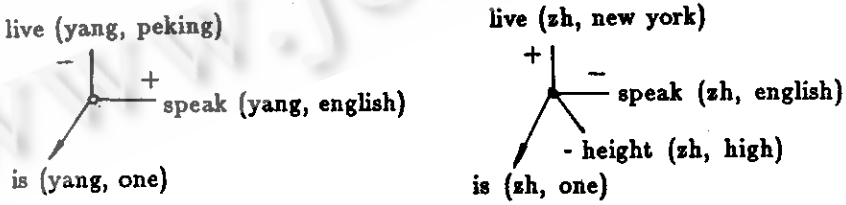

图 2

当把下述知识：

hair (X, yellow), eye (X, blue) → is (X, one)　　(2)
hair (X, yellow), eye (X, brown) → is (X, two)　　(3)

加入知识库(图2) 中，我们来看看知识库的变化情况。

把知识(2) 加入图2 的知识库，可推出is (yang, one) 这个新知识，但由于知识库中已存在is (yang, one) 这条知识，为保证知识库的非冗余性，则只把关于is (yang, one) 的证实加入其证实表中，如图3 所示。

hair (yang, yellow), eye (yang, blue),
hair (zh, yelow), eye (zh, brown),
contradiction ← is (X, one), is (X, two),
is (X, one) ← hair (X, yellow), eye (X, blue)。

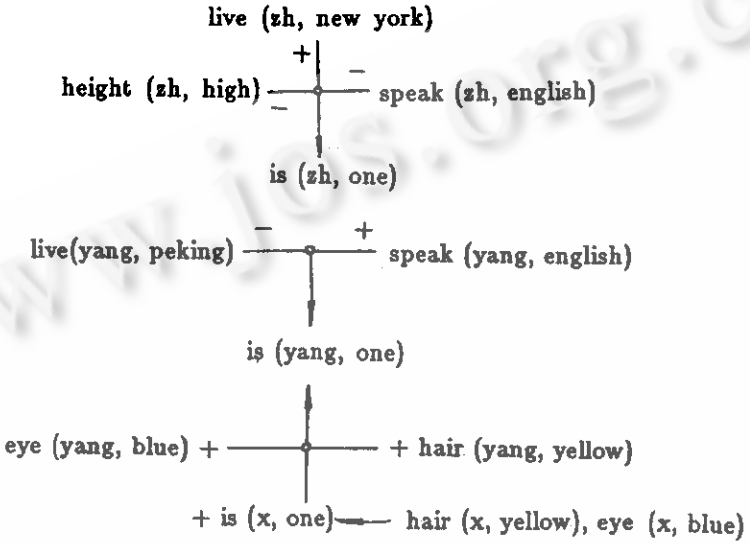

图 3

把知识(3) 加入图3 所示的知识库中，可推出新的知识is (zh, two)，使得知识库由于同时存在is (zh, one), is (zh, two) 这二条相互矛盾的知识而导致知识库的不一致性。当系统感知矛盾后，它沿着推理链逆向追踪，找到支持产生矛盾的有关假设(标有"–" 号)，修改为其相反的状态，即修改为"+"，然后再沿正向推理链修改所做的假设以消除矛盾。完整的做法是通过给知识speak (zh, english) 一个新的证实: [[is (zh, two), contradiction ← is (X, one), is (X, two)], [height (zh, high)]]，而使得speak (zh, english) 这条知识在当前知识库中变为可信的，从而使is (zh, one) 变为不可信而消除知识库的不一致性。最后知识库如图4 所示。

hair (yang, yellow), eye (yang, blue),
hair (zh, yellow), eye (zh, brown),
contradiction ← is (X, one), is (X, two),
is (X, one) ← hair (X, yellow), eye (X, blue),
is (X, two) ← hair (X, yellow), eye (X, brown),

§4. 结束语

用缺省推理来实现知识库的自动推理和维护的方法，已用于知识获取系统GKD–NKAS

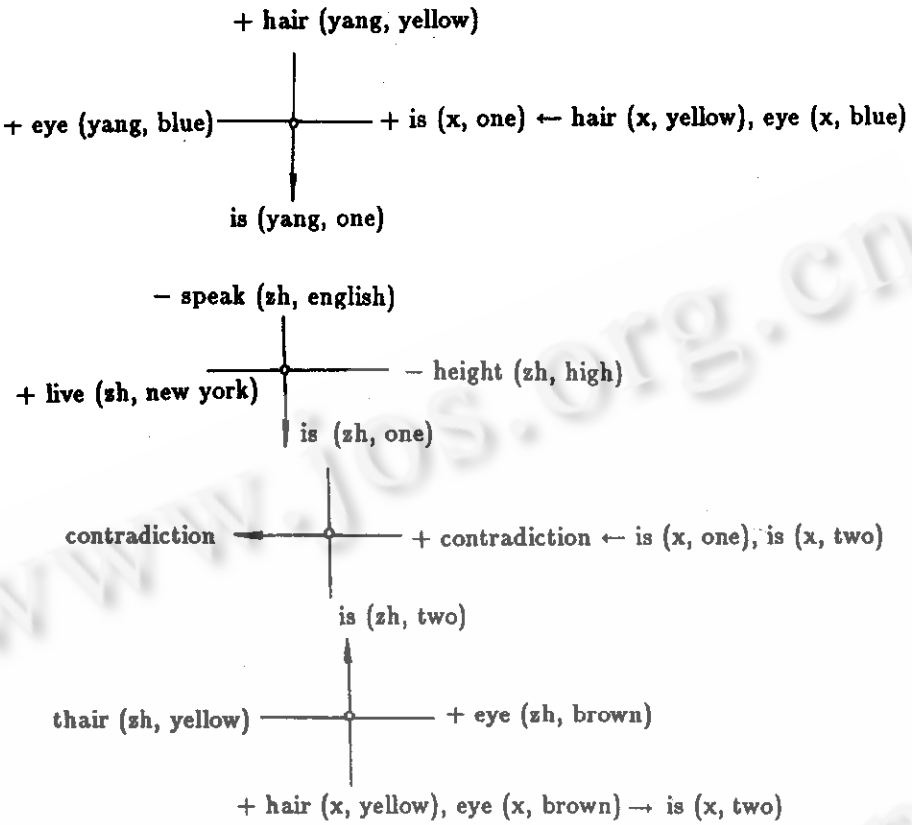

图4

中知识库的自动管理。在此系统中，承认不在系统内的命题不一定都是假命题，仍然可能是真命题，并且随着知识的增加原来系统内的命题也可能变为假命题。系统随着时间的推移可能不是增大而是减小，使系统的性能得以改善。因此，用此方法进行知识库的自动推理和维护可以使知识库更加智能化，并且可大大减轻一般人工智能系统开发人员的维护负担，提高智能系统的有效性和实用性。

参考文献

[1] Davis M, The Mathematics of Non-monotonic Reasoning, Arti. Intell, Vol. 13 (1980), PP 73–80.
[2] A Logic for Default Reasoning, R. Reiter, AI, vol. 13, No. 1, No. 2, April, 1980.
[3] Doyle J., The Ins And Outs of Reason Maintenance, IJCAI–8, 1983.
[4] 知识获取系统GKD–NKAS 鉴定会资料，国防科技大学，1989, 10。
[5] KIS: 一个新颖的知识精化系统，《89 年全国人工智能及应用学术会议论文集》。