基于人工神经网络的语义、句法模式识别

戴汝为

（中科院自动化所人工智能开放实验室）

摘　要：本文首先从科学中的启发性论据、系统的整体性能等方面加以引申来探讨形象（直感）思维。然后讨论人的模式识别，并以此为参照来考虑用计算机进行模式识别的问题，扼要地对以人工神经网络与物理符号系统相结合为着眼点的"语义、句法模式识别方法"加以介绍。并以手写汉字识别为例，说明这一方法在实际应用中的有效性。

1　前　　言

我国著名科学家钱学森于80年代初提出创建思维科学技术部门的主张，并认为思维科学研究的突破口在于形象思维[1]。1986年国家高技术计划把"智能计算机系统"作为信息领域的主题之一。由于开展了智能计算机的研讨，并对人工智能研究的进展缓慢加以反思后，越来越多的人对研究形象（直感）思维的重要意义有了更进一步的认识。

谈到形象（直感）思维往往使人联想到科学中的启发性论据（heuristics），就人工智能领域来说，专家系统取得了很大的进展，而启发式知识的应用及启发式搜索是专家系统不同于一般计算程序的关键之一。启发式知识是一些难以精确描述的知识，它以专家的经验为基础，以及一些直观感觉，带有假设的色彩。我们可以对启发式知识作这样的理解：有关目前问题的局况与合适的解之间的经验知识，这种经验知识难以用语言讲清楚，但又非常重要。另一个值得重视的问题是如何从整体上来看待一个系统的整体性能，把握全系统的形象，而不是系统中的一枝一节。格式塔（Gestalt）心理学的观点是值得借鉴的，格式塔的一个基本特征是，凡是格式塔，虽然它是由各种成份与要素构成的，但该格式塔决不等于构成它的所有成份之和。一个格式塔是一个完全独立于这些成份的全新整体，它是从原有的构成成份中"实现"出来的，它的特征与性质都是在原来的构成成份中找不到的，概而言之，部分不能决定整体；"整体"的性质反过来却可以对"部分"的性质有着极重要的影响。必须重视整体的形象。

还有一个问题是中医里的"证"，即从人体的整体状态考虑进行综合诊治，不是头疼医头，脚疼医脚，而是讲从整体上考虑问题。另外，人从眼、耳、鼻、舌等感官获得信息，进行模式识别，把感官感觉到的信息和知识，与大脑中存贮的信息从整体上加以比较，进行搜索与匹配，找出并识别"形象"。而这种搜索与匹配的效率与情感密切相关。从以上这些方面加以引伸来探讨形象（直感）思维，可以说是抓住了问题的实质。

2　中国传统文化中的比喻的方法

在讨论到形象（直感）思维时，认为一些经验体会是"只可意会，不可言传"。有人就问，既

然不可言传，那么怎么能理解，又如何加以研究呢？这是中国传统思维着重于“用心体会”的认知方式存在着不容易进行交流的弱点，但绝不是无法研究。中国历史上佛教到了六祖慧能继承衣钵创立了南宗顿教，规定的教义之一是“不立文字”。讲求的是创造的直观，亦即在感受中领悟到某种宇宙规律。慧能本人是不识字但能“悟”的典范，而佛教也一直流传下来。《易传·系辞》上曾有一段很深刻的文字，“书不尽言，言不尽意；……圣人立象以尽意”。前人对只可意会的东西的论述，采用的是比喻的方法。比较典型的一个例子，是范缜对形神关系的比喻，就非常精采，在中学教科书中可找到。形神的比喻是：“神之于质犹利之于刃，形之于用，犹刃之于利。利之名非刃也，刃之名非利也；然而舍利无刃，未闻刃没而利存，岂容形亡而神在？”(《神灭论》)。前人还采用“比类取象”、“援物比类”等方法。这里我们用中医诊断疾病来对形象(直感)思维作比喻，也许可以对形象(直感)思维有进一步了解与较好的说明。从中医诊断来加以分析，首先是立“象”[2]，大夫通过多种媒体(望、闻、问、切)等感知及自身的体会，建立子模式，再形成与各种病症对应的模式类。立“象”的过程是大夫与大量的病人接触、学习与诊断的过程，病人实际上起到为大夫提供样本的作用。立“象”是十分复杂的认知过程，是大夫的实践与经验积累的过程。可以说是多维的(望、闻、问、切)自下而上的综合集成(metasynthesis)过程[3]，这是由大夫的大脑这一系统来完成的。最终把实践经验沉积于大夫的脑中，形成表征与各种病症相对应的各种模式类的“意”，达到立象表意。立“象”很重要，在此过程中，采用以象说象的办法建立一种描绘整体形象的比较抽象的象称为“意”。中医大夫对病人疾病的诊断是靠对人体的整体了解以“意”之间的相似性来加以判断的。以象说象的办法可以用一个二元式来表示一个模式 P 的方式加以表达，P 包括象(用 I 表示)与意(用 E 表示)两个成分，

$$P=P(I,E),$$

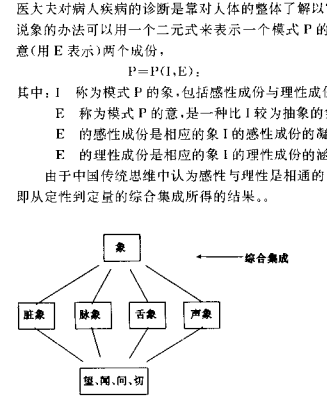
其中：I　称为模式 P 的象，包括感性成份与理性成份；

E　称为模式 P 的意，是一种比 I 较为抽象的象，也包括感性成份与理性成份；

E　的感性成份是相应的象 I 的感性成份的凝炼和浓缩；

E　的理性成份是相应的象 I 的理性成份的涵盖和总结。

由于中国传统思维中认为感性与理性是相通的。所以 E 可以认为是从 I 通过综合集成，即从定性到定量的综合集成所得的结果。。



如上图所示，从象到意可以是一个层次，也可以用多层次形象的类比与寓意来完成。

总之，人识别模式可以说是形象(直感)思维的一种方式，而综合集成在识别的过程中起

着核心的作用。关于模式识别，德国的涅曼教授有如下的说法：对于简单的模式，识别指的是分类。对于复杂的模式，识别指的是描述。可以说模式识别包括分类与描述两重含义。归结起来，人的模式识别有两个要点，即认知(cognition)与识别(recognition)，英文的 recognition 这个字表达得较为清楚，说得更详细一点。人的模式识别可概括为：

(1) 以实践与经验为基础，通过多种媒体的认知(cognition)经反复学习，包括有教师的学习，在人脑中建立模式类，即"建模"或立"象"，并通过综合集成而从总的方面掌握"意"存贮于脑中。

(2) 对新出现的模式，用一种衡量"意"之间的相似性度量进行搜索识别(recognition)。判定与存贮在脑中的哪一类模式相匹配。这就需要对存贮在脑中的所有模式进行搜索。而在搜索过程中不是一个一个的顺序搜索，搜索与匹配是两个重要的环节，与人的情感有关，那种突如其来的发现，例如突然领悟到了"呀！原来是这么回事"的体验，是需要借助于情感的。当人们颇为激动的时候，他们从象到意的综合集成及搜索与匹配的效率就会提高。总之，从中国传统思维中把形象(直感)思维概括为"象、意、情"的综合体现，应该说是很有道理的。

3 模式的语义与句法两者间的关系

上面我们讨论了人的模式识别。用计算机进行模式识别所形成的软、硬件技术已成为信息技术中的一个组成部分。虽然也有一些人从认知的角度来研究模式识别，但到目前为止，用计算机进行模式识别的工作中，所面对的都是十分死板的模式。对于"建模"这一重要问题几乎完全忽略，把问题转为"对样本的特征抽取"，而特征抽取并没有一般的行之有效的方法。在利用计算机进行处理的前题条件下，上节所谈到的关于一个模式包含两部分的构思得到充分的发挥。一个模式也可定义为：

$$P=P(x,u)$$

其中：x 表示 P 的结构信息，称为句法部分；

u 表示 P 的属性等，称为语义部分。

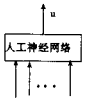
并考虑一个模式 P 由一些称为基元(primitive)的基本元素构成，这些基元具有某些能刻划基元性质的属性，属性可以用数值表示，基元之间的相互联系用关系属性表示，由若干基元可以构成子模式，再由若干子模式构成更复杂的模式，以此类推，最终构成所研究的模式。如果一个子模式由若干基元组成，而基元的属性已经知道，那么子模式的属性如何由基元的属性加以决定的办法称为语义规则或语义函数。于是就形成一个形式化的体系，可以采用一种称为"语义句法方法"的方法[1]来描述这一体系。这种描述也包括两个部分：一是句法部分，用一个上下文无关文法或有限状态文法的导出式表示；二是语义部分，它包括基元、子模式的属性、基元与子模式间的关系属性，以及语义函数。

在用计算机进行模式识别的发展过程中，最初采用的一般性的数学方法是统计法或决策理论法，以后又提出一种以形式语言理论为基础的句法方法[2]，其着眼点是构成模式的语法与语义之间存在着一种折衷的关系，即可以通过使语义的表达变得复杂一些而降低句法的复杂性，或者反之使句法的表达复杂一些而降低语义的复杂性。这一方法实质上是把统计方法与句法方法有效地综合在一起，既体现出统计方法的优点，又具有句法方法能利用结构信息的长处；传统的统计方法与句法方法实际上分别成为这一方法的特殊情况。

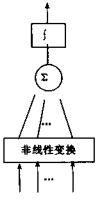
4 确定语义函数的近似方法

我们讨论人的模式识别时，从“象”到“意”是人脑加以综合集成而达到的，用计算机根本没法作到。局限于语义句法的描述而言，实际上在对语义的处理方面，如何计算语义函数也是个问题，换句话说，如果一个子模式 $P(x,u)$ 由 k 个子模式 $P_1,\ldots,P_k$ 模式构成，它的属性为 u，P_i 的属性为 u_i，$i=1,2,\ldots,k$。说得简单一点，所谓的语义函数，就是如何由 $u_1,u_2,\ldots,u_k$ 而得到 u。如果一个模式由一个树状结构表示，那么问题就是如何从树的叶节点的属性逐层往上综合以求得与根节点相对应的属性。语义函数是一个非线性的映射，这种非线性的映射计算起来比较困难。

近年来人工神经网络研究取得了很大的进展，人工神经网络可以作为一种分布式表达，所谓的分布式表达是相对于局部表达而言的。局部的含义在于所使用的基元具有独立的物理意义，这就是传统的物理符号系统的基本假设。而分布式表达则是表达模型所使用的基元，没有独立的物理含义。人们很自然地把人工神经网络与物理符号系统两者结合起来。另外，有人论证了在满足一定条件的情况下，一个多层人工神经网络可以实现一个非线性映射。根据这个理论结果，我们就可以用人工神经网络来作为语义函数，如图所示。而这种人工神经网络的参数即“权”值是通过有教师的学习来完成的，由教师把握住宏观的情况，所以人在这一过程中起到积极的作用。这样一来确定语义函数就有了一定的办法。我们把模式的语义句法描述及采用人工神经网络的识别方法称为“基于人工神经网络的语义句法模式识别方法”[5]。



图：语义函数的构建原理



图：语义函数构成

这一方法体现了把人工神经网络与物理符号系统两者相结合用于解决模式的识别问题。上述确定语义函数的方法已在手写汉字识别中得到应用[6]，识别手写汉字所遇到的困难是由于各人书写汉字时的方式与习惯不同，所以是千变万化的，计算机难以判断对于输入机器的一个汉字，有的笔划究竟是一个短划呢，还是由于噪声形成的“毛刺”。另外，由于连笔等问题把笔划当成基本元素不实际，所以把字根（偏旁部首）作为基本单元，称为子结构。通过一些运算把子结构检测出来，与子结构对应的运算是靠人确定的，一个子结构由若干笔划组

成，检测到的子结构由畸变的笔划构成、具有模糊性，把这些模糊了的笔划的属性 u_1，u_2，…，u_k 送到一个类似于感知器(perceptron)的网络中，并经过一定的非线性处理来求得与子结构对应的属性(用一个向量表示)，然后根据一系列的样本来进行学习，如图所示。例如对于字根"木"，把左偏旁正、负样本输入计算机，当一个样本输入时确定网络的输出应该是"正"或"负"，用一个算法对参数加以调整，并给出一定的准则，以确定在 n 个参数 P_1，P_2，…，P_n 中，有较大影响的 m 个，则这个量可以定义为字根的属性。该训练过的网络就起到语义函数的作用。

5 人-机结合的识别系统

研制模式识别系统有两种构思，一种是致力于研究完全不依赖于人的系统，也就是自主(autonomous)系统；另一种是研制人与计算机两者均能发挥作用的人-机结合的系统。关于人-机结合的观点，作者已经专门有过阐述[7]。以往对模式识别的研究过程中曾经引入过学习的机制，并分为有教师的学习(supervised learning)与无教师的学习(nonsupervised learning)两种类型，前者有作为"教师"的人参与系统的运行，后者则是按某种给定的准则进行学习，无人参与。在有教师的学习中，关键之处让人指点一下，大量繁琐的工作由计算机去完成，教师的作用是把握宏观状况。例如在识别手写汉字的过程中，采用人工神经网络来识别字根，需要通过包括该字根的正样本，与不包括该字根的负样本来进行学习，从而决定网络的参数。这就需要通过一个教师来判断当样本输入时，究竟是正样本还是负样本，需不需要调整网络的参数，教师起了大作用。从长远目标来看，今后我们要研究的是人和机器相结合的识别系统，或智能系统，但目前只能做些"妥协"，实事求是，尽量开拓当前计算机的科学技术，使计算机尽可能地多帮助人来完成一些识别的工作。

致谢 作者多次从钱学森先生的来信中得到启示、鼓励与帮助，在此表示衷心的感谢！

参 考 文 献

[1] 钱学森主编，《关于思维科学》，上海人民出版社，1986

[2] 王前、刘庚祥，从中医取象看中国传统抽象思维，《哲学研究》，1993 年第 4 期

[3] 戴汝为，从定性到定量综合集成技术，《模式识别与人工智能》，1991 年第 4 卷第 1 期

[4] J. W. Tai (戴汝为)、K. S. Fu (傅京孙)、Semantic Syntax-Directed Translation for Pictorial Pattern Recognition，Purdue Univ. Tech. Rep.、TR-EE-81-38，Oct.，1981

[5] R. W. Dai (戴汝为)，A Connectionist Syntactic Semantic Approach for Pictorial Pattern Recognition，Infoscience'93，Seoul Korea，(Invited paper)，1993

[6] 刘迎建、戴汝为、张立清，基于神经网络的手写汉字特征选择，《模式识别与人工智能》，1992 年第 5 卷第 3 期

[7] 戴汝为、王珏，关于智能系统的综合集成，《科学通报》，1993 年第 38 卷第 14 期

时段演算综述

周巢尘

（联合国大学国际软件技术研究所，澳门 3058 信箱）

摘　要

时段演算（Duration Calculus）的研究始于 1989 年。当时 Esprit 的研究项目 ProCos 正寻求设计严格安全系统的形式技术，应该项目的需要开始了时段演算的研究。迄今四个演算已相继建立，即时段演算（Duration Calculus），扩充时段演算（Extended Duration Calculus），平均值演算（Mlan valul (Calculus)），和概率时段演算（Probabilistic Duration Calculus）。

时段演算是一种实时区间逻辑（Interval Logic），见［14］。它将布尔函数在区间上的积分进行形式化，从而可用来描述和推导离散状态系统的实时和逻辑特性，其余的三个演算均是它的扩充。文献［16］中的扩充时段演算引入了分片连续式可微函数，这样就可用来描述连续状态的特性，它已应用于具有连续和离散状态的混成系统（Hybrid Systems）的设计，平均值演算[15]则以平均值取代积分，以 δ-函数表示瞬时动作，如通信和事件等。平均值演算可用来将基于状态的需求说明，经状态和事件混合描述，求精为以事件为基础的描述，甚至最终成为可执行的程序。概率时段演算[5,6]为设计人员提供规则，来推导和计算一个不可靠系统，满足时段演算中表达的需求的概率，这里，不可靠系统是由概率自动机作为模型的。

时段演算已应用了若干实例，如煤气燃烧器[9]，铁路交叉口控制[10]，水位控制[11]和自动导航[8]，该演算亦已用于定义 Qccam 式语言的实时语义[12,4]，描述调度程序的实时行为和线路设计[2]。

时段演算亦已应用于联合国大学国际软件技术研究所和我国 863 计划的联合项目——DeTfoRS（Design Technique for Real Time Systems）。该项目应用时段演算发展了混成系统的形式设计技术及离散化技术。DeTfoRS 的研究论文可见于［18－22］。

时段演算的工具正在开发中，请见文献［13］、［11］、［17］。

参 考 文 献

[1] M. Engel, M. Kubica, J. Madey, D. J. Parnas, A. P. Ravn, A. J. van Schouwen: A Formal Approach to Computer Systems Requirements Documentation, presented in the Workshop on Theory of Hybrid Systems, Lyngby, Denmark, 19－20 Oct. 1992

[2] M. R. Hansen, Zhou Chaochen, J. Staunstrup: A Real-Time Duration Semantics for Circuits, Proc. of the Workshop on Timing Issues in the Specification and Synthesis of Digital Systems, Princeton, March 1992

[3] M. R. Hansen, Zhou Chaochen: Semantics and Completeness of Duration Calculus. J. W. de Bakker, C. Huizing, W. -P. de Roever, G. Rozenberg, (Eds) Real-Time:: Theory in Practice, REX Workshop, LNCS 600, pp 209—225, 1992

[4] He Jifeng, J. Bowen: Time Interval Semantics and Implementation of a Real-Time Programming Language. Proc. 4th Euromicro Workshop on Real-Time Systems, IEEE Press, June 1992

[5] Liu Zhiming, A. P. Ravn, E. V. Sorensen, Zhou Chaochen: A Probabilistic Duration Calculus, presented in the 2nd Intl. Workshop on Responsive Computer Systems, Saitama, Japan, Oct. 1—2, 1992 published in H. Kopetz and Y. Kakuda (eds), Dependable Computing and Fault-Tolerant Systems Vol. 7: Responsive Computer Systems, pp 30—52. Springer Verlag Wien New York, 1993

[6] Liu Zhiming, A. P. Ravn, E. V. Sorensen, Zhou Chaochen: Towards a Calculus of Systems Dependability, presented in the Workshop on Theory of Hybrid Systems, Lyngby, Denmark, 19—20 Oct. 1992, accepted by High Integrity Systems Journal, Oxford University Press

[7] B. Moszkowski: A Temporal Logic for Multi-level Reasoning about Hardware. In IEEE Computer, Vol. 18 (2), pp 10—19, 1985

[8] A. P. Ravn, H. Rischel: Requirements Capture for Embedded Real Time Systems, Proc. IMACS-MCTS'91 Symp. Modelling and Control of Technolgical Systems, Vol 2, pp. 147—152, Villeneuve d'Ascq, France, 1991

[9] A. P. Ravn, H. Rischel, K. M. Hansen: Specifying and Verifying Requirements of Real-Time Systems, Proceedings of the ACM SIGSOFT'91 Conference on Software for Critical Systems, New Orleans, December 4—6, 1991, ACM Software Engineering Notes, Vol 15, No 5, pp 41—54, 1991 (published also in IEEE Trans. Software Eng., Vol 19, No 1, pp41—55, January 1993)

[10] J. U. Skakkebæk, A. P. Ravn, H. Rischel, Zhou Chaochen: Specification of Embedded Real-Time Systems, Proc. 4th Euromicro Workshop on Real-Time Systems, IEEE Press, pp 116—121, June 1992

[11] J. U. Skakkedæk, P. Sestoft: Checking Validity of Duration Calculus Formulas, submitted to Conference on Computer-Aided Verification, Crete, June 1993

[12] Zhou Chaochen, M. R. Hansen, A. P. Ravn, H. Rischel: Duration Specifications for Shared Processors, Proc. of the Symposium on Formal Techniques in Real—Time and Fault-Tolerant Systems, Nijmegen, January 1992, LNCS 571, pp 21—32, 1992

[13] Zhou Chaochen, M. R. Hansen, P. Sestoft: Decidability and Undecidability Results for Duration Calculus, Proc. of STACS '93. 10th Symposium on Theoretical Aspects of Computer Science, Wurzburg, Feb. 1993

[14] Zhou Chaochen, C. A. R. Hoare, A. P. Ravn: A Calculus of Durations, Information Processing Letter, 40, 5, pp. 269—276, 1991

[15] Zhou Chaochen, Li Xiaoshan: A Mean-Value Duration Calculus, UNU/IIST Report No. 5, March 1993, to be published in the Hoare Festschrift, Prentice-Hall International

[16] Zhou Chaochen, A. P. Ravn, M. R. Hansen: An Extended Duration Calculus for Hybrid Real-Time Systems, to be published in the Proc. of the Workshop on Theory of Hybrid Systems, Lyngby, Denmark, 19—20 Oct. 1992

[17] J. U. Skakkebæk, Natarajan Shankar: Towards a Duration Calculus Proof Assistant in PVS, ProCos II Report ID/DTH JUS 5/1, March 1994

[18] Michael R Hansen, Paritosh K Pandya, Zhou Chaochen: Finite Divergence, UNU/IIST Report No. 15, Nov 1993

[19] Yu Xinyao, Wang Ji, Zhou Chaochen and Paritosh K Pandya: Formal Design of Hybrid Systems, UNU/IIST Report No. 19, Feb. 1994

[20] Wang Ji, Yu Xinyao and Zhou Chaochen: Refinement of Digital Dynamic Systems, UNU/IIST Report No. 20, Feb. 1994

[21] Yu Huiqun, Paritosh K. Pandya, and Sun Yongqiang: A Calculus for Hybrid Sampled Data Systems, UNU/IIST Report No. 21, Feb. 1994

[22] Zheng Yuhua and Zhou Chaochen: A Formal Proof of the Deadline Driven Scheduler, UNU/IIST Report No. 16, Feb. 1994

计算智能：一个重要的研究方向

李 国 杰

（国家智能计算机研究开发中心）

摘 要：本文论述计算智能是值得重视的一个研究方向，着重讨论了演化计算的理论基础和优点，强调许多实际的智能应用只有靠大规模并行处理才能解决。本文的目的在于引起国内学者对人工智能研究新方向的重视。

863 计划智能机主题已经进行 6 年了，现在来回顾一下近十几年来国际上人工智能研究的历史和分析考察这一领域的新动向是件十分有意义的事。80 年代中期，人工智能曾经被认为是“下一件最大的事情”，有 400 多个厂商标榜生产人工智能产品。到 1993 年，只有少数几家保存下来，绝大多数人工智能技术与产品已溶入主流计算机产品，成为更大市场的一部分。现在一些成熟的 AI 技术已像空气一样渗透到计算机企业，而一些基础性的研究又回到了实验室。世界上第一本关于人工智能企业的杂志“AI Trends”，于 1994 年开始改名为“Critical Technology Trends”，编者在更名启事中讲到遗传算法、自适应系统、细胞自动机和混沌理论与人工智能一样都是对今后十年的计算技术有重大影响的关键技术。这些关键技术大多属于日益被重视的“计算智能”(Computational Intelligence)。

1 计算智能与人工智能

世界上有“计算智能”学术会议，其规模不亚于人工智能会议，也有“计算智能”学术刊物，但究竟什么是计算智能，并没有确切的定义。如同人工智能一样，不同的人对计算智能有不同的理解。我们不必急于为计算智能下定义，更不必像争论“智能计算机”一样在名词上浪费时间，重要的是弄明白“计算智能”究竟包含哪些新思想。

广义地讲，人工智能也是试图用计算来实现人的智能，所以人工智能也可以看作计算智能。当加拿大的学者创办“计算智能”学术刊物时，人们只觉得增添了一种人工智能学报，并未仔细考虑这两者的区别。随着人工神经网络、遗传算法、演化程序、混沌计算等研究逐渐兴旺，而每年召开的人工智能学术会议，如 AAAI 等，又不大乐意接受这方面的论文与产品演示[1]，从事上述研究的学者逐步形成相当规模的国际学术会议，取名为计算智能，似乎造成一种与人工智能分庭抗礼的局面。但从学术上讲，把计算智能看成人工智能研究的新方向也许更恰当。

本世纪 30 至 40 年代，当图灵等著名学者提出计算模型时，并没有强调数字计算与符号计算的区别。到 60 年代，采用数字计算来模拟思维的各种研究一直是很活跃的。60 年代中后期，尤其是 70 年代以后，建立在心理学刺激/反应模型基础上的物理符号系统逐渐成为思维模拟的主流，符号处理也几乎成了人工智能的代名词。近些年来，人们一谈到人工智能就

马上想到逻辑、规则、推理；而一谈到计算就联想到矩阵运算、解微分方程等，似乎智能与计算是两股道上跑的车。人工智能走过几十年曲折道路之后，现在是需要认真反思的时候了。

研究思维模拟主要的道路有四条：基于心理学的符号处理方法、基于社会学层次型的智能体(Agent)方法，基于生物进化的演化计算与自适应方法以及基于生理学的人工神经网络方法。目前集聚在计算智能大旗下的主要是后两个学派的学者(加上从事模糊计算和混沌计算等方面的学者)。实际上，只要在计算机上(不考虑模拟量计算机)模拟人类思想，不管用什么方法，其本质的基础还是二进制数字计算。所谓Church Turing假设原始的论述只是对数论函数(以自然数为输入)而言，后来人们有意无意地推广到任意函数、人脑功能甚至一切物理现象，这一推广是否有道理一直有争议[4]。我们不应过份强调数字计算与符号计算的区别，而应强调综合集成。在当前符号处理主宰人工智能的情况下，更应强调遗传算法等以数字计算为基础的方法对推动人工智能发展有特殊的作用。

人工智能研究中处处碰到组合爆炸，而局部搜索等演化计算是对付组合爆炸的有力工具。在局部搜索中甚至可以把离散优化问题转换成连续量优化问题，采用经典的数学方法求解[3]。近年来已经证明符号逻辑的基础问题——SAT问题以及许多典型的NP完全问题，可以转换为类似波动方程的微分方程问题，为有效地求解这些困难问题提供了新思路[4]。因为演化计算中算法已不是经典意义下的算法、神经网本质上是一动力学系统，这些新思想的引入可能使计算复杂性和可计算性的研究出现新局面。例如，S. Franklin已经证明给定任意二进制表示的实数，用图灵机无法判定它是不是整数，但用与图灵机相对应的具有无限离散量神经元的神经网络可以判定[5]。在集成神经网络与符号计算方面，国外已做了大量工作，预示着光明前景。

2 演化计算

人的智能是从哪里来的？归根结蒂是从生物进化中得到的，反映在遗传基因中。脑的结构变化也是通过基因的变化一代一代遗传下来。每一种基因产生的生物个体(看成一种结构)对环境有一定的适应性，或叫适合度(fitness)。杂交和基因突变可能产生对环境适应性强的后代，通过优胜劣汰的自然选择，适合度高的结构被保存下来。因此，从进化的观点来看，结构是适合度的结果。在这种观点启发下，60年代Fogel等提出了演化程序(Evolutional Programming)思想，70年代Holland提出了遗传算法。如同神经网络研究一样，经过近二十年的沉寂到80年代后期，由于在经济预测等应用领域获得成功，演化计算成为十分热门的研究课题。

演化计算实质上是自适应的机器学习方法，它的核心思想是利用演化历史中获得的信息指导搜索或计算。常用的演化计算包括遗传算法。遗传程序(Genetic Programming)、演化程序、爬山法即局部搜索、人工神经网络、决策树的归纳以及模拟退火等等。这些不同的方法具有以下几项共同的要素：(1)自适应的结构，(2)随机产生的或指定的初始结构，(3)适合度的评测函数或判据，(4)修改结构的操作，(5)每一步中系统的状态即存储器，(6)终止计算的条件，(7)指示结果的方法，(8)控制过程的参数。上述几种演化计算方法中，只有遗传算法与遗传程序是一组结构(a population)同时演化，其他方法是一个结构的演化。所谓遗传程序与通常的遗传算法的主要区别在于采用的"结构"(即问题的表示)不同。最初的遗传算法

的自适应结构为定长的二进制字符串；而遗传程序的结构是分层的树，表示 LISP 语言中的 S 表达式，即一个解决指定问题的程序[6]。遗传程序的目标是自动生成程序。不同演化计算方法采用不同的结构，实质是不同的问题表示。一个问题的复杂性决定于它的问题表示[7]，因为一种表示限制了系统观察世界的窗口。笔者认为在演化计算方面做研究应当从问题表示入手，即选择表示能力强又操作方便的结构。

众所周知，任何一种传统的科学或工程方法都具有正确性、一致性、可验证性、确定性、次序性及简洁性等特点。演化计算是模拟自然界的进化过程，自然界是靠适应性而不是靠简洁性解决问题，常常采用间接的复杂的方法。与自然进化类似，演化计算一般不提供简洁的求解方法。在演化计算中，随机选择是关键的因素，因此往往具有不确定性，甚至同时支持不一致的相互矛盾的途径去求解。演化计算一般不采用严格同步控制。与传统算法最大的不同是计算不是自动终止，往往是入为限制进化多少代结束，或通过控制演化结果的一致性程度设定终止条件。在演化过程中即使达到了最优解或要求的目标，程序本身并不知道（除非设置一个全局监视器）。许多从事神经网求解优化问题和演化计算的学者都深有体会，常常无法判断自己得到的结果好坏如何。演化计算的这些新特点给我们带来许多新的研究课题。国家智能计算机研究开发中心的一些学者研究用局部搜索和 D-P 算法求解 SAT 问题，对 SAT 问题的难易分布做了深入分析，得到了一些很有价值的研究成果，用统计的方法可以准确地指出最难的问题实例在问题空间的什么地方。

演化计算的主要优点是简单、通用、鲁棒强和适于并行处理。目前演化计算已广泛用于最优控制、符号回归、自动生成程序、发现博奕策略、符号积分微分及许多实际问题求解。它比盲目的搜索效率高得多，又比专门的针对特定问题的算法通用性强。它是一种与问题无关的求解模式。当然，如果配合与领域有关的知识，求解效率会明显提高。总之，演化计算不论从理论上，还是实际应用上都为我们提供了一片广阔的研究天地。

3 演化计算与并行处理

从事人工智能研究的权威学者，如 Simon，Feigenbaum 等都认为并行处理不是人工智能的重要研究课题，强调思维本质是串行的。日本进行的五代机研究将并行推理作为关键目标，结果碰到了困难，因为传统人工智能的推理过程确实只包括少量并行性。笔者曾深入研究过并行处理与启发式搜索的相互制约关系，发现启发式函数质量越高，越缺乏"或"并行。所以并行处理在以符号逻辑推理为中心的人工智能领域中没有找到合适的位置。

但是，对于演化计算情况则大不一样。演化计算非常适合大规模并行。神经网络计算本身是大规模并行，不必赘述。对于遗传算法、遗传程序等计算，最有效的并行方式是让几百甚至数千台计算机各自独立进行不同个体或个体子集的遗传计算，运行过程中不需要任何通信，等到最后运算结束时才通信比较，选出最佳结果。显而易见，演化计算需要大规模并行来提高速度与解的精度，而并行处理的平台可以是松耦合的系统，如工作站群集，甚至在局部计算网上都可实现，因为即使只在粗粒度级上并行，演化计算的并行度也很高。

目前，国内外人工智能的应用研究大部分还是在 PC 机或工作站上进行，已经取得一些可喜的成果。但是，对于真正的智能应用，比如实时的连续语音识别、机器翻译、通用的自然语言理解、全局性的复杂的计算辅助决策等等，只有在大规模并行机上才能真正实现。人工

智能的实际应用一定要解决可扩展(Scalability)问题，这是90年代对人工智能学者的巨大挑战。停留在微机与工作站研究上，串行推理不可能完成这一艰巨任务。1993年在法国召开的国际人工智能大会(IJCAI'93)上，日本学者关于大规模并行人工智能的论文获得最佳论文奖，会上还专门举行了关于人工智能的巨大挑战(Grand Challenge)的专题讨论会，这一动向反映了人工智能研究的发展趋势。我们应十分重视计算智能及其并行处理的研究，将我国人工智能与智能计算机的研究推向新高度。

参 考 文 献

[1] "AAAI: Not Gone, But Forgotten?" ACM SIGART Sept, 1993

[2] C. Cleland, "Is the Church-Turing Thesis True?" Minds and Machine, Vol. 3, No. 3, pp. 283—312, Aug, 1993

[3] 顾钧，李国杰：求解可满足性(SAT)问题的算法综述，《模式识别与人工智能》，Vol. 6, No. 2, pp. 89—98, 1993年第二期

[4] L. Grover, "Local Search and Local Structure of NP-complete problems", Operations Research letters, pp. 235—243, Oct, 1992

[5] S. Franklin and M. Garzon, "On Stability and Solvability (Or, when does a Neural Network Solve a Problem?)", *Minds and Machines*, Vol. 2, No. 1, pp. 71—84, Feb, 1992

[6] J. Koza, Genetic Programming, MIT Press, 1992

[7] 李国杰：人工智能的计算复杂性研究，《模式识别与人工智能》，Vol. 5, No. 3, pp. 169—176, 1992年9月

关于知识库维护的逻辑

李 未

（北京航空航天大学计算机科学系，北京 100083，Tel：201．6994）

摘 要：本文引入了知识库维护序列及其极限的概念，同时也给出了请如新规则、用户反驳以及知识库更新等概念，并证明了与之有关的定理．本文用规约方法定义了一个过程，该过程可以根据给定的用户模型和已有的知识库生成知识库维护序列；并证明这些序列均收敛，其极限为用户模型的全体真语句。本文还讨论了知识库更新的计算复杂性问题，并给出了一个R 演算。当知识库受到用户反驳时，使用该演算可以规约地导出知识库的所有更新方案。最后，对本文提出的理论与AGM 信念修正理论进行了比较。

1 引 言

在所有关于知识库维护的理论工作中，Doyle 的“真值维护系统”[Doy 79]以及Alchourrón，Gärdenfors 及 Makinson（以下简称 AGM）的“修改理论的逻辑”[AGM 85]占有最重要的地位。

Doyle 在其关于“真值维护系统”的论文中给出了第一个描述知识库中规则间规约关系的机制；当知识库需要修改时，系统可以根据这些规约关系，通过机械地改变规则的“in”及“out”状态，实现知识库的更新。

当知识库的某一结论遇到用户反驳时，Doyle 的系统将给出知识库的一个正确修改。但是，他的系统没有回答诸如这个解是否是最好的，或者至少在所有可能的诸修改方案中是好的这类问题。此外，从数学的观点看，这个系统比较繁琐，形式化程度比较低。

AGM 的理论形式地定义了知识库维护的基本特性，给出了诸如：扩展、约减和修正的概念，深入地研究了这些概念的性质，给出了关于这些概念的公理化系统。用他们的语言来讲是给出了关于这些概念的公设（POSTULATES）。

AGM 进一步指出，如果从知识库推出的结论 A 与事实不符，则根据他们的“信息经济原则”，知识库的最佳修改应该是此知识库中与¬A 协调的极大子集与¬A 的并集，AGM 称这个修改为极大选择修改。由于知识库中与¬A 协调的极大子集不只一个，AGM 假定根据他们的“信息经济原则”最佳修改只有一个。做这种假定是否合理值得商榷，因为在许多场合，人们不论从哪个角度讲都很难说在诸多极大子集中，某一个比其余的更重要。因此，很自然地要问：有没有其它更精确的方式来描述知识库修改的唯一性问题？如果有，它应是什麼？此外，AGM 的理论也没有给出一个构造性的方法去建立极大选择修改。

为了回答 AGM 理论遗留下来的问题，本文将提出一种知识库修改的逻辑理论。它的出发点是考察知识库修改的历史。事实上，表示这个历史的最直接的办法是按顺序列出知识库的文本：

$$K_1, K_2, \cdots, K_n, K_{n+1}, \cdots$$

这里，版本 K_{n+1} 是通过对版本 K_n 的修改得到的。对知识库的修改有两种：一种是将用户或专家提供的某些新规则收入知识库；另一种是当从知识库 K_n 推导出的某个结论与事实不符时，删除知识库中某些规则。一个知库的版本序列反映了该知识库的进化和维护过程。

为了从理论上描述这些问题，本文将用一个一阶形式理论表示一个(广义)知识库，用形式理论的序列来刻画版本序列。本文还引入了形式理论序列的极限来描述知识库进化的最终结果。本文将给出一个可以生成知识新版本的一般过程。此过程将根据给定的实际问题(称为用户模型)，从一给定的知识库版本开始，产生出新版本。当然，知识库的任何一个版本都容许有错误，即，它的某些知识与(用户模型的)事实相矛盾。本文将证明这些版本构成的序列均收敛，其极限就是刻画这个问题全部特征的规律的集合。这个结果说明 AGM 提出的有关极大选择修改的唯一性问题并不确切。精确的说法应是：对给定的用户模型和知识库，由此过程生成的版本序列的极限是唯一的。

本文还定义了一个规约系统，以便当知识库与事实不符时，产生极大选择修改。我们称这个规约系统为 R_演算。

总之，本文第二节引入了形式理论序列及其极限概念；第三节讨论了用模型论与证明论的概念刻画了用户与知识库的交互作用问题；第四节使用规约方法定义了知识库维护的一般过程，并证明了相应的收敛性定理；第五节定义了 R_演算；最后，第六节将本文提出的理论与 AGM 的信念修正理论做了比较。

2 序列和极限

本文使用的形式语言是一阶语言 L，在[Gall 97]中可以找到 L 的严格定义。A，Γ 和 Th(Γ)将被用来表示公式、公式的集合以及由公式集Γ推出的全部定理的集合。本文用 Γ⊢A 表示一个序贯(sequent)。

本文将使用[Paul 87]给出的 Gentzen 推理规则，因此在必要时，将不区分 Γ 是序列还是集合。

模型 **M** 是一偶对<M，I>，其中 M 为论域，I 是一个解释。有时 $\mathbf{M}_\varphi$ 被用来表示关于特定问题 φ 的模型，而 $\tau_{\mathbf{M}_\varphi}$ 表示 $\mathbf{M}_\varphi$ 中所有真语句的集合，显然 $\tau_{\mathbf{M}_\varphi}$ 是一可数集。M⊨Γ 表示 M⊨A，A 属于 Γ。

有关有效性(validity)，可满足性(satisfiability)，以及可证谬性(falsifiability)本文将严格使用在[Gall 87]中的定义。因此，本文使用的证明规则是可靠的和完全的。

为简单起见，本文假定知识库中的规则是协调的。

定义 2.1 知识库序列

称任意有穷或无穷的协调语句集 Γ 为一知识库抽象，简称为知识库。被包含在知识库 Γ 中的语句称为规则。

序列 Γ_1，Γ_2，… Γ_n，… 被称为知识库序列，简称序列，如果对任意 n，Γ_n 是一知识库。

如果 $\Gamma_n \subseteq \Gamma_{n+1}$(或 $\Gamma_n \supseteq \Gamma_{n+1}$) 对所有的 n 成立，称序列为增序列(或减序列)；否则称序列是非单调的。

用数理逻辑的术语来讲，知识库抽象是一非逻辑公理集合，即一个形式理论。

本文假定语句 P 及 Q 将被视为同一语句，如果 P≡Q 恒真(也就是(P⊃Q)∧(Q⊃P)是一同义反复)。

定义 2.2 序列的极限

令{Γ_n}为一序列，称语句集：

$$\Gamma^* = \bigcap_{n}\bigcup_{m>n}\Gamma_m$$

为序列{Γ_n}的上极限。称语句集：

$$\Gamma_* = \bigcup_{n}\bigcap_{m>n}\Gamma_m$$

为序列{Γ_n}的下极限。

称序列是收敛的当且仅当 $\Gamma_* = \Gamma^*$。收敛序列的极限是此序列的下极限(或上极限)，并记为 $\lim_{n\to\infty}\Gamma_n$。

下述定理给出了上、下极限的含义：

引理 2.1 ① A∈Γ^*当且仅当存在一子列{k_n}使得 A∈Γ_{k_n}成立。

② A∈Γ_*当且仅当存在 N>0 使得 A∈Γ_m对 m>N 成立。

证明 可从定义直接得出。

定理 2.1 如果序列{Γ_n}是增(或减)序列，则此序列收敛其极限为$\bigcup_{n}\Gamma_n$(或$\bigcap_{n=1}^{\infty}\Gamma_n$)。

例 2.1 增序列

在数理逻辑若干定理的证明中已多次使用增序列的概念。例如，Lindenbaum 定理："L 的任意形式理论都可扩充成一极大理论。"这个定理的证明是这样的：由于 L 中的语句是可数的，因此它们可以排序为：$A_1, A_2, \cdots, A_n, \cdots$，令 $\Gamma_0=\Gamma$，

$$\Gamma_{n+1}=\begin{cases}\Gamma_n \cup \{A_n\} & \text{如果 } \Gamma_n \text{与 } A_n \text{协调}\\ \Gamma_{n+1}=\Gamma_n & \text{如果 } \Gamma_n \text{与 } A_n \text{不协调}\end{cases}$$

这里{Γ_n}显然是增序列，而其极限$\bigcup_{n}\Gamma_n$是一极大理论。

例 2.2 发散序列

$$\Gamma_n=\begin{cases}\{A\} & n=2k-1\\ \{\neg A\} & n=2k\end{cases}$$

显然，$\Gamma^*=\{A,\neg A\}$且 $\Gamma_*=\varnothing$，序列{Γ_n}不收敛。

例 2.3 随机序列

令 A 表示："掷一硬币，得其反面"，Γ_n代表掷第 n 次硬币的结果，序列{Γ_n}就是一关于 A 与¬A 的随机序列。显然，这个序列是不收敛的。

事实上，如果一个知识库的维护序列不收敛，这往往说明我们对所研究问题的特征没有抓准。例如对这个例子，如果 A 代表："掷一硬币，得其反面的概率是 50%"，就抓住了问题的特征。

3 新规则和事实反驳

本节将建立维护序列所需的概念，并给出维护的一般过程。

定义 3.1 新规则

称 A 为关于 Γ 的新规则，如果存在两个模型 **M** 和 **M**′使得

$$\mathbf{M}\models\Gamma,\mathbf{M}\models A \text{ 且 } \mathbf{M}'\models\Gamma,\mathbf{M}'\models\neg A$$

定理 3.1 A 是关于 Γ 的新规则，当且仅当 A 与 Γ 逻辑独立，即 Γ⊬A 及 Γ⊬¬A 均不可证。

证明 可由一阶逻辑的完全性定理直接得出。

本文用 Γ⊢A 表示 A 是 Γ 的逻辑结论[Gall 87]，它表示对任意模型 **M**，如果 **M**⊨Γ 则 **M**⊨A。

推论 3.1 如果 A 是关于 Γ 的新规则，有{Γ,A}和{Γ,¬A}都是协调集。

关于事实反驳的概念可严格定义如下：

定义 3.2 事实反驳

假定 Γ⊨A，称模型 **M** 为一关于 A 的事实反驳，如果 **M**⊨¬A 成立。

令 $\Gamma_{\mathbf{M}}(A)=\{A_i|A_i\in\Gamma,\mathbf{M}\models A_i,\mathbf{M}\models\neg A\}$。

称 M 为关于 A 的理想事实反驳，如果 $\Gamma_{\mathbf{M}}(A)$ 在下述意义下是极大的：不存在关于 A 的另一反驳 **M**′ 使得 $\Gamma_{\mathbf{M}'}(A)\subset\Gamma_{\mathbf{M}'}(A)$ 成立。今后关于 A 的理想事实反驳将记成 M(A)。

由于关于 A 的理想事实反驳不只一个，故定义：

$$\mathbf{R}(\Gamma,A)=\{\Gamma_{\mathbf{M}(A)}|\mathbf{M}\text{ 是 }A\text{ 的理想事实反驳}\}$$

事实上，一个形式理论或知识库是否被承认和接受，既不取决于形式逻辑的推理规则，也不取决于以知识库中的知识为前提的推理过程，它仅取决于所推出的结果是否与事实相符合或与用户的要求一致。事实反驳的概念使我们有可能用数理逻辑的概念刻画这种思想。这也是我们常把这种逻辑理论称为"开放逻辑"的原因[Li 92]。

理想事实反驳的概念是 Occam 原则在逻辑中的一种反映。这个原则认为"Entities are not to be multiplied beyond necessity"[Flew 85]。这句话在这里可以理解为：如果从知识库中推出的某一结论与事实不符，则库中只有导致此结论的极小子集应被更正，而剩下的(与事实协调的极大)子集应予保留，并认为它们至少在当前是正确的。

定义 3.3 可接受更改

令 Γ⊢A，称 A 为 Γ 关于¬A 的可接受更改，如果它是 Γ 的一个与¬A 协调的极大子集。

今后用 A(Γ,A)表示 Γ 关于¬A 的所有可接受更改组成的集合。

定理 3.2 A(Γ,A)≠R(Γ,A)

证明 先证 A(Γ,A)⊆R(Γ,A)

假定 A∈A(Γ,A)，由于它与¬A 协调，故存在 **M**′ 使得 **M**′⊨A 及 **M**′⊨¬A 成立。因此 **M**′ 是关于 A 的事实反驳。**M**′ 必是极大的，因为否则就应存在另一模型 **M**″，使得 **M**″⊨¬A 及 $\Gamma_{\mathbf{M}''}(A)\supset\Gamma_{\mathbf{M}'}(A)\supseteq A$ 成立，由 A 的定义这是不可能的。

例 3.1 令 Γ≡{A,A⊃B,B⊃C,E⊃F}，显然，Γ⊢C 成立。根据定义 A(Γ,C)由下述三个极大子集组成：

$$\{A,A\supset B,E\supset F\},\quad \{A,B\supset C,E\supset F\},\quad \{A\supset B,B\supset C,F\}$$

下述定义告诉我们：当出现新规则和事实反驳时，什麽是对知识库的进行"一次性"修改。

定义 3.4 重构

给定 Γ 及一语句 A，Γ 关于 A 的二种重构定义如下：

如果 A 是 Γ 的推导结果，则 Γ 关于 A 的 E-重构是 Γ 自身。

如果 A 是 Γ 的新规则，则 Γ 关于 A 的 N 重构是{Γ，A}。

如果 Γ⊢A 且 A 受到事实反驳，则 Γ 关于 A 的 R 重构。为△，满足△∈R(Γ，A)。称 Γ′ 为 Γ 的重构，如果 Γ′ 是 Γ 的 E-或 N 或 R-重构。

容易看出 R-重构是不唯一的，而且，如果△是 Γ 关于 A 的 R-重构，则△是 Γ 的关于 ¬A 协调的极大子集。

N 重构及 R-重构[AGM 85]中的扩展和极大选择缩减类似，不同之处在于 AGM 的理论只处理逻辑闭包 Th(Γ)并且只讨论命题演算。此外，AGM 侧重于使用证明论概念，而本文则用模型论概念来刻画用户与知识库的交互作用。更为本质的区别是本文的兴趣在于研究更新的历史，研究知识库序列的极限和怎样构造收敛的序列，以及如何用演算的方法建立重构。

4 维护序列的极限

建立了前两节的基本概念之后，本节将讨论知识库进化的极限问题。

定义 4.1 维护序列

称序列 $\Gamma_1, \Gamma_2, \cdots, \Gamma_n, \cdots$ 是一维护序列，如果 Γ_{i+1} 是 Γ_i 的重构对 $i \geq 1$ 成立。

应该指出，如果 $A \sim (\Gamma_n, A)$ 包含不只一个元素，则 Γ_n 的 R-重构不是唯一的。因此知识库所有可能的历史将是一树型结构，这个树的每一个枝都是一维护序列。

定理 4.1 (1)维护序列{Γ_n}是增(减)序列，当且仅当对任意 $n \geq 1$，Γ_{n+1} 是 Γ_n 的一个 N-重构(R-重构)。

(2)维护序列{Γ_n}是非单调序列，当且仅当序列中 N-重构与 R-重构交替出现。

证明 由定义直接得出。

现在可以严格地定义知识库维护的一般过程了，假定：

(1)M 是一给定的关于某一特定问题的模型，显然其真语句是可数的，真语句集 τ_M 用序列{A_n}表示。

(2)Γ 是一知识库，但它可能与{A_n}不协调。Γ 将被用做维护过程的初始知识库。

构造一般过程的基本思想如下：令 $\Gamma_1 = \Gamma$，而 Γ_{n+1} 将被归纳地定义为：如果 $\Gamma_n \vdash A_n$，则令 $\Gamma_{n+1} = \Gamma_n$；如果 A_n 是 Γ_n 的新规则，则 Γ_{n+1} 是 Γ_n 关于 A_n 的 N-重构，即{A_n，Γ_n}；如果 $\Gamma_n \vdash \neg A_n$，也就是 $\neg A_n$ 受到事实反驳(注意必须接受 A_n)，此时 Γ_{n+1} 是 Γ_n 的关于 $\neg A$ 的 R 重构。

在知识库维护的第 n 阶段，当某一 R 重构被选定用以响应事实反驳时，以下两件事值得提出来进行讨论：

(1)被选中的 R 重构，必须包含在前 n 次维护中被接受的全部新规则，否则 Γ_{n+1} 将会遗漏过去实施维护时取得的成果。因此，需要引入集合△以便搜集在 n^{th} 阶段以前接受的新规则。

(2)即便如此，R 重构被选定后还会丢失信息。例如，令 $\Gamma = \{A \wedge B\}$，则 $\Gamma \vdash \neg A$ 及 $\Gamma \vdash B$ 均成立。假定 A 受到事实反驳，则 Γ 的与 ¬A 协调的极大子集是空集。在这种情况下，R-重构是空集，B 就被漏掉了！为了避免这类事发生，我们引入集合 Θ 用来搜集在前 n 次