国内图书分类号：O141.1 ,TP181

西 南 交 通 大 学

研 究 生 学 位 论 文

基于Petri网模型的归结自动推理研究

年　　级　1999级

姓　　名　夏世芬

申请学位　博　士

专　　业　交通信息工程及控制

指导教师　徐扬 教授

二零零六年十二月

Classified Index: O141.1, TP181

Southwest Jiaotong University

Doctor Degree Dissertation

THE STUDY OF AUTOMATED RESOLUTION REASONING BASED ON MODEL OF PETRI NET

Grade: 1999

Candidate: Xia Shifen

Degree: Ph.D

Specialty: Transportation Information Engineering and Control

Supervisor: Prof. Xu Yang

Dec., 2006

2

西南交通大学学
位论文版权使用授权书

本学位论文作者完全了解有关保留、使用学位论文的规定，同意学校保留并向有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权西南交通大学可以将本学位论文的全部或部分内容编入数据库进行检索，可以采用影印、缩印或扫描等复印手段保存和汇编本学位论文。

本学位论文属于

1、保密□，在____________年解密后适用本授权书；

2、不保密☑，适用本授权书。

（请在以上方框内打“√”）

学位论文作者签名：夏世芬　　　　指导教师签名：

日期：06年12月28日　　　　日期：2006年12月28日

西南交通大学
学位论文创新性声明

本人郑重声明：所呈交的学位论文，是本人在导师指导下独立进行研究工作所取得的成果。除文中已经注明的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的研究成果。对本文的研究做出贡献的个人和集体，均已在文中作了明确的说明。本文完全意识到本声明的法律结果由本人承担。

本学位论文的主要创新点如下：

1. 本文第二章提出的子句集的矩阵表示形式，子句集矩阵的初等变换以及将 T-不变量判定法与一般归结算法相结合的多种矩阵归结算法；
2. 本文第三章提出的基于 Horn 基子句集、一般基子句集以及一阶 Horn 子句集的 Petri 网删除归结算法及算法的完备性；
3. 本文第四章提出的算子命题逻辑系统、算子命题逻辑的 $\lambda-$归结、算子命题逻辑中命题公式的极简规则型、极简规则型命题子句集的 Petri 网模型、基于 Petri 网模型的 T-不变量推理算法和删除归结推理算法以及推理算法的完备性；本文第五章提出的算子模糊逻辑中的提升引理、$\lambda-$归结的完备性、算子模糊逻辑中子句集的两种 Petri 网模型－Horn 型和一般型以及基于这两种模型的归结推理算法及推理算法的完备性；
4. 本文第六章定义的格值命题逻辑系统 LP(X)中公式的规则型范式以及公式的极简规则型子句集，MP 归结，(A,α)-归结演绎，极简规则型子句集的 Petri 网模型，基于 Petri 网模型的 T-不变量归结推理算法以及基于标识的可达性推理算法。

学位论文作者签名：夏世芬

日期：06 年 12 月 28 日

摘 要

知识表示、知识推理和知识应用是人工智能的核心问题。目前，基于非经典逻辑的自动推理系统由于它加速了人工智能的发展而越来越引起人们的广泛重视。为了处理不同信息的推理，人们提出并发展了多种数学理论、方法和工具。如归结原理、经典逻辑、模糊逻辑、格值逻辑、Rough 逻辑以及 Petri 网理论等，这些理论和方法为自动推理提供了强有力的支持。本文在自动推理领域中，进一步发展了现有的理论和方法，研究了基于 Petri 网模型的归结自动推理系统，并取得了如下主要研究成果：

1. 给出了子句集的矩阵表示形式，根据单文字规则与纯文字规则，讨论了矩阵的化简策略，并定义了子句集矩阵的初等变换，研究了变换的性质。根据 Petri 网中 T-不变量推理判定法的思想，并结合输入归结、单元归结以及支撑归结，提出了多种矩阵归结的推理方法，证明了推理算法的完备性。

2. 给出了基于 Horn 基子句集、一般基子句集和一阶 Horn 子句集的 Petri 网模型。根据 Petri 网中标识的流动规则，结合归结原理，给出了基于 Petri 网模型的删除归结推理算法，证明了算法的完备性。并将删除归结推理算法与 T-不变量推理算法进行了比较，得到重要的结论：T-不变量推理算法简单，但使用范围有限，推理过程不直观；删除归结推理算法具有普适性，有直观的推理过程，推理具有高效性，结构表示具有简洁性等优点。这些研究结果为基于 Petri 网模型的归结自动推理提供了简单有效的推理方法。

3. 提出了算子命题逻辑系统，讨论了算子命题逻辑系统中的λ-恒假、λ-归结以及λ-归结演绎等的逻辑性质，证明了λ-归结演绎的完备性。讨论了命题子句的极简规则型范式，给出了算子命题公式的 Petri 网模型，提出了算子命题逻辑系统中的两种归结推理算法－T-不变量推理算法与删除归结推理算法，证明了推理算法的完备性。

4. 针对一类推理模型，提出了一种更一般的算子逻辑系统即算子模糊逻辑系统，讨论了算子模糊逻辑系统中的λ-归结以及λ-归结演绎的逻辑性质，给出了算子模糊逻辑系统中的提升引理，由此证明了λ-归结演绎的完备性。讨论了算子模糊逻辑系统中算子模糊子句的 Horn 型及一般型两种 Petri 网模型，给

出了基于表示模型的 T-不变量推理算法和删除归结推理算法，证明了推理算法的完备性。所有这些表示方法和推理算法，为模糊逻辑系统的推理提供了新的方法与思路。

5. 为了进一步研究非经典逻辑系统的自动推理，特别是较直观的推理形式，给出了格值命题逻辑系统 LP(X)中公式的 Petri 网模型及基于 Petri 网模型的归结推理算法。讨论了格值命题逻辑系统 LP(X)中公式的规则型范式，定义了极简规则型子句的 MP 归结以及(A,α)-归结演绎，证明了归结演绎的可靠性与弱完备性。讨论了极简规则型子句集的 Petri 网模型以及任意公式的 Petri 网模型。详细给出了四值非链格值命题逻辑系统 $LP_4(X)$的 Petri 网推理模型及推理算法，进一步给出了格值命题逻辑系统 LP(X)中公式的 Petri 网推理模型以及基于 Petri 网模型的 T-不变量推理算法和可达性推理算法，证明了推理算法的完备性，得到了有意义的结论，从而为研究格值命题逻辑系统中的自动推理提供了切实可行的方法。

关键词：Petri 网；归结推理；算子命题逻辑；算子模糊逻辑；格值命题逻辑

Abstract

There is no denying that representation, reference and application of knowledge are key questions in artificial intelligence. In recent years, automated deduction based on non-classical logics, which speed up the development of artificial intelligence, has attracted many researcher's attention. In order to deal with inference of different information, people has proposed and developed many mathematical theories, methods and tools, such as theory of resolution, classical logic, fuzzy logic, lattice- valued logic, Rough logic, Petri net and so on. These theories and methods play an important role in providing support for automated deduction. This paper further develops these theories and method, studies resolution inference based on the Petri net, and achieves the following research results:

1. The matrix representation of base clause set is presented and the method of simplification for matrix is discussed according to single literal rule and pure rule. Furthermore, based on T- invariant and the completeness of input resolution, unit resolution and supporting resolution, we give the definition of elementary transformation for matrix, discuss its properties, provide many methods of matrix inference and prove the completeness.

2. Models of Petri net are given for base Horn clause set, general base clause set, and one-order Horn clause set. Then, based on the flowing rule of token in Petri net and theory of resolution, the canceling algorithms and their completeness are presented. After that, we have got important conclusions by comparing T-invariant algorithm with the canceling algorithm. For example, we have got the facts that the T-invariant algorithm is simple but has limited range for application, the process of inference is not visualized, and the canceling algorithm can be widely and efficiently applied with its visualized process and brief structure. All these results are efficient and helpful for the automated inference based on Petri net.

3. We first provide the system of operator proposition logic. Then, λ -resolution and λ -resolved deduction are given. Also, their properties are discussed and

completeness is proved. Furthermore, the Petri net model of operator proposition clauses is discussed based on the extra simple ruled clause set of proposition clause, and the inference algorithms of T-invariant and canceling resolution are presented.

4. As the extension of the proposition logic, an operator fuzzy logic is proposed, many logical properties are discussed. Then, we give the definitions of λ -resolution and λ -resolved deduction, discuss their properties and give the proof of completeness for the lifting lemma of operator fuzzy logic. To realize the λ -resolution reasoning concretely, the Horn and general Petri net models for fuzzy clauses are discussed, and the inference algorithms of T-invariant and canceling are given. All these will provide new method and idea for deduction of fuzzy logic.

5. In order to further study the automated deduction of non-classical logics directly, this paper has discussed the Petri net models and inferring algorithms for system of lattice-valued proposition logic system LP(X). First, we discuss the ruled and the extra simple normal form of formula in LP(X). Then, we give the definitions of MP resolution and (A,α)—resolution inference, prove the soundness and weak completeness. Also, we discuss the Petri net models. As an example, the four-valued non-line lattice-valued proposition logic system is studied in detail. After that, the reasoning model and the inferring algorithms－T-invariant and reachable inference are given, their completeness are proved , we also get meaningful conclusions which will be more helpful to find useful method for lattice-valued proposition logic system LP(X).

Key words: Petri net; Reasoning of resolution; Operator proposition logic; Operator fuzzy logic; Lattice-valued proposition logic

目 录

第1章 绪论

人工智能是二十世纪科学发展的一朵奇葩，其思想与方法已经渗透到信息论、控制论、计算机科学、工程技术等领域，对科学的发展产生了深远的影响。自动推理及定理机器证明是人工智能理论和技术中十分重要的研究课题，本文的工作属于这一范畴。本文主要讨论基于Petri网的算子命题逻辑系统、算子模糊逻辑系统和格值命题逻辑系统中自动推理方法的研究，为进一步研究基于非经典逻辑的定理自动证明及其应用软件的设计提供了一定的理论和方法。现对本文所涉及的相关研究的进展情况作一个简要概述。

1.1 形成本文的学术背景

通过机器实现模仿人的行为，使之具有人类的智能是人工智能研究和探索的目标。从1956年正式提出人工智能科学到现在，已走过了50年的历史，这期间，人工智能产生了符号主义、联结主义及行为主义三大主要学派。

符号主义又称逻辑主义，其原理主要为物理符号系统假设。它认为，人工智能源于数学逻辑，认为人的认知基元是符号，而且认知过程即符号操作过程，人是一个物理符号系统，计算机也是一个物理符号系统，因此，我们就能够用计算机模拟人的智能行为，即用计算机的符号操作来模拟人的认知过程。它还认为，知识是信息的一种形式，是构成智能的基础，人工智能的核心问题是知识表示、知识推理和知识应用。知识可用符号表示，也可用符号进行推理，因而有可能建立起基于知识的人类智能和机器智能的同一理论体系。人工智能的研究方法应为功能模拟方法，通过分析人类认知系统所具备的功能和机能，然后用计算机模拟这些功能，实现人工智能。

联结主义又称仿生学派，其原理主要为神经网络及神经网络间的连接机制与学习算法。联结主义认为人工智能起源于仿生学，特别是人脑模型的研究。认为人的思维基元是神经元，而不是符号处理过程，对物理系统假设持反对意见，认为人脑不同于电脑，并提出联结主义的大脑工作模式，用于取代符号操作的电脑工作模式。认为人工智能应着重于结构模拟，即模拟人的生理神经网络结构，并认为功能、结构和智能行为是密切相关的，不同的结构表现出不同

的功能和行为。

行为主义又称进化主义或控制论学派，其原理为控制论及感知-动作型控制系统。它认为，人工智能起源于控制论，智能取决于感知和行为，提出智能行为的感知-动作模式。它还认为，智能不需要知识、不需要表示、不需要推理，人工智能可以像人一样逐步进化，智能行为只能在现实世界中与周围环境交互作用而表现出来。

符号主义可以较好地模拟人的逻辑思维，但由于其机制固定于知识与推理，有很大局限性，往往对“常识性”问题的处理并不能见效；而联结主义则可以较好地模拟人的形象思维，但由于固定的体系结构与组成，所构成的系统达不到开发形形色色变化的知识的要求。在模拟高层次思维方面，如基于目标的推理、剖析及因果分析等方面，则性能远不如符号主义。因此将符号主义与联结主义有机地结合起来，则是人工智能研究的一个重要方法。

随着人工智能的发展，三大学派相互渗透，取长补短，共同应对复杂多变的系统。模糊控制、智能控制、神经网络控制以及模糊神经网络、神经网络推理等研究领域正引起人们的广泛兴趣。虽然各学派在人工智能的基本理论、研究方法和技术路线等方面有着不同的观点，但他们的目的是相同的，都是研究如何模仿人的智能，实现机器智能，造福人类。符号主义、联结主义和行为主义已开始携手合作，走向集成，共同攀登人工智能的新高峰。

在人工智能的众多课题中，逻辑的研究是人工智能的中心课题之一。一方面，人工智能的研究为逻辑学的研究提出了新的课题；另一方面，逻辑学的研究成果也促进了人工智能的发展。

经典逻辑以其严谨的逻辑结构和精确的推理体系在处理非此即彼的确定性信息的过程中取得了巨大的成功。然而在处理现实世界中大量存在的不确定性信息的过程中，却显得力不从心。因为经典逻辑的真值域为{0，1}，描述的是一种边界确定、非此即彼的清晰概念，它丢失了事物的渐进性和事物发展过程中本身具有的多态性。为有效地刻画事物本身具有的渐进性和多态性，非经典逻辑应运而生。非经典逻辑突破了真值域{0，1}的限制，更适合处理不确定信息。1920 年，J.Lukasiewicz 拓展了真值域{0，1}，提出了包含不分明(Vague)状态 u 的三值逻辑，随后又将三值逻辑系统推广到了 n 值及无穷值，以后又出现了 Kleene 强三值逻辑系统。1965 年 L.A.Zdeh 首先发现并阐明了模糊集合的概念，并引入隶属函数来描述各种中间过度状态，这种全新的数学命名为模糊

数学。在模糊逻辑中，命题的真值域{0，1}被拓展为连续变化的区间[0，1]。格值逻辑是一种重要的非经典逻辑。格是一类重要的代数结构，现实世界的许多现象都可以用格来刻画。1993年徐扬将代数格与蕴涵代数相结合，建立了格蕴涵代数。随后，徐扬、秦克云等进一步建立了基于格蕴涵代数的格值命题逻辑系统 LP(X)，格值一阶逻辑系统 LF(X)。为刻画"两个层次"(对象本身与对于对象处理过程)的模糊性与不可比较性，1999 年后，徐扬等建立了一种分层的格值命题逻辑系统 L_{vpl} 和分层的格值一阶逻辑 L_{vfl}。

建立了以上的逻辑基础，就可以着手人工智能的另一异常活跃的研究方向—定理机器证明和自动推理。从1965年至今，许多学者致力于此领域的研究，提出了许多方法。其中，有 J.A.Robinson 于 1965 年提出的归结方法因其简洁有效而受到广为重视。然而，由于归结过程是随机选取子句进行归结，会产生大量的冗余子句和重复归结式，因此效率不高。于是，人们提出了各种各样的归结策略来对归结原理进行改进，主要目的是减少冗余子句的产生和重复地进行归结。具有代表性的有支撑归结、线性归结、输入归结、单元归结、有序归结和锁归结等。对于基于经典逻辑的定理机器证明和自动推理，人们已进行了大量的、卓有成效的研究。目前，基于非经典逻辑的定理机器证明系统的研究加速了人工智能的发展而正越来越受到人们的广泛重视，而且其它数学工具如 Petri 网、神经网络、证据理论、粗集理论等正向定理机器证明系统渗透，以期研究基于非经典逻辑的归结理论与方法。

本文就是在这样的背景下产生的。作者试图以 Petri 网作为工具，研究基于 Petri 网的算子命题逻辑系统、算子模糊逻辑系统以及格值命题逻辑系统 LP(X) 上的归结原理与方法，从而为不确定推理和软定理机器证明提供一种行之有效的方法。

1.2 国内外研究现状

人工智能自问世以来，在自动推理技术领域取得了很大的进展。

基于逻辑系统研究推理是对推理研究的科学方法之一。确定性问题处理需要确定性推理，确定性推理的合理性基于经典逻辑的科学性。传统的推理技术是以经典的谓词逻辑为基础的、精确的、严格的推理技术，属于确定性推理的范畴。传统的推理方法是从已知的事实出发，运用一阶谓词演算的解析方法，

演绎出逻辑上蕴涵的结论。20 世纪 50 年代末 60 年代初的机器定理证明主要采用传统的推理技术[1]。典型的代表是 1965 年建立在 J.A.Robinson 归结原理基础上的定理机器证明[2]，这是人工智能最早采用的一种推理模型。但是，人工智能需要研究更接近人类思想规律的推理模型，而人们总是在不完全的知识环境下进行推理，信息不准确、信息不完全、模糊信息、不确定性假设等因素造成推理过程的不确定和非单调性，而传统的一阶谓词逻辑不能解决这些问题。20 世纪 70 年代后期以及 80 年代，随着人工智能求解问题的复杂化，推理技术得到迅速发展，开始了不确定性推理、非单调推理、定性推理方法的研究。

不确定推理方法是 20 世纪 70 年代提出并开始研究的。典型的不确定推理模型有：基于可信模型的确定性推理，基于主观概率模型的主观概率法，基于概率指派函数 m、信任函数 Bel、和似然函数 Pl 的证据推理[3,4]，基于条件概率的信念网络[5,6]，基于语义记录、结构化知识表示的批注理论，以关联式、发生率为逻辑基础的关联式方法，以及以模糊逻辑、可能性理论、隶属函数为基础的模糊推理。

对于非单调推理的研究始于 70 年代，对常识、歧义性等的讨论，导致了知识表示的研究，非单调逻辑有效地改善了传统逻辑描述不完备知识时的不足。典型的模型有基于信任集和假设集的假设推理模型，基于谓词逻辑的谓词完备化方法，以非单调逻辑为逻辑基础的缺省推理法(default reasoning)[7]，基于最小语义非单调逻辑的限制推理(Circumscription)[8]，非单调逻辑推理[9]等。

为简化问题的描述，1977 年 Reiger 发表了第一篇定性推理的论文，提出了处理问题时，忽略被描述对象的次要因素，掌握主要因素进行定性推理的思想。典型的推理模型有：定性方程推理法，定性进程法，定性模拟法以及代数系统法。

下面，我们详细介绍文中涉及的推理模型的研究概况。

1.2.1 基于模糊性的不确定推理研究概况

1965 年，L.A.Zadeh 发表了他的著名论文“Fuzzy Sets”[10]，标志“模糊数学”的诞生。以模糊逻辑作为逻辑基础的模糊推理的研究是从 1975 年 L.A.Zadeh[11]提出推理的合成规则(CRI)开始的，其中有代表性的主要有：Zadeh(1975)的推理合成规则、Baldwin(1979)[12]的 CRI 与语言真值蕴涵相结合的

真值限定法、Mizumoto(1985)[13]的扩展原理与语言真值蕴涵相结合的推理方法、Tsukamoto(1979)[14]的扩展原理与语言真值 Lukasiewicz 蕴涵相结合的推理方法和 Yager(1978)[15]基于相似性度量提出的推理方法。基于这些方法，人们进一步提出了多重推理方法、多维推理方法、多重多维推理方法，以及各种各样的改进方法，其中主要有：Sugeno , Takagi(1983)多维模糊推理方法、陈永义，陈图云(1984)的特征展开近似推理方法、Zadeh(1985)的量化命题模糊推理方法、汪培庄(1989)的真值流模糊推理方法、吴望名(1990)的多重模糊推理方法、Chen(1992)基于模糊矩阵与事实向量的模糊推理方法，Bien(1994)的基于模糊集合的双向模糊推理网络，Dubois(1995)[16]的基于模糊关系方程的模糊推理方法，张文修(1996)[17]的数值模糊推理方法，Gorzalczany(1987)等[18~23]广泛研究了区间值模糊推理方法。1996 年 Takagi 等[24]利用多层概念模糊集提出了一个多层模糊推理方法。1996 年 Raha 等[25]研究了与时间相关的不确定性推理，提出了一种表达和处理时态变化世界中模糊概念的方法，1997 年 Raha 等[26]又对基于模糊逻辑处理不分明缺省值的缺省推理方法进行了研究，并于 1999 年提出了表达和处理部分真实知识的不确定性推理方法[27]。1999 年，Emami 等[28]通过将 Mamdani 近似推理和形式推理相结合提出了一个统一参数化的形式推理方法，Kuhu 等[29]提出了关于模糊推理的神经网络体系结构，王国俊[30]利用蕴涵算子 R_0 研究了全 3-I 模糊推理方法，2002 年，宋士吉等[31]又研究了反向 3-I 模糊推理方法。2000 年，Chen 等[32]研究了基于规则系统的双向不确定性推理方法，Yager 等[33]提出了一种基于模糊集合论、处理知识冲突的推理机制。2003 年 Niittym 等[34]进一步研究了基于相似度的模糊推理方法。从 1988 年后，徐扬等先后提出了基于属性描述的推理方法、基于集映射的推理方法、基于推理原则的多重推理方法和多维推理方法[35]，基于广义扩展原理的模糊推理方法(2001)[36]，基于广义 IF-THEN 规则的模糊推理方法(2002)[37]。

在模糊逻辑推理的研究中，一个重要且很有前途的方向是基于严密的逻辑基础建立相应推理理论和方法。徐扬等(2000)[38]在 Pavelka[39]和 Novak[40]工作的基础上，研究了基于格值命题逻辑系统 L_{vpl} 的不确定性推理，进一步研究了基于格值一阶逻辑系统 L_{vfl} 的不确定性推理，并讨论了逻辑系统 L_{vfl} 的语义[124]以及基于格值一阶逻辑系统 LF(X)的规则库的相容性。马骏(2004, 2006) 讨论了系统 LF(X)的处理语言项的模型[129]以及从机器智能的观点讨论模糊逻辑[130]， 陈树伟

(2004, 2005) 讨论了系统 L_{vfl} 的不确定性推理以及格值逻辑中的语言值不确定性推理， 秦克云(2003)对滤子和推理规则的关系进行了研究[131]，李海明(2005, 2006) 对格值推理策略以及模糊正则的隶属度表示进行了研究[135, 136]，Sofronie-Stokkermans,V.(2001)研究了格值逻辑的语义和表示定理[142, 143]， Lu, James J. (2005)研究了正则多值逻辑的推理和搜索策略[144]。随着人工智能的不断向前发展，对模糊逻辑特别是多值逻辑的研究必将达到一个新的热潮。

1.2.2 基于归结原理的自动推理的研究概况

从 1965 年 Newell, Shaw 和 Simon 发表他们的著名论文《逻辑理论机》(The Logic Theory Machine) 算起，自动定理证明(ATP)这个领域已经有四十多年历史了。1959 年，Gelernter 等人做出了几何定理证明机(GTM)，在 GTM 产生的同时，出现了在逻辑上建立定理证明机器的研究工作(Paul Gilmore 和王浩)。定理自动证明的目的是寻求逻辑公式有效性的机械的判定方法。K.Gödel 于 1931 年得到的不完全性定理表明这样的算法是不存在的。A.Church 和 J.B.Rosser 对 Gödel 的不完全性定理进行了改进，同时指出：存在一些算法，如果逻辑公式是恒真的，则该算法可以在有限步内给出判定，即半可判定算法。因此定理自动证明就是寻求这样的半可判定算法。1930 年 Herbrand 提出了著名的 Herbrand 定理，首先给出了一个这样的判定算法，将谓词演算中的定理证明转化为命题逻辑公式的可满足性问题。Herbrand 算法奠定了自动定理证明的基础。1959 年，P. Gilmore 发表的证明程序，是对谓词演算的第一个可用的机械证明程序。

1958 年到 1960 年期间，王浩研制出的定理自动证明的三个程序，可以高效地证明《数学原理》中的 220 个定理。1960 年，M.Davis 和 H.Putnam 提出 DP 预处理方法，对 Gilmore 的判定方法进行了改进，有效地提高了判定算法的执行效率。1960 年，D.Prawitz 采用人工智能中的匹配技术，将消极的枚举基例来产生不可满足的基例集变为主动寻找产生恒假公式的替换，避免了组合爆炸，但是结果并不理想。

1965 年，J.A.Robinson[41]提出了著名的归结原理。归结原理在定理自动证明的研究中具有重要意义，为定理自动证明找到了一个简便可行的方法。由于它形式简洁，推理可靠、完备，因此受到各国学者的广泛关注。因此，许多研究者投入到归结原理的改进工作中，重要的改进工作有：1965 年，L.Wos， J.A.Robinson 和 D.Carson[2]提出了支撑集策略。1965 年，J.A.Robinson 提出超归结方法。1967 年，J.R.Slagle[42]提出了语义归结原理并证明了它的完备性。1970 年，Boyer 提出锁归结。1970 年，D.W.Loveland[43]和 D.Luckham[44]提出的线性归结。1970 年，C.L.Chang 提出的输入归结，其基本思想都是采用限制参与归结的子句类型，从而达到减少冗余子句产生的目的。1979 年，王湘浩和刘叙华提出了广义归结方法[45]。其后，刘叙华、孙吉贵、曾云峰和刘瑞胜等人进一步提出了广义锁归结、广义线性归结、广义语义归结等方法[45]。N. V. Murry[46]在 1982 年提出了 NC-归结。刘叙华和安直[45]在 1985 年提出了广义调解法。1986 年王元元提出了广义替换归结方法[45]。1990 年~1992 年，刘叙华、孙吉贵等[45]提出了广义 RUN-NRF 归结方法，NC-调解法，广义 NC- RUN-NRF 归结方法。另外，C.L.Chang[47]、L.Henschen、L.Wos[48]、陆汝钤[49, 50]、刘叙华、孙吉贵、扬凤杰、欧阳丹彤和欧阳继红等人[51~60]对 Horn 集上的归结方法都做了许多深入的探讨。

上述归结方法都是在经典逻辑的基础上进行的。随着非经典逻辑的研究日益深入，基于非经典逻辑的归结方法也迫切需要发展。

1976 年，C.G.Morgan[61]首先提出了一个有限多值逻辑的归结原理。之后，1978 年 E.Orlowska[62]又给出了一个ω^+-值逻辑中的归结自动推理，1985 年，Orlowska[63]又给出了一个基于有限值 Post 逻辑的归结自动推理。1986 年，P.H.Schmitt[64]提出了一个基于三值逻辑的归结自动推理。这些归结自动推理有一个共同点是引入 Signs 公式，表示为 S:F，其中 F 是一个公式，S 被称为 Sign，它可以是真值域中的单元素或多个元素的集合，由此便可将二值逻辑的手段用于多值逻辑。与此相关联的是 1989~1993 年，H.A.Blair，V.S.Subrahamnian[65]，N.C.A.da Costa[66]，M.Kifer，E. Lozinskii[67~69]，J.J.Lu，L.J.Henschen，N.V.Murray，E.Rosenthal 等[70, 71]人研究了 Annotated 逻辑与 Sign 逻辑，证明了 Annotated 逻辑中的归结等价于特殊的 Sign 逻辑归结，Sign 逻辑是算子模糊逻辑的推广。与上述子句集类型归结方法相对应的是非子句集类型的归结方法(或称为广义归结)。N.V.Murray[72](1991 年)和 Z.Stachniak[73, 74](1988 年~1996 年)分别构造了多值逻辑的广义归结自动推理方法，其中包含了很多多值逻辑系统的归结自动推理方法。

1972 年，R.C.T. Lee[75, 76]基于[0,1]上的模糊逻辑建立了一个归结自动推理系统，开创了模糊逻辑中自动演绎的工作，1975 年~1982 年，M.Mukaidono[77~79]

等人引入了模糊归结式，推广了 Lee 的工作。1985 年，R.R.Yager 建立了多值命题逻辑的归结自动推理，其中的蕴涵采用模糊蕴涵。1986 年，J.P.Benejam[80] 应用公式的分解规则提出了一种基于 Lukasewicz 蕴涵的[0,1]上的与语法归结方法对称的归结自动推理方法。1989 年，提出了算子模糊逻辑中的λ-归结原理[56, 57]。1995 年，刘叙华和司徒芊[58]又提出了广义λ-调解法。

除上述关于模糊逻辑中归结原理的研究工作之外，1982 年，Farinas del Cerro 将经典逻辑的归结推理用于模态逻辑[81]，1987 年，D.Dubois 和 H.Prade 提出了可能性归结。1990 年，陈永义[82]提出了概率逻辑中的归结原理，并进一步提出了一种可适用于概率推理、模糊推理、信度推理和证据推理的软归结原理。2000 年~2001 年，徐扬等[83, 84, 124]讨论了基于格值命题逻辑系统 LP(X)和格值一阶逻辑系统 LF(X)的归结问题，提出了α-归结原理的概念，并讨论了α-归结的性质，得到了α-归结原理的可靠性和完备性定理，刘军等(2003)讨论了格值逻辑系统中的类-归结策略以及模糊逻辑中的归结[127, 128]，马俊(2002)提出了利用滤子进行归结的原理及方法[85]，孟丹(2002，2003)讨论了含中介元的格值命题逻辑以及有限链命题逻辑的归结原理[132~134]，李文江(2004) 讨论了格值模态命题逻辑中的语义和归结方法[137, 138]，王伟(2003)[86]针对归结中要使用的不可分极简式的形式进行了探讨，并研究了格值命题逻辑系统中的α-归结域，孟丹(2004)研究了格值逻辑系统中的矩阵归结算法[87]。其它具有代表性的有：Guller(2006) 的布尔格上的二元归结[141]，Cser, Laszlo(2004)的自动归结[145]，Hans Kleine Buning(1995)的带谓词的布尔逻辑公式的归结[146]，刘清等(2004)的 G-逻辑及其归结推理[147]。

1.2.3 关于 Petri 网与人工智能研究概况

Petri 网是由德国科学家 C.A.Petri 教授于 1962 年在其博士论文《用自动机通信》(Communication with Automata)中首先创立的[123]，它是异步并发系统建模与分析的一种重要工具[111]，现已成为计算机界和自动化界的热门研究课题。

自 Petri 教授开创性的工作之后，Petri 网得到了长足进展，至今形成了相当规模的研究领域。在理论方面，首先建立了很多分析技术[88, 89]，主要包括基于状态方程的代数分析技术[112~114, 118~121]、基于可覆盖树(图)的图分析技术[90, 91]、基于化简分解的归纳分析技术[92, 93, 115~117]。代数分析技术主要是以关联矩阵和不

变量分析为主，其中关联矩阵是 Petri 网的一种静态结构，描述的是基网的性质，与标识无关，而 Petri 网的 P-不变量反映了部分位置中标记总量的一种加权守恒性，利用 P-不变量可以研究互斥行为、死锁分析和错误识别等。Petri 网的 T-不变量反映了使状态回归的可能变迁序列，利用 T-不变量可以研究周期性、循环性等性质。代数分析以关联矩阵形式对一个网系统的结构给予刻画，然后建立状态可达的线性系统关系，这种分析途径首先是 Peterson[88]提出，之后 Murata[89]的工作最为出色。它的优点在于可以借助线性代数的有关结果，简洁地展现 Petri 网的一些性质。可达树是系统分析最基本和最简单的方法，在可达树上节点对应可达标识，弧对应可实施的转移结点，用可达树可以判断一个 Petri 网是否安全、活的和可逆转的。Petri 网的另一主流方向是建立在通用网络基础上的并发行为的特性研究。通用网络是从更为基础、更为抽象的网络模型上讨论系统的特性研究，其中最为重要的是并发性。Petri 网语言也是 Petri 网中一个重要的研究方向。

人工智能的本质是研究如何制造出人造机器或用智能系统模拟人类智能活动的能力，以延伸人们智能的科学，即对提出的问题进行推理求出问题的解。一般步骤是根据知识库中存放的知识(事实和规则)，对用户提出的具体问题，选择有关的知识，控制求解步骤，执行知识库的规则，逐步获得问题的解答。Petri 网利用网络图形描述对象之间的输入输出关系，能很好反映系统的静、动态特性。Petri 网的动态特性通过标识(token)的移动和传播进行控制，一般地，规则的前提和结论子句与 Petri 网的位置结点对应，一条规则与一个转移结点对应；一旦规则的前提条件得到满足(输入位置结点有一个标识)，当规则被激活(转移点火)时，可确认该规则的结论(标识置于输出位置结点)。在这种转移原理的基础上，目前对于一个给定的规则，仍存在多种 Petri 网表达形式，规则的 Petri 网模型表达的正确性和检验错误的能力是人们选择 Petri 网表达形式的标准。文献[94][95]用 Petri 网描述 Horn 子句的逻辑推论，文[96][97]将 Petri 网应用于专家系统的一致性问题的检查，文[98]应用于非单调逻辑推论。Murata 等[99]给出了从一组 Horn 子句转换成一个 Petri 网模型的算法过程，建立了一组 Horn 子句不一致性的必要和充分网论条件。林闯等[94]给出了 Horn 子句 Petri 网模型的 T-不变量算法以及化简方法。对于相对来说比较模糊的人工智能的规则库，人们正在探索用一种扩展的 Petri 网来描述即模糊 Petri 网[100]。Looney 与 Alfize 等[101,

[102]提出了针对命题型模糊推理的规则矩阵方法，Chen 等[103]用模糊 Petri 网表示模糊合成规则，Scarpelli 等[104]讨论基于高阶模糊 Petri 网提信息，Konar 和 Mandal[105]对不清晰、不确定、不相容的推理专家系统给出了两种模糊 Petri 网模型，Cao 和 Sanderson[106]讨论模糊 Petri 网对机器人系统作任务序列规划，Bugarin 和 Barro[107]用模糊 Petri 网模型化模糊知识基系统，并通过该模型，提取规则，Ahson[108]用加强时序 Petri 网模拟神经网络，提出模糊神经网络的 Petri 网模型，文[109]基于人工神经网络提出高阶 Petri 网模型，文[110]将模糊 Petri 网应用于模式识别中，Lin, Chuang(2004)对基于时间 Petri 网的推理系统进行了定量和定性的分析[148, 149]，Li, Xiaou(2000)讨论了自适应模糊 Petri 网的知识推理和学习[150]，Looney(2003)给出了模糊信度 Petri 网的推理，Egorou, A.F.(2002)给出了模糊区间的不确定性模糊推理模型[152]，Lee, Jonathan(2003) 给出了概率 Petri 网的不确定性推理模型[153]，Aura, Tuomas(2000)讨论了时间 Petri 网语义[154]，Liang, Lily R.(2004) 给出了模糊信度推理的判定[155]，Shyi-Ming Chen(2000, 2004)讨论了模糊 Petri 网的模糊向后推理以及加权模糊 Petri 网的加权模糊推理[156, 157]。总之，Petri 网已经深入到人工智能的许多领域，正在不断推进人工智能向前发展。

1.3 本文的研究工作

Petri 网虽然有较完善的分析技术，可在不同的系统中，Petri 网的表示模型是不同的，没有固定的模式，并且对于逻辑系统的研究而言，利用 Petri 网模型研究逻辑特别是多值逻辑系统还只是处于起步阶段，没有形成系统的研究方法。本文探讨多值逻辑系统特别是算子命题逻辑系统、算子模糊逻辑系统及格值命题逻辑系统 LP(X)的 Petri 网模型与基于 Petri 网模型的归结自动推理方法，以全新的角度为自动机器证明提供一种思路。

本文的研究工作由五部分内容组成：

一、对矩阵归结进行了研究。给出了子句集的矩阵表示形式，定义了子句集矩阵的几种化简策略以及初等变换，然后基于化简与支撑策略，给出了一般基子句集的矩阵归结算法，针对 Horn 基子句集，讨论了基于 T-不变量的几种矩阵归结算法，证明了矩阵归结算法的完备性。(第二章)

二、讨论了 Petri 网的删除归结算法。对于基子句集以及一阶 Horn 子句集，

证明了删除归结算法的完备性，并将删除归结算法与 T-不变量推理算法进行了比较，得到了有意义的结论。(第三章)

三、提出了算子命题逻辑系统，讨论了算子命题逻辑中的逻辑性质，定义了算子命题逻辑中的λ-归结，证明了λ-归结的完备性。通过算子命题逻辑中公式的极简规则型子句集的表示，给出了算子命题逻辑中子句集的 Petri 网模型，讨论了算子命题逻辑的 T-不变量推理算法和删除归结推理算法，证明了推理算法的完备性。(第四章)

四、提出了算子模糊逻辑系统，分析讨论了该系统的逻辑性质，定义了λ-归结以及λ-归结演绎，证明了算子模糊逻辑中的提升引理以及λ-归结的完备性。在讨论算子模糊逻辑的逻辑性质的基础上，给出了算子模糊逻辑公式的两种 Petri 网模型即算子模糊 Horn 子句的 Petri 网模型和一般算子模糊子句的 Petri 网模型，并分别讨论了基于两种模型的推理算法－T-不变量归结推理法与删除归结推理算法，证明了推理算法的完备性。(第五章)

五、针对格值命题逻辑系统 LP(X)的特点，定义了 LP(X)中公式的规则型范式以及极简规则型子句集，在极简规则型子句上定义了 MP 归结、(A,α)—归结演绎、A-不可满足以及α-逻辑推论等概念，根据 MP 归结定义(A,α)—归结演绎能将归结自动推理与不确定性推理建立一定的联系。文中证明了语义推导与语法推导的可靠性以及弱完备性，并对 LP(X)中公式的 Petri 网模型进行了研究，提出了广义位置结点的概念，进一步对 LP(X)系统中基于 Petri 网模型的自动归结推理进行了判定。讨论了 LP(X)系统的 Petri 网模型上的 T-不变量归结推理与可达性推理，证明了推理算法的完备性。(第六章)

1.4 预备知识

本节简要介绍 Petri 网、格值命题逻辑系统 LP(X)中的一些概念和性质，作为本文的准备。

1.4.1 Petri 网有关的概念与命题

普通的 Petri 网主要有两种结点，即转移结点(变迁，transition)和位置结点(place)，分别用竖线和圆圈表示。Petri 网中有两种连线：输入连线，指从转移

结点到位置结点的连接线，反向为输出连接线。一个 Petri 网就是用这两种结点和两种连线互连成的图。例如图 1-1 就是一个 Petri 网。

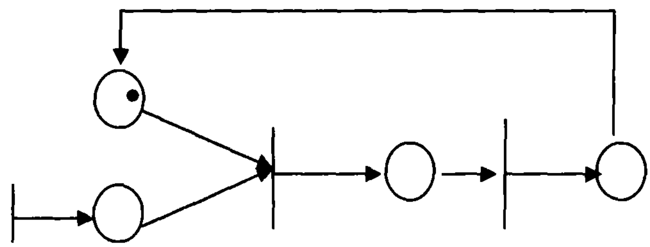


图 1-1 一个 Petri 网

Petri 网是一个可运行的图，在运行开始时，先在某些位置结点中放置一些标识(token)，用圆点表示。只有当一个变迁结点的所有输入结点中至少有一个标识时，它才可能被点火(激活)，点火的结果是从它的每一个输入结点中各减去一个标识，并向它的所有输出结点各增加一个标识。如此一步一步地运行，一些转移结点不断点火，一些位置结点中的标记数不断随着变化。所以 Petri 网很适合描述具有并发或并行行为或操作的各种系统。

定义 1.4.1[95] 命题逻辑的 Petri 网(P/T 网)是一个五元组(P，T，F，M_0，W)，其中：

(1) (P, T, F)是一个有限网，P 是位置集合表示命题集，T 是转移集合表示公式集，F 是弧集合。

(2) 标记 M：$P\rightarrow N\cup\{0\}$，表示标识在位置中的分布。记 M_0 为初始标识。

(3) W：$F\rightarrow\{1\}$是弧的权函数。

定义 1.4.2[95] 谓词逻辑的 Petri 网(Pr/T 网)是一个九元组Σ=(P, T，F，D，V，A_P , A_T , A_F , M_0) ，其中：

(1) (P, T，F)是Σ的基网，P 是位置集合，T 是转移集合，F 是弧集合；

(2) D 是Σ的个体集，D 上有给定的运算符集 Ω;

(3) V 是 D 上的变量集；

(4) A_P：$P\rightarrow\Pi$，其中 Π 是 D 上的可变谓词集。对 $p\in P$，若 $A_P(p)$是 n 元谓

词，就说 p 是 n 元谓词；

(5) A_T：T→ f_D，其中 f_D 是 D 的公式集。对 t∈T，$A_T(t)$只能包含静态谓词和 Ω 中的运算符；

(6) A_F: F→ f_S，其中 f_S 是 D 的符号之和集，使得对 n 元谓词 p∈P，若(t,p)∈F 或(p,t)∈F，则 $A_F(t,p)$或 $A_F(p,t)$为 n 元符号和；

(7) M_0：P→ f_S，对于 n 元谓词 p∈P，$M_0(p)$是 n 元符号和。M_0 是个体集 D 在位置集 P 上的一个分布。

定义 1.4.3[100] 模糊 Petri 网模型(简记为 FPM)为一个 12-元组 FPM =(P, T, F，D，V，π，A_P , A_T , A_F , f，α ，M_0)，其中：

(1) (P, T，F)是 Σ 的基网，P 是位置集合，T 是转移集合，F 是弧集合，满足： $F \subseteq (P\times T)\cup(T\times P)$，$P\cap T=\phi$(空集合)，$P\cup T\neq\phi$。

(2) D 是个体集，D 上有给定的运算符集 Ω;

(3) V 是 D 上的变量集；

(4) π 是 D 上的可变谓词集；

(5) A_P：P→Π，对 p∈P， 若 $A_P(p)$是 n 元谓词，就说 p 是 n 元谓词；

(6) A_T：T→ f_D,其中 f_D 是 D 的公式集。对 t∈T, $A_T(t)$只能包含静态谓词和 Ω 中的运算符，f_D 可以看作是对 t 引发的一种约束；

(7) A_F: F→ f_S，其中 f_S 是 D 的符号之和集，使得对 n 元谓词 p∈P，若(t,p)∈F 或(p,t)∈F，则 $A_F(t,p)$或 $A_F(p,t)$为 n 元符号和，对于 t∈T 出现在公式 $A_T(t)$中的自由变量恰为出现在以 t 为一端的有向弧上的自由变量；

(8) f：T→[0，1]是结合函数。对 t∈T，$f(t) = v_i$，$v_i\in$ [0，1]对应规则的置信因子；

(9) α：P→$2^{[0,1]}$是结合函数。对 p∈P, α(p)谓词 p 的真值；

(10) M_0：$P \to 2^{N \times f_S \times [0,1]}$为标识函数。对于 n 元谓词 p∈P，$M_0(p)$是 n 元符号和。$M_0$是个体集 D 在位置集 P 上的一个分布。

定义 1.4.4[117] 设 PN = (P，T，F，M_0，W)是一个含n个转移结点，m个位置结点的 Petri 网，称矩阵 $C=(c_{ij})_{n\times m}$ 是 PN 的关联矩阵，如果 $c_{ij}=\mathrm{W}(t_i,p_j)\text{-}\mathrm{W}(p_j,t_i), t_i\in T, p_j\in P$。

定义 1.4.5[117]一个m元整数列向量X称为 PN 的一个 T-不变量，如果满足 $C\times X=0$。

对于 Horn 子句模型(每个转移结点的后集最多只有一个位置结点)，Martinez[122]提出了一个基于线性代数的求 T-不变量的算法：

算法 1.1 设 C 是一个 Petri 网的 n×m 关联矩阵，I_n是 n 阶单位矩阵。

(1) 令 A=C；E=I_n(n 是 Horn 子句集中子句个数，即每一个变迁表示一个子句)；

(2) 重复 i=1 到 m(m 是子句集中原子个数，即每一个原子用一个位置表示)：

2. 1) 对矩阵 A 中的第 i 列符号相反的任意两行，在矩阵(E¦A)中进行相加运算，并将产生的新行加入(E¦A)中；

2. 2) 从(E¦A)中消除在 A 中第 i 列元素不为零的行，而后消除第 i 列。

最后得到的矩阵行向量即为关联矩阵 C 的所有 T-不变量。

命题 1.4.1[89]若基子句集 S 是不可满足的，那么表示 S 的 Petri 网有非负的 T-不变量。

命题 1.4.2[89] Horn 基子句集 S 是不可满足的当且仅当 S 的 Petri 网中有 T-不变量 X≥0，且 $X(t_g)\neq0$，其中 t_g是目标变迁。

命题 1.4.3[89] 一阶 Horn 子句集 S 是不可满足的当且仅当 S 的 Petri 网中有 T-不变量 X≥0，且 $X(t_g)\neq0$，其中 t_g是目标变迁。

从命题 1.4.1-1.4.3 可以看到：已有的方法都是以代数分析的技术计算网络的 T-不变量，根据 T-不变量判定子句集特别是 Horn 子句集的不可满足性。

1.4.2 格值命题逻辑系统 LP(X)有关的概念与命题

定义 1.4.6[159]设(L,∨,∧,O,I)是一个具有逆序对合“′”的且有泛界 O 和 I 的有界

格，若映射→：$L\times L\to L$ 满足：对任意 $x,y,z\in L$

(I_1) $x\to(y\to z)=y\to(x\to z)$;

(I_2) $x\to x=I$;

(I_3) $x\to y=y'\to x'$;

(I_4) 如果 $x\to y=y\to x=I$，则 $x=y$;

(I_5) $(x\to y)\to y=(y\to x)\to x$;

(I_6) $(x\vee y)\to z=(x\to z)\wedge(y\to z)$;

(I_7) $(x\wedge y)\to z=(x\to z)\vee(y\to z)$。

则称$(L,\vee,\wedge,',\to,O,I)$为格蕴涵代数。

定义 1.4.7[159] 若$(L,\vee,\wedge,',\to,O,I)$为格蕴涵代数，满足$x\vee y\vee((x\wedge y)\to z)=I$，则称$(L,\vee,\wedge,',\to,O,I)$为格 H 蕴涵代数。

定义 1.4.8[159] 设 X 是命题变元集合，$T=L\cup\{',\to\}$是一型，其中对于任意的$\alpha\in L$，$ar(\alpha)=0$，$ar(')=1$，$ar(\to)=2$，称 X 上的自由 T 代数为命题变元集合 X 上的格值命题逻辑系统的命题代数，记为 LP(X)。

定义 1.4.9[159] 如果映射v：$LP(X)\to L$ 是一个 T 代数同态， 则称v是 LP(X)的一个赋值。

定义 1.4.10[159]设 $F\in LP(X)$，$\alpha\in L$。如果对于 LP(X)上的任意赋值v，均有$v(F)\geq\alpha$，称 F 为α-恒真。如果$\alpha=I$，简称 F 为恒真。

定义 1.4.11[159]设 $F\in LP(X)$，$\alpha\in L$。如果对于 LP(X)上的任意赋值v，均有$v(F)\leq\alpha$，称 F 为α-恒假。如果$\alpha=O$，简称 F 为恒假。

定义 1.4.12 [159] 设 F，$G\in LP(X)$。如果对于 LP(X)上的任意赋值v，均有$v(F)=v(G)$，称 F 和 G 为等值命题，记为 $F=G$。

对于任意的 F，$G\in LP(X)$，令 $F\vee G=(F\to G)\to G$，$F\wedge G=(F'\vee G')'$。

命题 1.4.4[159]对于任意的 F，G，$H\in LP(X)$，下列结论成立：

(1) $F\to(G\to H)=G\to(F\to H)$;

(2) $F\to G=G'\to F'$;

(3) $(F\to G)\to G=(G\to F)\to F$;

(4) $F\to F=I$;

(5) $(F\vee G)\to H=(F\to H)\wedge(G\to H)$;

(6) $(F\wedge G)\to H=(F\to H)\vee(G\to H)$;

(7) $F\to(G\vee H) = (F\to G)\vee(F\to H)$;

(8) $F\to(G\wedge H) = (F\to G)\wedge(F\to H)$;

(9) $F\to O = F'$;

(10) $I\to F = F$;

(11) $O\to F = F\to I = I$;

(12) $(F\to G)\to(H\to G) = H\to(F\vee G) = (G\to F)\to(H\to F)$;

(13) $(F\to G)\to(F\to H) = (F\wedge G)\to H = (G\to F)\to(G\to H)$;

(14) $F\vee(G\wedge H)=(F\vee G)\wedge(F\vee H)$;

(15) $F\wedge(G\vee H)=(F\wedge G)\vee(F\wedge H)$。

第2章 基于Petri网的矩阵归结推理

逻辑问题的求解和证明一般采用符号逻辑的方法，在Herbrand定理基础上应用归结原理来解决。尽管已有了相当严格的理论基础，但在实际应用中由于归结是采用回溯机制实现的，因此推理的效率往往很低，这也是制约其发展的主要因素。但可以在此基础上研究改进归结的实现方法，从而以新的实现机制提高推理效率，采用Petri 网T-不变量判别法的思想可以达到此目的。

从Murata的求T-不变量的过程(算法1.1)可以看到：关联矩阵的每一行表示一个子句，如文字A对应的位置元素为1表示子句中出现文字A，-1表示出现文字¬A，0表示文字A不出现，结合命题1.4.3，可以得到结论：求T-不变量的过程其实是对子句的归结过程，而这种算法在有穷步内实现。我们借用这种思想，并结合子句集的单文字、纯文字以及分裂规则[45]，探讨基于归结的自动推理的矩阵方法。

本章主要研究：

1. 子句集的矩阵表示形式；
2. 子句集矩阵的化简策略；
3. 子句集矩阵的初等变换；
4. 几种矩阵归结策略。

2.1 子句集的矩阵表示形式

在本章中，总假定子句集中不含重言式，即文字L与¬L不在同一子句中出现。

为了用代数方法进行归结，将子句集的每个子句用一个向量表示，并作为行向量构成子句集的矩阵。

定义 2.1.1 设S是一个子句集，$\{p_1,p_2,\ldots,p_n\}$是S中的所有原子集合，$(a_{i1},a_{i2},\ldots,a_{in})(i=1,2,\ldots,m$；m是S中的子句个数) 是一个n维的行向量，其中：

$$a_{ij}=\begin{cases}1 & 若p_j在第i个子句中出现\\ -1 & 若\neg p_j\ 在第i个子句中出现\\ 0 & 其他\end{cases}\qquad (j=1,2,\ldots,n)$$

称$(a_{i1},a_{i2},\ldots,a_{in})(i=1,2,\ldots,m)$是子句集 S 中第 i 个子句的向量表示，称矩阵

$$C=\begin{pmatrix}a_{11} & a_{12} & \cdots & a_{1n}\\ a_{21} & a_{22} & \cdots & a_{2n}\\ & \cdots & \cdots & \\ a_{m1} & a_{m2} & \cdots & a_{mn}\end{pmatrix}$$

为子句集 S 的矩阵表示形式。若子句集 S 是不可(可)满足的，则称子句集矩阵是不可(可)满足的。

显然，一个子句集的矩阵不是唯一的，但它们对应的子句集是唯一的，子句集矩阵仅仅是子句集的另一种表示形式。判断子句集的不可满足性可转化为判定子句集矩阵的不可满足性。

仍然用 S 记子句集 S 的矩阵。

例 2.1.1 设 S={A, B, ¬A∨¬B∨C, ¬B∨¬C ∨D, A∨¬D, C ∨¬D , ¬C ∨¬D }，则 S 的矩阵表示为：

$$S=\begin{pmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ -1 & -1 & 1 & 0\\ 0 & -1 & -1 & 1\\ 1 & 0 & 0 & -1\\ 0 & 0 & 1 & -1\\ 0 & 0 & -1 & -1\end{pmatrix}$$

它亦为 Horn 子句集 S 的 Petri 网模型的关联矩阵[117]。

2.2　子句集矩阵的化简策略

为了简化子句集的矩阵，根据基子句集的单文字、纯文字以及分裂规则[45]，得到下列简化矩阵的策略。

策略 1(矩阵的单文字规则) 若子句集的矩阵 S 中有一行只有一个非零元

素，那么删除该行以及与该非零元素同列且相等的元素所在行得矩阵 S′。若 S′ 为空，则 S 是可满足的，否则设将 S′ 中与非零元素同列的非零元素变为 0 后的矩阵为 S″，若 S 不可满足当且仅当 S″ 不可满足。

例 2.2.1 续例 2.1.1 在矩阵 S 中第一行是单文字子句的向量，那么使用策略 1 得：

$$S_1' = \begin{pmatrix} 0 & 1 & 0 & 0 \\ -1 & -1 & 1 & 0 \\ 0 & -1 & -1 & 1 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & -1 & -1 \end{pmatrix}, \quad S_1'' = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & -1 & -1 & 1 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & -1 & -1 \end{pmatrix};$$

再对 S_1'' 第一行使用策略 1 得：

$$S_2'' = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 1 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & -1 & -1 \end{pmatrix};$$

继续上过程得：

$$S_3'' = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -1 \end{pmatrix};$$

$$S_4'' = \begin{pmatrix} 0 & 0 & 0 & 0 \end{pmatrix}。$$

S_4'' 表示空子句，不可满足，故 S 不可满足。

策略 2(矩阵的纯文字规则) 若子句集的矩阵 S 中有一列元素只有 0、1 或 0、-1，则删除非零元素所在行，设矩阵变为矩阵 S″。若 S″ 为空矩阵(没有任何元素的矩阵)，则 S 可满足，否则 S 不可满足当且仅当 S″ 不可满足。

策略 3(矩阵的分裂规则) 若子句集的矩阵 S 的行数较大时，将矩阵 S 分裂成两个子矩阵 S_1 与 S_2。具体分解办法是：若 S 中的某列元素有最少的 0 元素，选择该列。那么 S 中所选列中为 0 与 1 的元素所在行组成一个矩阵，然后将这个矩阵的所选列的元素 1 变为 0 得矩阵 S_1；而 S 中所选列中为 0 与-1 的元素所在行组成一个矩阵，然后将这个矩阵的所选列的元素-1 变为 0 得矩阵 S_2。则矩阵 S 不可满足当且仅当矩阵 S_1 与 S_2 都不可满足。

策略 4(删除规则) 若矩阵 S 有两行元素相同，删除其中一行。

若两行元素相同，则对应子句重复，所以删除该行，并不影响不可满足性。又根据基子句集的单文字、纯文字以及分裂规则，我们可以得到：

定理 2.2.1 设矩阵 S′ 是由矩阵 S 经过上述化简策略后所得矩阵，则矩阵 S 不可满足当且仅当矩阵 S′ 不可满足。

例 2.2.2 续例 2.1.1，选 S 第三列分裂为：

$$S_1=\begin{pmatrix}1&0&0&0\\0&1&0&0\\-1&-1&0&0\\1&0&0&-1\\0&0&0&-1\end{pmatrix},\quad S_2=\begin{pmatrix}1&0&0&0\\0&1&0&0\\0&-1&0&1\\1&0&0&-1\\0&0&0&-1\end{pmatrix}$$

对 S_1，S_2 第一行使用单文字规则分别可得：

$$S_3=\begin{pmatrix}0&1&0&0\\0&-1&0&0\\0&0&0&-1\end{pmatrix},\quad S_4=\begin{pmatrix}0&1&0&0\\0&-1&0&1\\0&0&0&-1\end{pmatrix}$$

对 S_3，S_4 第一行使用单文字规则分别可得：

$$S_5=\begin{pmatrix}0&0&0&0\\0&0&0&-1\end{pmatrix}\qquad S_6=\begin{pmatrix}0&0&0&1\\0&0&0&-1\end{pmatrix}$$

由定理 2.2.1 可知 S 不可满足当且仅当 S_5 与 S_6 同时不可满足，而 S_5 与 S_6 要比 S 简化得多。

2.3　子句集矩阵的初等变换

一般矩阵的初等变换包括三种：交换、倍乘与倍加。 由于子句集矩阵的特殊性，为了能经过初等变换达到归结之目的，这里给出子句集矩阵 S 的三种变换形式，仍称为初等变换。

(1)　交换：交换 S 的某两行($c_i \leftrightarrow c_j$)；

(2)　除倍：将矩阵中元数 k 变为 1，-k 变为-1，其中 k>0($k \div a_{ij}$)；

(3)　倍加(一倍)：S 中的某两行(其中只有一列元素互为相反数)相加，并将所得行(对应列变为元素 0)除倍后置于 S 的最后一行(c_i+c_j)。

定义 2.3.1 判定一个子句集的满足性与不可满足性问题，称为相容性判定。

定理 2.3.1 设矩阵 S′ 是由矩阵 S 经过上述初等变换后所得矩阵，则矩阵 S 不可满足当且仅当矩阵 S′ 不可满足。

证明：交换与除倍不会改变子句集的相容性，倍加运算是将归结式放入子句集中，根据水平渗透法，这也不会改变子句集的相容性，所以定理得证。

由定理 2.2.1 及定理 2.3.1 可得：

定理 2.3.2 设矩阵 S′ 是由矩阵 S 经过化简策略或初等变换后所得矩阵，则矩阵 S 不可满足当且仅当矩阵 S′ 不可满足。

例 2. 3. 1 设 $S=\begin{pmatrix}1 & 0\\ 0 & 1\\ -1 & -1\end{pmatrix}$，

则 $$S=\begin{pmatrix}1 & 0\\ 0 & 1\\ -1 & -1\end{pmatrix}\xrightarrow{c_3+c_1}\begin{pmatrix}1 & 0\\ 0 & 1\\ -1 & -1\\ 0 & -1\end{pmatrix}\xrightarrow{c_2+c_4}\begin{pmatrix}1 & 0\\ 0 & 1\\ -1 & -1\\ 0 & -1\\ 0 & 0\end{pmatrix}=S'$$

由于 S′ 中包含 S 中子句以及 S 的所有归结式，按照水平渗透法思想，S 不可满足当且仅当 S′ 不可满足。显然 S′ 不可满足，因为行(0，0)对应空子句，故 S 不可满足。

定理 2.3.3 矩阵 S 不可满足当且仅当矩阵 S 能经过化简策略或初等变换后矩阵 S′ 含有零向量作为行。

证明：(充分性)若子句集矩阵 S 经过化简策略或初等变换后含有零向量作为行，由于零向量对应空子句，由定理 2.3.1 知，子句集矩阵 S 是不可满足的。

(必要性)若子句集矩阵 S 不可满足，即矩阵 S 对应的子句集不可满足，故存在从 S 对应的子句集推出空子句的归结演绎，归结演绎中子句或者是子句集中子句，或者是归结式。归结过程与倍加变换对应，故归结演绎结果必在 S 的经化简或初等变换后的矩阵系列中。 当然，空子句对应的零向量也在矩阵中，所以子句集矩阵 S 经过化简策略或初等变换后含有零向量作为行。

2.4 矩阵的归结判别法

上述化简策略以及初等变换就可以对矩阵的不可满足性进行判定，但为了

得到更一般的矩阵归结算法，特别是询问式归结推理，我们引入合成运算。

定义 2.4.1 设 $a_i=(a_{i1},a_{i2},\ldots,a_{in})$, $a_j=(a_{j1},a_{j2},\ldots,a_{jn})$是两子句的向量表示形式，且其中只有一列元素互为相反数。定义 $a_i \circ a_j$ 为 a_i+a_j 中元素除倍后所得向量，称 $a_i \circ a_j$ 为向量 a_i 与 a_j 的合成。

显然有：

性质 2.4.1 $a_i \circ a_j = a_j \circ a_i$。

性质 2.4.2 $R(a_i, a_j) = a_i \circ a_j$，其中 $R(a_i, a_j)$ 为 a_i 与 a_j 的二元归结式的向量表示。

定义 2.4.2 设 S_1, S_2 是两个子句集的矩阵，定义 S_1 与 S_2 的合成(记为 $S_1 \circ S_2$)为一个子句集矩阵，它的行由 S_1 的行向量分别与 S_2 的行向量合成构成。

例 2.4.1 设 $S_1=(-1, -1)$，$S_2=\begin{pmatrix} 1 & 1 \\ 1 & -1 \\ -1 & 1 \end{pmatrix}$，则 $S_1 \circ S_2=\begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix}$。

性质 2.4.3 $S_1 \circ S_2 = S_2 \circ S_1$。

从定义 2.4.2 可以看到：S_1 与 S_2 的合成，其实是求两个子句集 S_1 与 S_2 之间的二元归结式构成的子句集矩阵。

2.4.1 基于支撑演绎的矩阵归结算法

由于一般子句集对支撑归结是完备的。在询问式推理中，如果前提子句是矛盾的，那么可以推出任意结论，这样的讨论是没有意义的，所以我们总假定前提子句集是可满足的。而一个子句集合包含前提子句与目标子句，如果一个子句集是不可满足的，那么它的目标子句集必是支撑集。这一节，结合上述讨论的化简策略以及初等变换，对一般子句集进行相容性判定。

算法 2.1(一般子句集的支撑归结+矩阵归结)

(1) 令 S_0 = 目标子句的矩阵，S = 不含目标子句的子句集矩阵；

(2) 令 i=1；

(3) 计算 $S_{i-1} \circ S_{i-1}$，$S_{i-1} \circ S$，若 $S_{i-1} \circ S_{i-1}$，$S_{i-1} \circ S$ 中存在一行为零向量，转(8)，否则(4)；

(4) 令 $S_i=\begin{pmatrix} S_{i-1} \\ S_{i-1}\circ S_{i-1} \\ S_{i-1}\circ S \end{pmatrix}$;

(5) 若 S_i 中存在一行为零向量，则转(8)，若 $S_i=S_{i-1}$，转(9)，否则(6);

(6) 应用化简策略对 S_i 化简，若化简后矩阵存在一行为零向量，则转(8)，若为空矩阵，转(9)，否则(7);

(7) 令 i=i+1，转(3);

(8) 子句集 $S_0\cup S$ 不可满足;

(9) 子句集 $S_0\cup S$ 可满足。

定理2.4.1(算法 2.1 的完备性)子句集矩阵 $\begin{pmatrix} S \\ S_0 \end{pmatrix}$ 是不可满足的当且仅当算法 2.1 终止在第(8)步，其中 S 为不含目标子句的矩阵，S_0 为目标子句的矩阵。

证明：算法 2.1 第(3)步是支撑归结，第(6)步是对子句集及其后裔化简。若子句集矩阵 $\begin{pmatrix} S \\ S_0 \end{pmatrix}$ 是不可满足的，由定理 2.2.1 以及支撑归结的完备性知，算法 2.1 终止在第(8)步。

若算法 2.1 终止在第(8)步，说明从子句集矩阵 $\begin{pmatrix} S \\ S_0 \end{pmatrix}$ 可以经过化简或初等变换或支撑归结后矩阵含有零向量作为行，由定理 2.2.3 以及支撑归结的完备性知，子句集矩阵 $\begin{pmatrix} S \\ S_0 \end{pmatrix}$ 是不可满足的。

例 2.4.1 判定 $P\wedge Q$ 是否可由($P\vee Q$，$P\vee\neg Q$，$\neg P\vee Q$)推出。

(1) $S_0=(-1,\ -1)$，$S=\begin{pmatrix} 1 & 1 \\ 1 & -1 \\ -1 & 1 \end{pmatrix}$;

(2) $S_0\circ S=\begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix}$，所以 $S_1=\begin{pmatrix} -1 & -1 \\ 0 & -1 \\ -1 & 0 \end{pmatrix}$;

(3)　$S_1 \circ S = \begin{pmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \end{pmatrix}$，所以 $S_2 = \begin{pmatrix} -1 & -1 \\ 0 & -1 \\ -1 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}$；

(4)　$S_2 \circ S_2 = \begin{pmatrix} 0 & -1 \\ 0 & 0 \end{pmatrix}$，故子句集 $S_0 \cup S$ 不可满足，即 $P \wedge Q$ 可由($P \vee Q$，$P \vee \neg Q$，$\neg P \vee Q$)推出。

若 $S_0 \cup S$ 是 Horn 子句集，由于 Horn 子句集对输入归结、单元归结都是完备的，结合一般子句集的支撑归结，故可得 Horn 子句集的矩阵归结策略：

算法 2. 2(Horn 子句集的输入归结+支撑归结+化简归结)

(1) S_0 = 目标子句的矩阵，S = 不含目标子句的子句集矩阵；

(2) 令 i=1；

(3) 计算 $S_{i-1} \circ S$，令 $S_i = S_{i-1} \circ S$，若 S_i 中存在一行为零向量，转(6)；若 $S_i = S_{i-1}$，转(7)，否则(4)；

(4) 应用化简策略对 $\begin{pmatrix} S_0 \\ S \\ S_i \end{pmatrix}$ 化简，若化简后矩阵存在一行为零向量，则转(6)，若为空矩阵，转(7)，否则(5)；

(5) 令 i=i+1，转(3)；

(6) 子句集 $S_0 \cup S$ 不可满足；

(7) 子句集 $S_0 \cup S$ 可满足。

定理 2.4.2(算法 2.2 的完备性) Horn 子句集 $S_0 \cup S$ 是不可满足的当且仅当算法 2.2 终止在第(6)步，其中 S 为不含目标子句的矩阵，S_0 为目标子句的矩阵。

证明：(必要性)算法 2.2 第(3)步是计算目标子句及其后裔，由于只能与 S 合成(即归结)，所以它是输入归结，第(4)步是对子句集及其后裔化简。若 Horn 子句集 $S_0 \cup S$ 是不可满足的，由于 Horn 子句集对输入归结以及支撑归结的完备性以及定理 2.3.3 知，算法 2.2 终止在第(6)步。

(充分性)由合成运算的性质以及输入归结、支撑归结和化简归结对 Horn 子句集的完备性，结论显然。

例 2.4.2 已知 $A \to B$，$B \to C$，$E \to F$，$F \to D$，A，E，证明：$C \wedge D$。

证明：(1) S_0= (0,0,-1,-1,0,0),$S=\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 \end{pmatrix}$；

(2) $S_1=S_0\circ S=\begin{pmatrix} 0 & -1 & 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & -1 \end{pmatrix}$；

(3) $S_2=S_1\circ S=\begin{pmatrix} -1 & 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & -1 \\ 0 & 0 & -1 & 0 & -1 & 0 \end{pmatrix}$；

(4) $S_3=S_2\circ S=\begin{pmatrix} 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & -1 \\ 0 & -1 & 0 & 0 & -1 & 0 \end{pmatrix}$；

(5) $S_4=S_3\circ S=\begin{pmatrix} 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \\ -1 & 0 & 0 & 0 & -1 & 0 \end{pmatrix}$；

(6) $S_5=S_4\circ S=\begin{pmatrix} -1 & 0 & 0 & 0 & -1 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \end{pmatrix}$；

(7) $S_6=S_5\circ S=\begin{pmatrix} -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \end{pmatrix}$；

(8) $S_7=S_6\circ S=\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$。

因为 S_7 中含零行，所以 $S_0\cup S$ 不可满足，即 $C\wedge D$ 得证。

由于单元归结与输入归结对 Horn 子句集是等价的，所以下面讨论 Horn 子句集的单元矩阵归结。

算法 2.3(Horn 子句集的单元归结+化简归结)：

(1) S_0 = 单元子句的矩阵，S = 非单元子句的子句集矩阵；

(2) 令 i=1；

(3) 计算 $S_{i-1}\circ S_{i-1}$，若 $S_{i-1}\circ S_{i-1}$ 中存在一行为零向量，转(7)，否则(4)；

(4) 计算 $S_{i-1}\circ S$，令 $S_i=\begin{pmatrix} S_{i-1} \\ S_i\circ S\text{中的单元子句集}\end{pmatrix}$，若 S_i 中存在一行为零向量，转(6)；若 $S_i=S_{i-1}$，转(8)，否则(5)；

(5) 应用化简策略对 $\begin{pmatrix} S_0 \\ S \\ S_i \end{pmatrix}$ 化简，若化简后矩阵存在一行为零向量，则转(7)，若为空矩阵，转(8)，否则(6)；

(6) 令 i=i+1，转(3)；

(7) 子句集 $S_0\cup S$ 不可满足；

(8) 子句集 $S_0\cup S$ 可满足。

定理 2.4.3(算法 2.3 的完备性)Horn 子句集矩阵 $\begin{pmatrix} S \\ S_0 \end{pmatrix}$ 是不可满足的当且仅当算法 2.3 终止在第(7)步，其中 S 为不含单元子句的子句集矩阵，S_0 为单元子句的矩阵。

根据 Horn 子句集对单元归结以及化简的完备性，容易得到结论。

例 2.4.3 续例 2.4.2 已知 $A\rightarrow B$，$B\rightarrow C$，$E\rightarrow F$，$F\rightarrow D$，A，E，证明：$C\wedge D$。

证明：(1) $S_0=\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$，$S=\begin{pmatrix} -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 & -1 & 1 \end{pmatrix}$；

(2) $S_1=\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$；

(3) $S_2=\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$；

(4) S_2。$S_2=\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$。

因为 S_2。S_2 中含零行，所以所以 $S_0 \cup S$ 不可满足，即 $C \wedge D$ 得证。

2.4.2 基于 T-不变量推理的矩阵归结算法

从 Murata 求 T-不变量的算法 1.1 过程可以看到，新追加的行为(E ¦ A)中任意两行的合成(即归结，不使用文字的化简)，E 的引入是为了记录转移结点激活的次数，即子句参与归结的次数。每一次的循环消去一个原子，由于得到了有关该原子的所有归结式，所以可以删除该原子及其非所在的子句，并删除它所在的列。由于 T-不变量的每个元素都非负，所以在消去某元素时只能进行正的线性组合，而矩阵(E ¦ A)中 A 的列表示位置即子句中的原子，同一列中元素不可能都为非负数或非正数，否则对应文字为纯文字，是可以删除的。那么按照这样的推理，若 S 是一个 Horn 子句集，且无纯文字，那么一定有 T-不变量。

由于 Horn 子句集的子句集矩阵就是其 Petri 网的关联矩阵，定义的子句的合成具有同行向量的非零线性组合同样的功能(归结)，所以可以撇开变迁的记录，直接将求 T－不变量的思想应用到矩阵归结上。又由支撑归结原理，以支撑集合即目标子句集合中的文字作为归结文字，这样可以加快归结的速度，结合化简策略以及输入归结，得到下列矩阵归结算法。

算法 2.4 (Horn 子句集的 T-不变量归结法)

(1) 令 S_0 为子句集的矩阵；

(2) 令 i=1；

(3) 令 S_{i-1}' 为矩阵 S_{i-1} 中第 i 列元素为 1 的行向量组成的矩阵；而 S_{i-1}'' 为 S_{i-1} 中第 i 列元素为-1 的行向量组成的矩阵；

(4) 计算 $S_{i-1}' \circ S_{i-1}''$，若 $S_{i-1}' \circ S_{i-1}''$ 中有零向量，则子句集 S 不可满足，否则(5)；

(5) 令 S_i 为 S_{i-1} 中去除 S_{i-1}' 与 S_{i-1}'' 中行向量，再加入 $S_{i-1}' \circ S_{i-1}''$ 中行向量的矩阵；

(6) 令 i=i+1，若 i>m(m 为 S_0 的列数)，则子句集 S 可满足，否则转(3)。

定理 2.4.4(算法 2.4 的完备性) Horn 子句集矩阵 S_0 是不可满足的当且仅当算法 2.4 终止在第(4) 步。

证明：若 Horn 子句集矩阵 S_0 是不可满足的，则由命题 1.4.2 知，存在 T-不变量 $X \geq 0$。而算法 2.4 中第(4)步对应算法 1.1 中 2.1)步，第(5)步对应算法 1.1 中 2.2)步，存在 T-不变量 $X \geq 0$，表明算法 2.4 第(4)步一定出现零向量，故算法 2.4 终止在第(4)步。

反之，若算法 2.4 终止在第(4) 步，则表明对某个 i(i≤m，m 是 S_0 中的列数)，$S_{i-1}' \circ S_{i-1}''$ 中有零向量，则对应算法 1.1，可得 A 的位置有零向量，那么 E 的位置即为一个 T-不变量 $X \geq 0$，由命题 1.4.2 知，Horn 子句集矩阵 S_0 是不可满足的。

例 2.4.4 续例 2.4.2 已知 A→B，B→C，E→F，F→D，A，E，证明：C∧D。

证明：(1) $S_0=\begin{pmatrix} -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 & -1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$；

(2) S_0' =(1,0,0,0,0,0)，S_0'' =(-1,1,0,0,0,0)；

(3) $S_0' \circ S_0''$ =(0,1,0,0,0,0)；

(4) $S_1=\begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$;

(5) S_1' =(0,1,0,0,0,0)，S_1'' =(0,-1,1,0,0,0);

(6) $S_1' \circ S_1''$ =(0,0,1,0,0,0);

(7)按上述步骤，最后可得 S_6=(0,0,0,0,0,0)。由此判定 S 是不可满足的，即 $C \wedge D$ 得证。

算法 2.5 (Horn 子句集的 T-不变量法+支撑归结)

(1) 令 T_0 为目标子句的向量，S_0 为不含目标子句的子句集的矩阵；

(2) i=1；

(3) 令 k 为 T_{i-1} 中第一个非零列向量所在的列数(用 k 指导归结文字的选择)；

(4) 令 S_i' 为 S_{i-1} 中的第 k 列中元素为 1 的行向量组成的矩阵；S_i'' 为 S_{i-1} 中的第 k 列中元素为-1 的行向量组成的矩阵；

(5) 计算 $T_{i-1} \circ S_i'$ 或 $T_{i-1} \circ S_i''$ 以及 $T_{i-1} \circ T_{i-1}$，令 $T_i=\begin{pmatrix} T_{i-1} \circ T_{i-1} \\ T_{i-1} \circ S_i' \\ T_{i-1} \circ S_i'' \end{pmatrix}$，若 T_i 中有零向量，则子句集 S 不可满足；若 $T_i=T_{i-1}$，则子句集 S 可满足；否则(6)；

(6) 设 S_i 为 S_{i-1} 中去除 S_i' 与 S_i'' 中行向量后的矩阵，令 i=i+1 转(3)。

定理 2.4.5(算法 2.5 的完备性)Horn 子句集矩阵$\begin{pmatrix} S_0 \\ T_0 \end{pmatrix}$是不可满足的当且仅当算法 2.5 终止在第(5) 步，其中 T_0 为目标子句的向量，S_0 为不含目标子句的子句集的矩阵。

证明：根据 Horn 子句集的 T-不变量法(算法 2. 4)以及支撑归结(算法 2. 1)的完备性，可得算法 2.5 的完备性。

算法 2.5 与算法 2.2、算法 2.3 比较，只考察目标子句的后裔，而循环的次数最多 m 次，故可以大大缩短归结的时间，提高归结的速度，降低归结复杂度。

例 2.4.5 续例 2.4.2 已知 $A\to B$，$B\to C$，$E\to F$，$F\to D$，A，E，证明：$C\wedge D$。

证明：(1)$T_0=(0,0,-1,-1,0,0)$，$S_0=\begin{pmatrix}1&0&0&0&0&0\\0&0&0&0&1&0\\-1&1&0&0&0&0\\0&-1&1&0&0&0\\0&0&0&1&0&-1\\0&0&0&0&-1&1\end{pmatrix}$；

(2) k=3，$S_0'=(0,-1,1,0,0,0)$；

(3) $T_0\circ S_0'=(0,-1,0,-1,0,0)$，$T_1=(0,-1,0,-1,0,0)$；

(4) $S_1=\begin{pmatrix}1&0&0&0&0&0\\0&0&0&0&1&0\\-1&1&0&0&0&0\\0&0&0&1&0&-1\\0&0&0&0&-1&1\end{pmatrix}$；

(5) k=2，$S_0'=(-1,1,0,0,0,0)$；

(6) $T_1\circ S_1'=(-1,0,0,-1,0,0)$，$T_2=(-1,0,0,-1,0,0)$；

(7) $S_2=\begin{pmatrix}1&0&0&0&0&0\\0&0&0&0&1&0\\0&0&0&1&0&-1\\0&0&0&0&-1&1\end{pmatrix}$；

(8) k=1，$S_2'=(1,0,0,0,0,0)$；

(9) $T_2\circ S_2'=(0,0,0,-1,0,0)$，$T_3=(0,0,0,-1,0,0)$；

(10) $S_3=\begin{pmatrix}0&0&0&0&1&0\\0&0&0&1&0&-1\\0&0&0&0&-1&1\end{pmatrix}$；

(11) k=4，$S_3'=(0,0,0,1,0,-1)$；

(12) $T_3\circ S_3'=(0,0,0,0,0,-1)$, $T_4=(0,0,0,0,0,-1)$；

(13) $S_4=\begin{pmatrix}0&0&0&0&1&0\\0&0&0&0&-1&1\end{pmatrix}$；

(14) k=6，$S_4'=(0,0,0,0,-1,1)$；

(15) $T_4\circ S_4'=(0,0,0,0,-1,0)$, $T_4=(0,0,0,0,-1,0)$；

(16) S_4=(0,0,0,0,1,0)；

(17) k=5，S_5' =(0,0,0,0, 1,0)；

(18) $T_5 \circ S_5'$ =(0,0,0,0,0,0)。

由此判断 S 是不可满足的，即 C∧D 得证。

如果算法 2.5 第(5)步不进行 T_{i-1} 的自归结，则得算法 2.6。

算法 2.6(Horn 子句集的 T-不变量法+输入归结)

(1) 令 T_0 为目标子句的向量，S_0 为不含目标子句的子句集的矩阵；

(2) i=1；

(3) 令 k 为 T_{i-1} 中第一个非零列向量所在的列数(用 k 指导归结文字的选择)；

(4) 令 S_i' 为 S_{i-1} 中的第 k 列中元素为 1 的行向量组成的矩阵；S_i'' 为 S_{i-1} 中的第 k 列中元素为-1 的行向量组成的矩阵；

(5) 计算 $T_{i-1} \circ S_i'$ 或 $T_{i-1} \circ S_i''$，令 $T_i=\begin{pmatrix} T_{i-1} \circ S_i' \\ T_{i-1} \circ S_i'' \end{pmatrix}$，若 T_i 中有零向量，则子句集 S 不可满足；若 $T_i=T_{i-1}$，则子句集 S 可满足；否则(6)；

(6) 设 S_i 为 S_{i-1} 中去除 S_i' 与 S_i'' 中行向量后的矩阵，令 i=i+1 转(3)。

定理 2.4.6(算法 2.6 的完备性)Horn 子句集矩阵$\begin{pmatrix} S_0 \\ T_0 \end{pmatrix}$是不可满足的当且仅当算法 2.6 终止在第(5) 步，其中 T_0 为目标子句的向量，S_0 为不含目标子句的子句集的矩阵。

证明：由 Horn 子句集的 T-不变量法(算法 2.4)以及输入归结的完备性，可得算法 2.6 的完备性。

在上文中，针对基子句集，讨论了几种矩阵归结算法，证明了算法的完备性。对于一般子句集，由 Herbrand 定理：子句集 S 是不可满足的当且仅当存在一个有限不可满足的基例集 S′。根据矩阵归结算法 2.1，从理论上可以对子句集 S 的有限基例集 S′ 进行判定，从而可以对一般子句集的相容性进行判定。

第 3 章　Petri 网的删除归结算法

在第 2 章，我们从矩阵的角度，讨论了基子句集的矩阵归结以及一般子句的矩阵归结的可能性，其中的几种归结算法应用了 Petri 网的结构分析工具 T-不变量。这一章，从 Petri 网的动态分析工具标识的转移以及 Petri 网的结构分析两方面入手，讨论二值逻辑系统中 Petri 网删除归结算法。

Petri 网的归纳分析技术是针对 Petri 网的状态复杂度而提出的，一个规模不大的系统，可能出现状态组合爆炸的危险，从而给分析带来困难。对此人们提出化简和分解的思想。将化简的思想应用到 Petri 网的逻辑推理过程中，可以从物理结构入手降低状态空间的复杂性，又可以实现推理。在推理的同时，记录下标识的转移过程，即系统的动态特征，最终根据标识的转移序列得到与 T-不变量对应的向量，从而完全由简化的物理结构入手达到代数分析技术的效果，而且推理的动态过程完全显示出来。本章从命题逻辑到一阶逻辑，从 Horn 子句到一般子句对删除归结算法进行了详细的论证。

本章主要讨论：

1. 基于命题逻辑的 Petri 网删除归结原理；
2. 基于一阶逻辑的 Petri 网删除归结原理；
3. 两种推理算法的比较与结论。

3.1　Horn 基子句集的 Petri 网删除归结原理

设 HPNM 为 Murata[17]的 Horn 基子句集的 Petri 网模型，则每个转移结点 t 表示一个规则，$^*t = \{p|p\in P, (p,t)\in F\}$表示该规则的前件，$t^* = \{p|p\in P, (t, p)\in F\}$表示该规则的后件。对于 Horn 子句集，$t^*$只有一个元素。若$^*t =\phi$(空集合)，此时结点 t 表示前提即源变迁，$t^*= \phi$表示结论即目标变迁。为了表示空子句，引入空变迁的概念。

定义 3.1.1 若变迁 t 满足$^*t=t^*=\phi$(空集合)，称 t 为空变迁，记为“|”。

为了能通过删除达到归结的目的，删除的元素包括构成 Petri 网的要素：位置结点、转移结点以及弧连接，删除的原则按照归结的原理进行，这样可以保证删除算法的完备性。

对于一个位置结点 p，记 $*p=\{t|t\in T,(t,p)\in F\}$，$p*=\{t|t\in T,(p,t)\in F\}$，并且记|G|为集合 G 包含的元素的个数。

算法 3.1(删除算法)：设 p 是一个位置结点，

(1) 设 $*p=\{t_1, t_2, \ldots, t_k\}$，$p*=\{t_1', t_2', \ldots, t_m'\}$，若 p 至少含有一个标识，则删除这个结点 p，以及所有弧连接$(t_i, p)(i=1,2,\ldots,k)$和$(p, t_j')(j=1,2,\ldots,m)$，同时合并转移结点 t_i，t_j'为结点 $t_{ij}(i=1,2,\ldots,k; j=1,2,\ldots,m)$，并将一个标识放入转移结点 t_j' $(j=1,2,\ldots,m)$的后集的每个元素中；

(2) 若 $t*\subseteq *t$ 或$*p=p*$，即对应 t 的子句为重言式时，删除 t 或 p 以及与 t 或 p 的所有弧连接；

(3) 若$|*p|=|p*|=0$，则删除位置结点 p。

注记 3.1 算法 3.1 中初始删除的位置结点是从源变迁的后集开始，因为源变迁总可以激活，那么源变迁的后集中元素总存在标识。又 Horn 子句集总存在单元子句，对应到 Petri 网，就一定存在源变迁。另外，删除位置结点的选择过程，其实就是归结文字的选择过程；

注记 3.2 在删除转移结点时，我们用下标记录了转移结点如算法 3.1(1)中的转移结点 t_i、t_j' 合并为结点 t_{ij}，目的是记录 HPNM 网的标识的动态变化过程；

注记 3.3 如果将下标如算法 3.1 (1)中 t_{ij} 的 i 与 j 看成集合{i, j}，那么 t_i 与 t_j 的下标是它的子集。删除即可以认为是删除有子下标的转移结点。

定理 3.1.1 Horn 子句集 S 不可满足当且仅当 S 的 Petri 网模型中，由算法 3.1 可得空转移结点"|"。

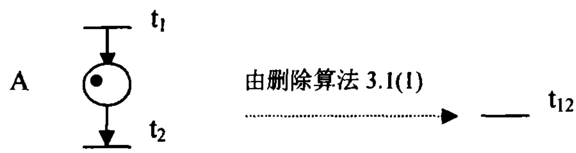
证明：若 Horn 子句集 S 不可满足的，由命题 1.4.2 知，存在 T-不变量 $X\geq 0$，使得目标变迁的激活次数大于零。既然目标变迁可以激活，由算法 3.1，目标变迁的前集中元素以及与目标变迁的弧连接都可删除，由此得空转移结点"|"。

反之，若由算法 3.1 可得空转移结点"|"。由于算法 3.1 中(1)是对应两个子句的归结，而(2)删除重言式即 Petri 网中的圈；(3)是删除对归结不起任何作用

的孤立的文字。能够得到空转移结点，说明目标变迁可达，标识可以由目标变迁溢出，说明由初始标识出发可以回到初始标识，所以可得 T-不变量$X \geq 0$，使得目标变迁的激活次数大于零，由命题 1.4.2 知：Horn 子句集 S 是不可满足的。

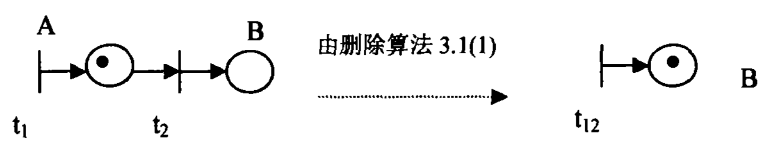
记 SHPNM 模型为将 HPNM 应用删除算法后的模型。下面我们针对几种模型对该删除算法加以说明。图中的初始标识一般置于源变迁的下集元素中，因为源变迁总可以激活。

模型一：S= {A, ¬A}，其 Petri 网模型及经删除算法后的 Petri 网模型表示如下：



标识由目标变迁 t_2 溢出，空转移 t_{12} 恰表示子句 A，¬A 的归结式-空子句。

模型二：设 S= {A，¬A∨B }，其 Petri 网模型及经删除算法后的 Petri 网模型表示如下：

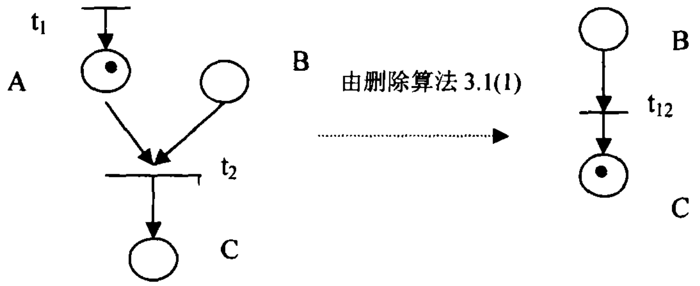


其归结式为 B，这时 t_{12} 变为了 SHPNM 网的源变迁，这是合理的。因为若 A 可达(获得标识)，那么 B 一定可达。

模型三：S= {A∨¬C，¬A∨B}，其 HPNM 模型以及 SHPNM 模型为：

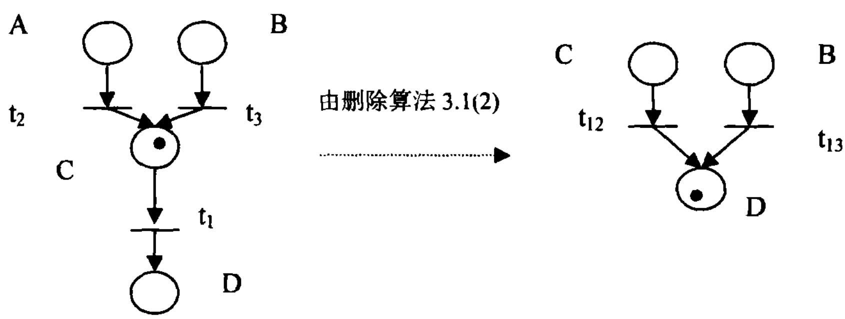
这里，为了使用删除算法 3.1，将一个标识置于位置结点 A 中。在应用删除算法进行推理时，标识的获得是根据标识的流动规则获取的。仍然得到：SHPNM 网表示的子句¬C∨B 恰为 S 的归结式。

模型四：设 S＝{A，¬A∨¬B∨C}，其 HPNM 模型以及 SHPNM 模型为：



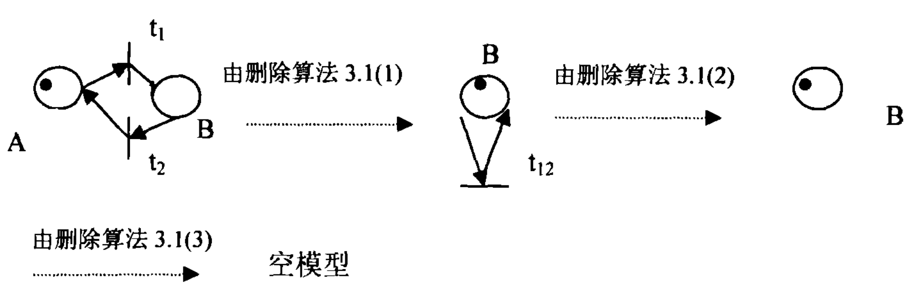
这里涉及到两个过程：一是对位置 A，根据删除算法 3.1(1)，得转移 t_{12}；二是位置 B 与转移结点 t_2 有弧连接，而 t_2 的下标属于 t_{12} 的下标集，故有一个用 t_{12} 替代 t_2 的过程。同样有 SHPNM 模型是 HPNM 的归结模型。

模型五： $S=\{A\vee\neg B, A\vee\neg C, \neg A\vee D\}$，其 HPNM 模型以及 SHPNM 模型为：



SHPNM 模型是 S 的所有归结式构成的 Petri 网。与模型三同样理由，在位置结点处放置一个标识。

模型六： $S=\{\neg A\vee B，A\vee\neg B\}$。显然其归结式为重言式，而其 HPNM 模型以及 SHPNM 模型为：



从该模型可以发现：如果 SHPNM 模型为空，它表示只能推出重言式，说明原子句集 S 是可满足的。空模型与空变迁是不同的，空变迁是指一个没有前提与结论的子句，它与空子句对应；而空模型是指没有任何子句的集合。

由上述六种模型，我们可以对 HPNM 模型应用删除算法 3.1 进行归结推理了，但是我们是否能保证算法的完备性，即当 Horn 子句集 S 是不可满足时，是否一定可以由该算法进行判定？反之，又如何判定一个子句是否可满足？以下我们对这些问题进行讨论。

定理 3.1.2 若 S 是一个 Horn 子句集，则 S 是不可满足的当且仅当 S 的 Petri 网模型 HPNM 经删除算法 3.1 作用后所得模型 SHPNM 能产生空结点“|”。

证明：若 Horn 子句集 S 是不可满足的，那么一定可以从 S 归结出空子句. 算法 3.1(1)是对应两个子句的归结或多个子句与分别与某个子句的归结；而(2)是删除重言式；(3)是删除孤立的文字，不影响归结。如果从 Horn 子句的单文字对应 HPNM 模型的源变迁的下集出发，应用删除算法 3.1，每次删除一个位置及其相应弧连接。将这样的操作作为一步，那么进行一步操作后的 SHPNM 网表示所取单文字作为归结文字的所有归结式以及 S 中不含该文字的子句的集合，即一步归结消去一个 S 中文字。若一步一步这样作下去，那么在有限步后(最多为 S 包含的原子的个数)，当不能再使用删除算法时，假若记所有 SHPNM 为一个网络模型列 SHPNM1，SHPNM2，…，SHPNMk(k 不超过 S 包含的原子的个数)。那么能从 S 归结出的所有子句必在这些模型序列表示的子句集合中，因而表示空子句的空结点也必然在这些模型序列中，故得到空结点。

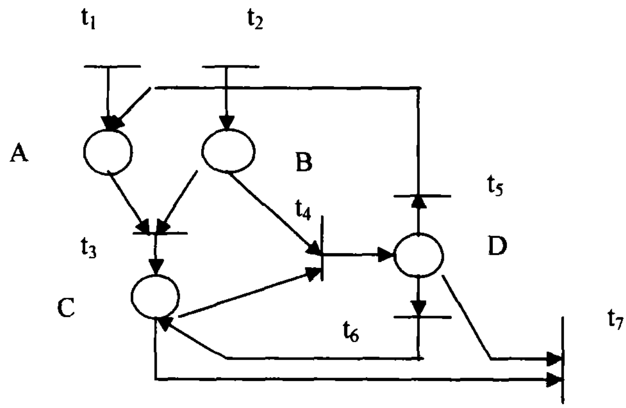
反之，若在 SHPNM 网中出现了空结点，由于 S 的归结式必在 SHPNM 表示的集合中，说明空子句可以由 S 归结推出，故 S 是不可满足的。

例 3.1.1 判定 $C\wedge D$ 是否可由命题$\{A，B，A\wedge B\rightarrow C，B\wedge C\rightarrow D，D\rightarrow A，D\rightarrow C，\neg C\vee\neg D\}$推出。

代数方法：因为 $S=\{A，B，\neg A\vee\neg B\vee C，\neg B\vee\neg C\vee D，A\vee\neg D，C\vee\neg D\}\cup G$，其中 $G=\{\neg C\vee\neg D\}$是目标子句。其 Petri 网模型及关联矩阵为：

$$C=\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & -1 & 1 & 0 \\ 0 & -1 & -1 & 1 \\ 1 & 0 & 0 & -1 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & -1 & -1 \end{pmatrix}$$

Petri 网模型为：



其中：t_1,t_2 为初始变迁, t_7 是目标变迁。

Murata 的 T-不变量的算法：

$$
\begin{array}{c}
 \\ t_1 \\ t_2 \\ t_3 \\ t_4 \\ t_5 \\ t_6 \\ t_7
\end{array}
\begin{array}{c}
\begin{array}{cccc} \quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad A & B & C & D \end{array} \\
\left(\begin{array}{ccccccc:cccc}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & -1 & 1 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & -1 & 1 \\
0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & -1 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & -1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & -1
\end{array}\right)
\end{array}
$$

$$
\Rightarrow
\begin{array}{c}
 \\ t_1+t_3 \\ t_3+t_5 \\ t_2 \\ t_4 \\ t_6 \\ t_7 \\ \
\end{array}
\begin{array}{c}
\begin{array}{ccc} \quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad B & C & D \end{array} \\
\left(\begin{array}{ccccccc:ccc}
1 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 1 & 0 \\
0 & 0 & 1 & 0 & 1 & 0 & 0 & -1 & 1 & -1 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & -1 & 1 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & -1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & -1
\end{array}\right)
\end{array}
$$

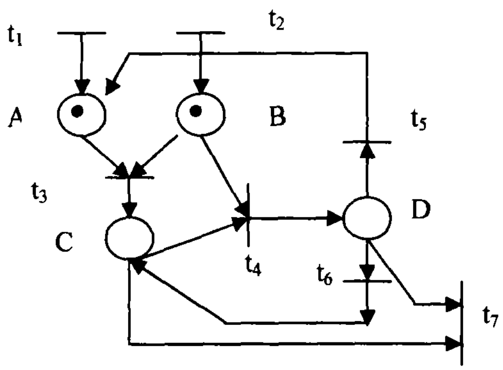
$$\Rightarrow \begin{array}{ll} & \begin{array}{cccccccccc} &&&&&&&&C&D \end{array} \\ \begin{array}{l} t_1+t_2+t_3 \\ t_2+t_3+t_5 \\ t_2+t_4 \\ t_6 \\ t_7 \end{array} & \begin{pmatrix} 1&1&1&0&0&0&0&\vdots&1&0 \\ 0&1&1&0&1&0&0&\vdots&1&-1 \\ 0&1&0&1&0&0&0&\vdots&-1&1 \\ 0&0&0&0&0&1&0&\vdots&1&-1 \\ 0&0&0&0&0&0&1&\vdots&-1&-1 \end{pmatrix} \end{array}$$

$$\Rightarrow \begin{array}{ll} & \begin{array}{ccccccccc} &&&&&&&&D \end{array} \\ \begin{array}{l} t_1+2t_2+t_3+t_4 \\ 2t_2+t_3+t_4+t_5 \\ t_2+t_4+t_6 \\ t_1+t_2+t_3+t_7 \\ t_2+t_3+t_5+t_7 \\ t_6+t_7 \end{array} & \begin{pmatrix} 1&2&1&1&0&0&0&\vdots&1 \\ 0&2&1&1&1&0&0&\vdots&0 \\ 0&1&0&1&0&1&0&\vdots&0 \\ 1&1&1&0&0&0&1&\vdots&-1 \\ 0&1&1&0&1&0&1&\vdots&-2 \\ 0&0&0&0&0&1&1&\vdots&-2 \end{pmatrix} \end{array}$$

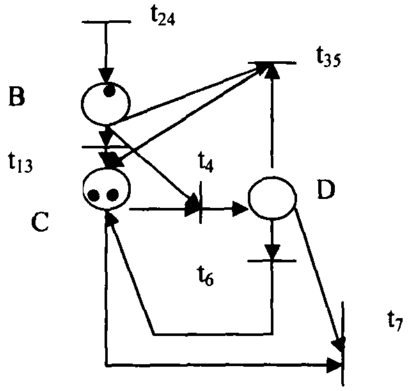
$$\Rightarrow \begin{array}{l} 2t_2+t_3+t_4+t_5 \\ t_2+t_4+t_6 \\ 2t_1+3t_2+2t_3+t_4+t_7 \\ 2t_1+4t_2+2t_3+2t_4+t_6+t_7 \\ 2t_1+5t_2+3t_3+2t_4+t_5+t_7 \end{array} \quad \begin{pmatrix} 0&2&1&1&1&0&0 \\ 1&0&1&0&1&0&0 \\ 2&3&2&1&0&0&1 \\ 2&4&3&2&0&1&1 \\ 2&5&3&2&1&0&1 \end{pmatrix}$$

$T_1=(0,2,1,1,1,0,0)$，$T_2=(1,0,1,0,1,0,0)$，$T_3=(2,3,2,1,0,0,1)$，$T_4=(2,4,3,2,1,0,1)$，$T_5=(2,5,3,2,1,0,1)$即为 T-不变量，由于 T_3，T_4，T_5 的支撑集合中都包含目标变迁 t_7，故由命题 1.4.2 知：Horn 子句集 S 是不可满足的。

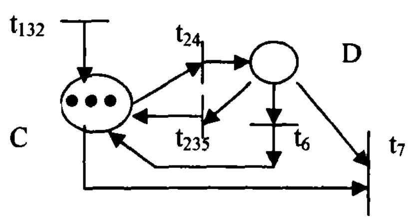
结构方法：HPNM 模型以及 SHPNM 模型系列如下：



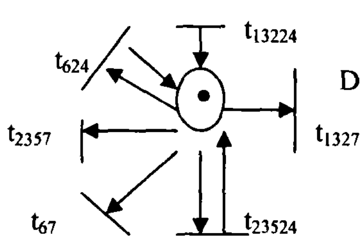
⇓ 删除位置 A 及弧连接



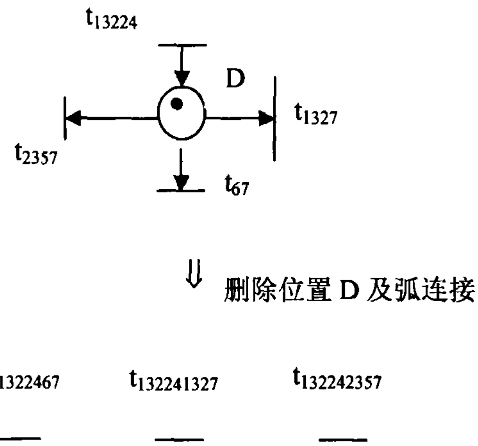
⇓ 删除位置 B 及弧连接



⇓ 删除位置 C 及弧连接



⇓ 删除重言式



最后得三个空结点 $t_{1322467}$，$t_{132241327}$ 与 $t_{132242357}$，从而可得：Horn 子句集 S∪G 是不可满足的。

分析、比较与讨论：

a) 在三个空结点的下标序列中，转移结点出现的次数构成的向量为 X_1=(1，2，1，1，0，1，1)，X_2=(2，3，2，1，0，0，1)，X_3=(1，3，2，1，1，0，1)，只有 X_2 为 HPNM 模型的 T-不变量即为代数方法中的 T_3。为什么 X_1 与 X_3 不是呢？从两种方法比较可以看到：前三步(即去除文字或结点 A，B 与 C)中结点激活情况是一致的，只是表现形式不同而已，只有在最后几步出现了差异。显然 T_1 与 T_2 对应的是第三步中的两个重言式，由于对归结没有影响，故删除了。在代数方法的第二步，t_2+t_4 对应子句 $\neg C \vee D$，t_6 对应子句 $C \vee \neg D$，它们的线性组合对应空子句，这是不符合归结原理的，它对应的 T-不变量 T_2 不能作为判定子句不相容的依据，同理 T_1 也不能。而我们的方法是合理的，将它们的归结式即重言式删除。X_1 与 X_3 是与 T_4，T_5 对应的，由于在代数方法中进行了非零线性组合，从而出现了 X_1，X_3 与 T_4，T_5 的不同，但是它们作用是相同的，即都可以用来判定子句集的相容性问题；

b) 由于源结点对应单元子句，从整个删除过程看，每一步的删除文字是归结文字，而 Horn 子句集对单元归结是完备的，故如果 Horn 子句集是不相容的，那么一定可以得到空结点即空子句。我们的删除策略可以任意文字作为删除文字，文字的选择对归结的结果没有任何影响；

c) 显然，三个空结点的下标序列给出了标识的动态变化过程，即推理的变化过程，增加了推理的可视化。

令 X 表示空结点的下标序列中变迁出现的次数构成的向量集，T 表示 HPTN 模型的 T-不变量集，根据上述分析，下列定理成立。

定理 3.1.3 若存在 $X_i \in X$，则一定存在 $T_j \in T$，使得 $0 \le X_i \le T_j$。

证明：X 中向量没有记录重言式的删除过程，如果没有重言式，那么 X 中向量的分量记录的是从初始标识回到初始标识，变迁激活次数，所以此时它就是一个 T-不变量，否则它就小于对应的 T-不变量，所以结论成立。

定理 3.1.4 若存在 $X_i \in X$，$T_j \in T$，满足 $X_i \le T_j$，则 T_j 的支撑集合中包含目标变迁。

证明：若 $X_i \in X$，$T_j \in T$，且 $X_i \le T_j$，则 $1 \le X_i(t_g) \le T_j(t_g)$，$t_g$ 是目标变迁，故 T_j 的支撑集合中包含目标变迁。

定理 3.1.5 若 S 是可满足的 Horn 子句集，G 为目标子句，则 $S \cup G$ 是不可满足的充要条件是 $X \ne \phi$。

证明：充分性：若 $X \ne \phi$，说明存在 $X_i \in X$，使得 $X_i(t_g) \ne 0$，其中 t_g 是目标变迁。由定理 3.1.3 以及定理 3.1.4 知，存在 $T_j \in T$，使得 $T_j(t_g) \ne 0$。由命题 1.4.2 知 $S \cup G$ 不可满足。

必要性：若 $S \cup G$ 不可满足，则必然存在从 S 推出 G 的归结演绎，该演绎序列对应一个变迁序列，序列中变迁出现的次数构成的向量恰是 X 中的向量，所以 $X \ne \phi$。

从上述分析可以得到以下结论：

结论 3.1 删除算法与 T-不变量法有相同的判断能力，但是删除算法更优，它避免了对判定没有作用的 T-不变量的求解，这样提高了效率；

结论 3.2 转移结点的下标动态地显示了推理过程，弥补了 T-不变量法的不足；

结论 3.3 只要子句的文字是有限的，该方法就可以在有穷步内对子句集的相容性进行判定。删除算法与水平渗透法比较，大大加快了归结的进程。

3.2 一般基子句集的 Petri 网删除归结原理

在上文中，已经给出 Horn 基子句集的 Petri 网模型的归结算法。这一节，我们将删除算法推广到一般的基子句集。由于一般基子句的 Petri 网模型有多种形式，文献[133]通过引入逻辑冲突位置及逻辑冲突变迁来完成一般子句向 Petri 网模型的转变，目的是将一般子句集转化为 Horn 子句集，从而能应用 T-不变量法对子句的相容性进行判定；我们的归结方法不是从代数分析入手，故无需这样的转换。

3.2.1 一般子句的 Petri 网模型

这一节，首先讨论子句的另一种表示形式，再给出一般子句的 Petri 网模型。

定义 3.2.1 称公式 $A \rightarrow B$ 为合成规则，若 A 或 B 是文字的合取式或析取式。

定理 3.2.1 任何一个子句，等价于一个或多个合成规则的合取。

证明：记 l 为子句中文字个数，下面对 l 用数学归纳法。

(1) 当 $l=1$ 时，结论显然成立；

当 $l=2$ 时，此时子句的形式有三种：

a) 含两个正文字，显然为一个合成规则；

b) 含一个正文字，一个负文字，如 $\neg A_1 \vee A_2$，由于 $\neg A_1 \vee A_2 = A_1 \rightarrow A_2$ 为一个合成规则；

c) 含两个负文字，如 $\neg A_1 \vee \neg A_2$，由于 $\neg A_1 \vee \neg A_2 = \neg (A_1 \wedge A_2) = A_1 \wedge A_2 \rightarrow O$，O 为恒假子句，也为一个合成规则。

(2) 假设 $l<k$ 时，结论成立。

(3) $l=k$ 时

设子句为 G，且 $G = G_1 \vee G_2 \vee \ldots \vee G_{k-1} \vee G_k$，其中 $G_i(i=1,2,\ldots,k)$ 为文字。由假设知，$G=(A \rightarrow B) \vee G_k$，A 或 B 是文字的合取式或析取式。

1)若 G_k 为正文字

$G=(A \rightarrow B) \vee G_k = \neg A \vee B \vee G_k = A \rightarrow (B \vee G_k)$

当 B 为析取式时，G 为合成规则；当 B 为合取式时，不妨设 $B=B_1 \wedge B_2 \wedge \ldots \wedge B_{k1}$，则

$G=(A \rightarrow B) \vee G_k$

$=A\rightarrow(B\vee G_k)$

$=A\rightarrow((B_1\vee G_k)\ \wedge(B_2\vee G_k)\wedge\ldots\wedge(B_{k1}\vee G_k))$

$=(A\rightarrow(B_1\vee G_k)\ )\wedge(A\ \rightarrow(B_2\vee G_k))\wedge\ldots\wedge(A\rightarrow(B_{k1}\vee G_k))$

此时，公式 G 等价于 k_1 个合成规则的合取，故结论成立。

2)若 G_k 为负文字，

$G=(A\rightarrow B)\ \vee G_k=\neg A\vee B\vee\neg(\neg G_k)=(A\wedge\neg G_k)\rightarrow B$

若 A 为合取式，显然为合成规则；若 A 为析取式，由于

$G=(A\wedge\neg G_k)\rightarrow B=(A\rightarrow B)\ \wedge(\neg G_k\rightarrow B)$

此时，公式 G 等价于两个合成规则的合取，故结论成立。

由归纳假设知，结论成立。

由于每一个子句对应一个或多个合成规则的合取，故讨论子句的 Petri 网模型，可转化为对合成规则 Petri 网模型的讨论。一般地，可以将合成规则分为以下四种类型(不妨以含一个“∧”，一个“∨”，和一个“→”的公式为例讨论)：

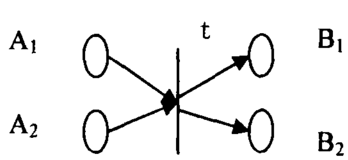
(1) $A_1\wedge A_2\rightarrow B_1\vee B_2$；

(2) $A_1\wedge A_2\rightarrow B_1\wedge B_2$；

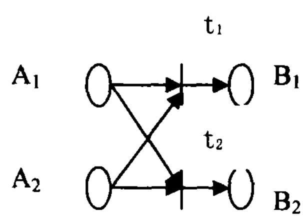
(3) $A_1\vee A_2\rightarrow B_1\vee B_2$；

(4) $A_1\vee A_2\rightarrow B_1\wedge B_2$。

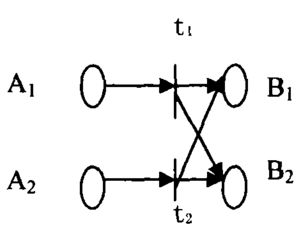
其 Petri 网模型分别为：



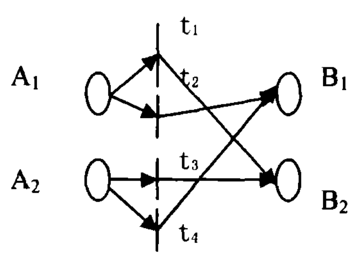
公式(1)的 Petri 网模型



公式(2)的 Petri 网模型



公式(3)的 Petri 网模型



公式(4)的 Petri 网模型

在子句集中，一般子句与 Horn 子句本质的区别是一般子句含有两个或两个以上的正文字，对于只含有最多一个正文字的子句，用 Horn 子句的 Petri 网模型表示，而对于含两个或两个以上的正文字的子句，其 Petri 网模型有不同的方法，文献[98]是用 t，f 表示连接权，从而表示相应的正文字与负文字，我们不改变逻辑推理的 Petri 网的连接权 1，而引入一种新的位置结点 p：$^*p=\phi$（空集)。在经典 Petri 网的模型中，没有这种类型的结点，它并不表示单元子句，仅仅表示一个文字，它的引入对以后的删除归结来讲，不受影响。

3.2.2　一般基子句的删除归结算法

由于一般子句的 Petri 网模型不一定有源变迁，标识不是一定可以获得，只能事先给定初始标识，故不能象 Horn 子句集的 Petri 模型那样，由标识指导删除算法中位置结点的选择。我们不考虑标识，直接由 Petri 网的结构得一般基子句集的删除归结算法。另外，一般子句的 Petri 网模型与 HPTN 模型相比，区别仅仅是对转移结点有$|t^*|>1$ 这种情形发生，但删除归结仍然选择删除文字即归结文字为基础，故一般基子句的删除归结算法与 Horn 基子句的删除归结算法类似，仅仅根据一般子句的特点作了适当修改。

算法 3.2：设 p 是一个位置结点，

(1)　若$|^*p|=k$ $(k\geq 1)$, $|p^*|=m\geq 1$, 设 $^*p=\{t_1, t_2, \ldots, t_k\}$, $p^*=\{t_1', t_2', \ldots, t_m'\}$，则删除这个结点 p，以及弧连接($t_i$，p)(i=1,2, …，k)和(p, t_j')(j=1,2, …，m)，同时合并转移结点 t_i ，t_j' 为结点 t_{ij}(i=1,2, …，k; j=1,2, …，m)；

(2)　若 $t^*\cap {}^*t\neq\Phi$ 或$^*p=p^*$，即对应 t 的子句为重言式时，删除 t 或 p 以及与 t 或 p 的所有弧连接；

(3)　若$|^*p|=0$, $|p^*|\neq 0$，则删除 p^*中每个结点及其所有连接(纯文字规则)；若$|^*p|\neq 0$, $|p^*|=0$，则删除$^*p$ 中每个结点及其所有连接(纯文字规则)；$|^*p|=|p^*|=0$，则删除位置结点 p。

定理 3.2.2 若一般子句 S 含有有限个文字，则 S 是不可满足的当且仅当 S 的 Petri 网模型经删除归结算法 3.2 后所得模型存在空结点。

证明：若 S 是不可满足的，则一定存在从 S 出发演绎出空子句的归结演绎

序列，由于算法3.2中(1)的每次转移结点的合并对应一次二元归结，(2)是删除重言式，(3)为纯文字规则。由于空子句对应空结点，所以按照演绎序列删除转移结点及其弧连接，一定得到空结点。

反之，若S的Petri网模型经删除归结算法3.2后所得模型存在空结点。说明存在从S出发演绎出空子句的归结演绎序列，故S是不可满足的。

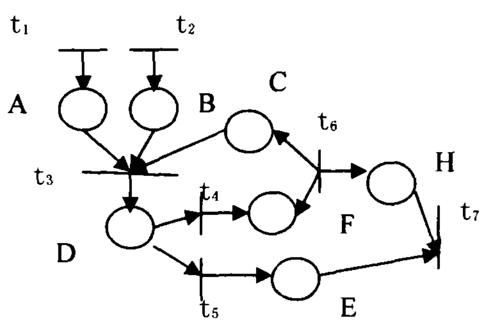
若X表示空结点的下标序列中转移出现的次数构成的向量集，那么有下列定理成立。

定理3.2.3 设S是一般基子句，则S不可满足的充要条件是$X \neq \phi$。

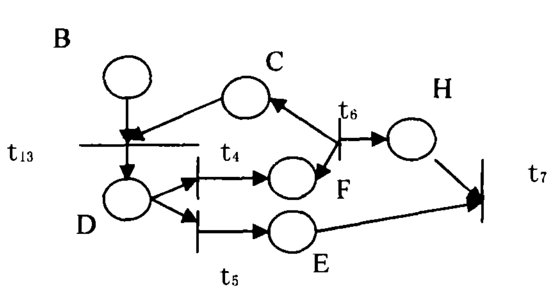
由定理2.2.1很容易证明该结论，这里略。

显然转移结点的下标序列亦记录了推理的过程。

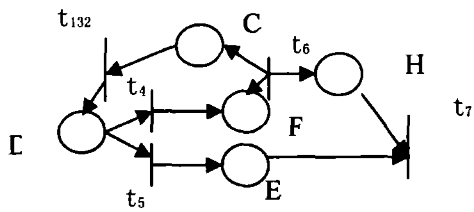
例3.2.2 设S={A，B，A∧B→┐C∨D，D→E∧F，┐C→F∨ H}，G=E∧H，S∪G的Petri网模型及删除归结算法为：



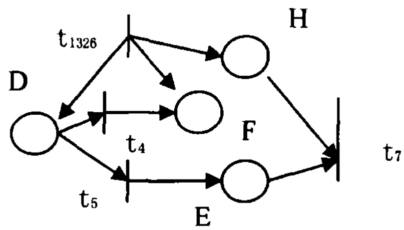
⇓ 删除位置A及弧连接



⇓ 删除位置B及弧连接

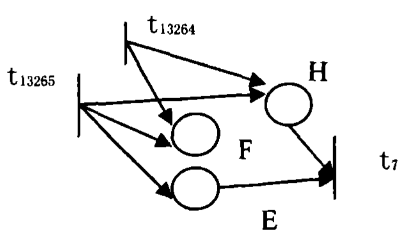


⇓ 删除位置 C 及弧连接



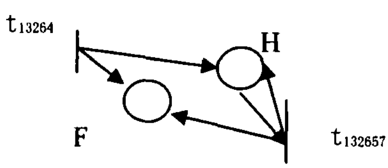
删除位置 C 及弧连接

⇓

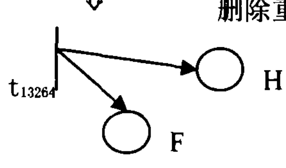


删除位置 D 及弧连接

⇓



⇓ 删除重言式



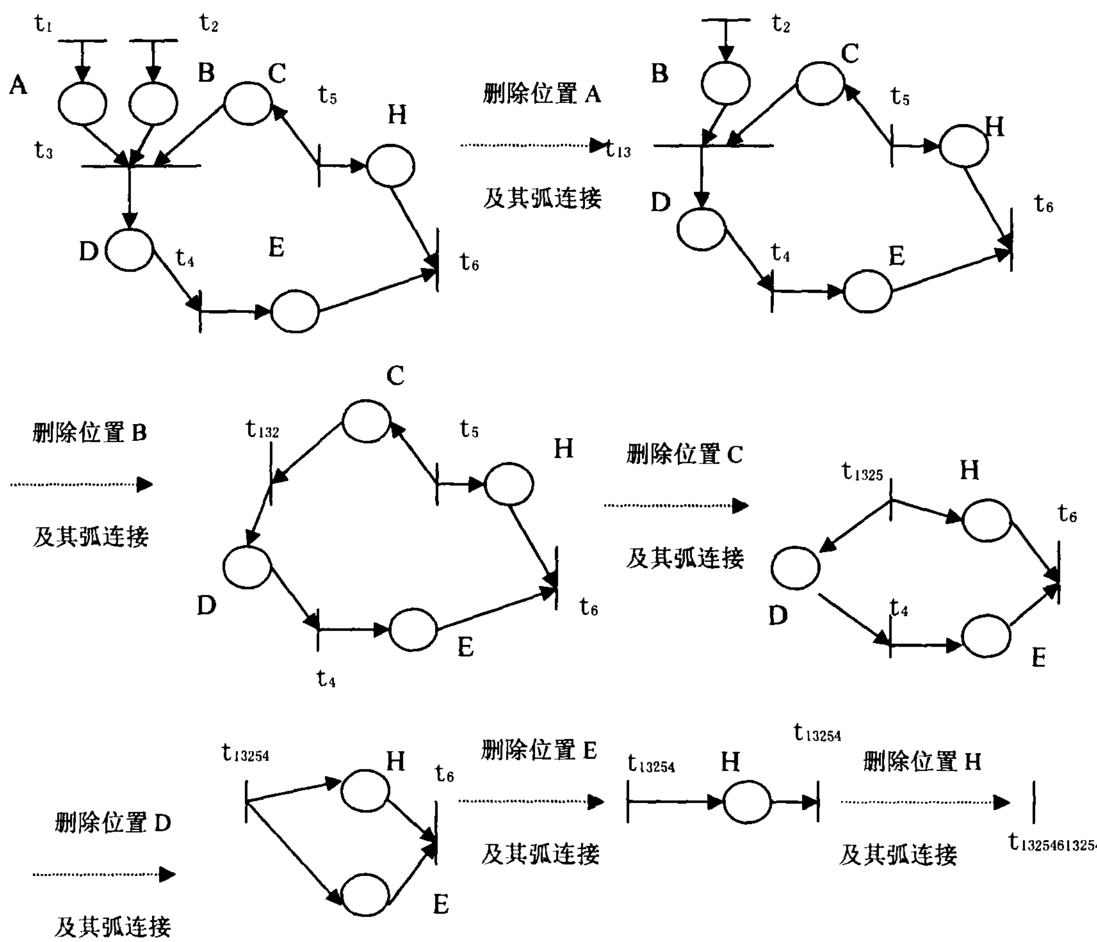
⇓ 纯文字规则

空模型

由于 X=ϕ，所以 S∪G 是可满足的。

若直接按 F 是纯文字原则，可以去除 t_4，t_6 及其弧连接，这样 C 又变为纯文字，删除 t_3 及其连接，…，依此这样下去，同样得到空模型。

例 3.2.3 若 S={A，B，A∧B→┐C∨D，D→E，┐C→ H}，G=E∧H，S∪G 的 Petri 网模型及删除归结算法为：



此时 X={(2，2，2，2，2，1)}≠ϕ，S∪G 不可满足的。

3.3 一阶 Horn 子句集的删除归结原理

对于一般子句集，由 Herbrand 定理：子句集 S 是不可满足的当且仅当存在一个有限不可满足的 S 的基例集 S′。根据矩阵归结算法 2.1，从理论上可以对子句集 S 的有限基例集 S′ 的相容性进行判定，从而可以对一般子句集的相容性进行判定。但是在实际操作中存在许多局限性，Herbrand 定理只给出了有限基例集 S′ 的存在性，并没有给出具体的基例集，如果对所有的基例集一个一个进行判定，那是一个十分复杂的过程，有时几乎是不可能完成的，所以必须探寻其它的判定方法。能不能不从基例集出发，而直接对一阶逻辑中的子句集进行判定，是一个值得探讨的问题。一阶逻辑中子句集 S 中含有谓词、变元与常元，其二元归结中归结文字是能经一般化合一算子转化为两个互补的文字。由于 Petri 网可以表示一阶逻辑中的子句及子句集，所以表示是没有问题的。已有的分析工具主要采用 T-不变量分析法，这一节，我们欲从结构与标识出发，给出基于一阶逻辑的 Petri 网删除归结算法。

定义 3.3.1 在一阶逻辑中，最多含一个正文字的子句称为 Horn 子句。由 Horn 子句构成的集合称为 Horn 子句集。

由定义 1.4.2，可以将任何一个一阶逻辑中的子句集表达成 Petri 网模型。下面给出一阶 Horn 子句集的 Petri 网删除归结算法。

算法 3.3：设 p 是一个位置结点，

(1) 设 $^*p=\{t_1, t_2, \ldots, t_k\}$，$p^*=\{t_1', t_2', \ldots, t_m'\}$，若 p 至少含有一个标识，且 $W(t_i,p)$ $(i=1,2,\ldots,k)$与 $W(p,t_j')$ $(j=1,2,\ldots,m)$中符号串可一般化合一θ，则删除这个结点 p，以及弧连接$(t_i, p)(i=1,2,\ldots,k)$和$(p, t_j')(j=1,2,\ldots,m)$，同时合并转移结点 t_i，t_j' 为结点 $t_{ij}(i=1,2,\ldots,k; j=1,2,\ldots,m)$，将一个标识放入 t_j' 的后集中的每个位置结点中，并将合一算法θ作用到与新的位置结点连接的权中；

(2) 若 $t^*\subseteq {}^*t$ 或$^*p=p^*$，且 $W(t,p)$与 $W(p,t)$可合一，则删除 t 或 p 以及与 t 或 p 的所有弧连接；

(3) 若$|^*p|=|p^*|=0$，则删除位置结点 p。

记 FHPNM 为一阶逻辑中 Horn 子句集的 Petri 网模型，FHPNM′ 为经过算法 3.3 一次作用后的 Petri 网模型，则有下列结论成立。

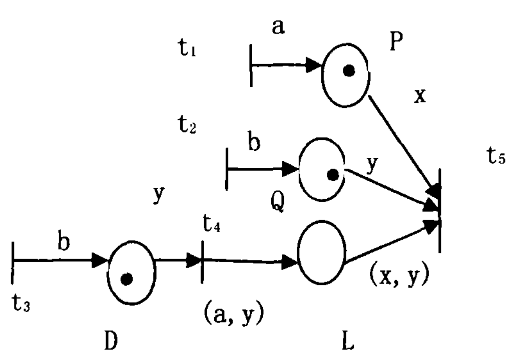
定理 3.3.1 若 FHPNM′ 所表示子句集合是不可满足的，则 FHPNM 所表示子句集合是不可满足的。

证明：算法 3.3 中(1)是进行二元归结，(2)是删除重言式，(3)是删除纯文字。(2)与(3)步不会改变子句集的不可满足性，对于(1)步进行的二元归结，由于归结式是原亲本子句的逻辑推论。若对任一解释 I，存在归结式是不可满足的，那么亲本子句中至少有一个在解释下是不可满足的，若不是归结式，则该子句也在 FHPNM 所表示子句集合中，故若 FHPNM′ 所表示子句集合是不可满足的，则 FHPNM 所表示子句集合是不可满足的。

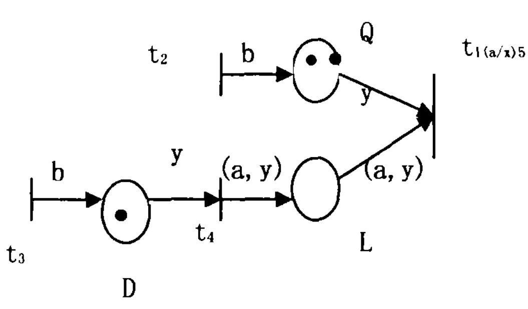
推论 3.3.1(算法 3.3 的可靠性) 在一阶逻辑中，若 Horn 子句集 S 的 Petri 网模型经算法 3.3 作用后的模型中存在空转移结点，则 S 是不可满足的。

证明：由于空转移结点对应的子句为空子句，而空子句是不可满足的，所以由定理 3.3.1 知：S 是不可满足的。

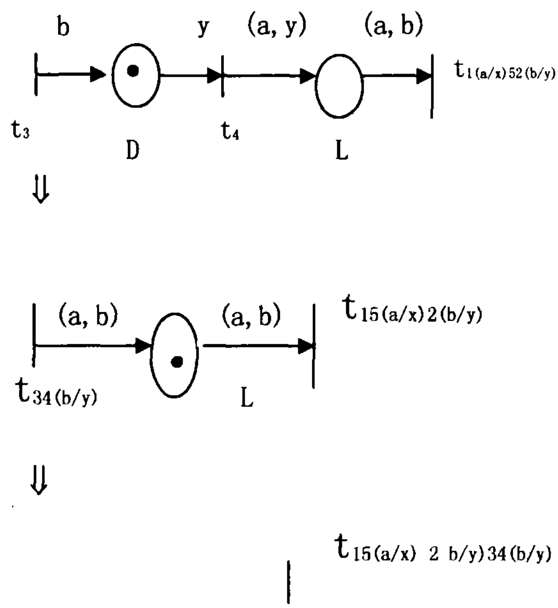
例 3.3.1 设 S = {P(a), Q(b), D(b), ¬ D(y)∨L(a,y), ¬ P(x)∨¬ Q(y)∨¬ L(x,y) }，S 的 Petri 网模型为：



⇓



⇓



由于得到空转移结点，所以 S 是不可满足的。同样可得技术分析中的 T-不变量 X=(1,1,1,1,1)。

3.4　两种推理算法的比较与结论

1. T-不变量法

(1) 算法方面：T-不变量法是对 Petri 网的关联矩阵的行向量进行非负线性组合而得到的，它是从代数的角度对 Petri 网的结构进行分析，其思想是归结原理(每次非负线性组合对应一次归结)。 但这种算法中存在缺点，即对于形如：$\neg p \vee q$ 与 $p \vee \neg q$ 的子句，它们在关联矩阵中表示为(-1，1)与(1，-1)，其非负线性组合为(0，0)，对应为空子句，这与归结原理是不相符合的，由此产生了对归结推理的判定无用的 T-不变量法，从而增加了计算的复杂度。

(2) 结构方面： T-不变量法是针对 Horn 子句集，所以对应的 Petri 网模型要求每个位置结点表示一个正文字，且每个转移结点的后集最多含一个正文字，所以对模型的结构具有一定的限制，从而也限定了其使用范围，即只适用 Horn 型子句集合推理。

从上述讨论得出 T-不变量法的优缺点如下：

(i) 算法比较简单，但有不恰当之处，即有的 T-不变量对推理判断不起作

用；

(ii) 只适用于 Horn 型子句集，故使用范围受限；

(iii) 无法显示标识的流动过程，故推理过程不直观。

2. 删除推理算法

删除推理算法是根据归结特别是集团归结的思想设计的，这点与 T-不变量法的思想是一致的。在具体处理上，该算法直接从 Petri 网的结构而不是从关联矩阵出发，根据归结的思想通过删除，简化 Petri 网的结构。对于复杂的状态空间(维数较大，位置结点个数称为状态空间的维数)，应用该算法，可以大大简化状态空间的维数，而对于 Horn 子句集，该方法亦可以得到 T-不变量，且避免了 T-不变量法的缺点。另一方面，该算法可以处理一般子句集合的推理，由于直接从结构出发进行推理，故 Petri 网模型的表示与 T-不变量法不一样，它可以用更一般的方法表示子句集合了，使用范围更广泛。而且算法的推理过程可以用转移结点的下标动态显示出来，这样推理过程更直观。

从上述分析，我们可以得到删除推理算法的优缺点如下：

(i) 适合于一般子句集的推理，故具有普适性；

(ii) 推理过程可由空转移结点的下标显示出来，所以推理较直观；

(iii) 推理速度的高效性；(从集团归结的角度来看)

(iv) 由于在不断的删除位置结点和转移结点，所以结构越来越简单；

(v) 易于将其它归结策略应用到该删除算法中，如纯文字规则、重言式规则等，故融合性较好。

第 4 章　算子命题逻辑系统及其 Petri 网推理算法

这一章，为讨论基于形如：x_1 is $P_1(\lambda_1)$，x_2 is $P_2(\lambda_2)$，…，x_m is $P_m(\lambda_m)$，then y is q (λ^*)的规则的命题推理，其中 λ^*及 λ_i 可以认为表示命题 q 及 P_i 为真的程度。我们定义一种只带命题文字以及命题算子的命题逻辑(称为算子命题逻辑 OPL(Operator Proposition Logic))，命题逻辑的这种表示形式是合理的，如“张三是高(0.8)”这个命题可以用一个子句 0.8P 表示，P 表示命题“张三是高”，而 0.8 表示对命题“张三是高的”的信任程度。又 Petri 网可以用于表示命题逻辑、一阶逻辑与模糊逻辑，根据 Horn 子句集的 Petri 网模型的 T-不变量可以实现询问式推理。所以，只要能将算子命题逻辑用 Petri 网表示出来，就可以利用 Petri 网的分析技术，实现算子命题逻辑系统的归结推理。本章主要研究：

1．算子命题逻辑系统；
2．算子命题逻辑的 λ-归结；
3．算子命题逻辑的 Petri 网模型；
4．算子命题逻辑的 Petri 网推理算法。

4.1　算子命题逻辑系统 OPL

定义 4.1.1 设＊，⊕及′是[0，1]上的两个二元运算以及一个一元运算，任取 a，b∈[0，1]，定义：

$$a * b = \min\{a,b\}$$

$$a \oplus b = \max\{a,b\}$$

$$a' = 1-a$$

则([0，1]，＊，⊕，′，0，1)是一个完备有余格。

一、系统 OPL 的语言

1．符号

系统 OPL 的符号包括下列五类：

i)　命题变元集：$X=\{p,q,r,\cdots\}$；

ii) 个体常元集：L={0，1}；

iii) 算子集：[0，1]；

iv) 逻辑连接词：$\vee$，$\wedge$，～，→；

v) 技术性符号：(，)。

2．算子命题公式的构成

定义 4.1.2 设p是原子命题，$\lambda\in[0, 1]$，称λp为命题原子，称λp与～(λp)为命题文字，称λ为命题算子。

命题公式是如下递归定义的符号串：

i) 命题原子是命题公式；

ii) 若G是命题公式，则～G是命题公式；

iii) 若G与H是命题公式，则$G\wedge H$，$G\vee H$，$G\rightarrow H$是命题公式；

iv) 所有的命题公式都是有限次使用i)- iii)生成的符号串。

记全体命题公式构成的集合为PL。

二、系统OPL的语义

根据原子命题的二值赋值，命题公式的赋值v由如下规定确定：

i) $v(\lambda p)=\begin{cases}\lambda & p\text{被指定为}1\\ 1-\lambda & p\text{被指定为}0\end{cases}$，$v(\sim(\lambda p))=\begin{cases}1-\lambda & p\text{被指定为}1\\ \lambda & p\text{被指定为}0\end{cases}$；

ii) $v(\sim G)=(v(G))'$；

iii) $v(G\wedge H)=v(G) * v(H)$；

iv) $v(G\vee H)=v(G)\oplus v(H)$；

v) $v(G\rightarrow H)=v(\sim G\vee H)$。

定义 4.1.3 命题公式G与H称为等价，如果对任意赋值v，都有$v(G)=v(H)$，记为$G=H$。

定理 4.1.1 设G，H，E均为命题公式集PL中的命题公式，p为原子命题，则下列结论成立：

(1) $\sim(\lambda p)=(1-\lambda)p$；

(2) $\sim(G\wedge H)=\sim G\vee\sim H$，$\sim(G\vee H)=\sim G\wedge\sim H$；

(3) $G\rightarrow H=\sim G\vee H$；

(4) $G\wedge(H\vee E)=(G\wedge H)\vee(G\wedge E)$，$G\vee(H\wedge E)=(G\vee H)\wedge(G\vee E)$；

(5) $\sim(\sim G)=G$。

证明：(1) 对任意赋值v，若原子命题 p 指定为 1，则$v(\sim(\lambda p))=1-\lambda$；若原子命题 p 指定为 0，则$v(\sim(\lambda p))=\lambda$，所以$\sim(\lambda p)=(1-\lambda)p$。

(2) 对任意赋值v，

$$
\begin{aligned}
& v(\sim G\vee\sim H) \\
&= v(\sim G)\oplus v(\sim H) \\
&= (v(G))'\oplus(v(H))' \\
&= (v(G) * v(H))' \\
&= (v(G\wedge H))' \\
&= v(\sim(G\wedge H))
\end{aligned}
$$

所以$\sim(G\wedge H)=\sim G\vee\sim H$，同理可证$\sim(G\vee H)=\sim G\wedge\sim H$成立。

(3) 由赋值的定义直接可证。

(4) 对任意赋值v，

$$
\begin{aligned}
& v(G\wedge(H\vee E)) \\
&= v(G) * v(H\vee E) \\
&= v(G) * (v(H)\oplus v(E)) \\
&= (v(G) * v(H))\oplus(v(G) * v(E)) \\
&= v(G\wedge H)\oplus v(G\wedge E) \\
&= v((G\wedge H)\vee(G\wedge E))
\end{aligned}
$$

所以$G\wedge(H\vee E)=(G\wedge H)\vee(G\wedge E)$，同理可证$G\vee(H\wedge E)=(G\vee H)\wedge(G\vee E)$成立。

(5) 对任意赋值v，

$$v(\sim(\sim G))=(v(\sim G))'=1-v(\sim G)=1-(1-v(G))=v(G)$$

所以$\sim(\sim G)=G$。

定义 4.1.4 若干个命题文字的析取式称为一个命题子句。特别地，没有命题文字的命题子句称为空子句，记为□。

定义 4.1.5 若干个命题子句的合取式称为合取范式。

定理 4.1.2 系统 OPL 中的每一个命题公式都可以化为与之等价的合取范式。

由定理 4.1.1 容易得到此结论。

每个命题公式都可以化成合取范式，由命题公式的构成以及定理 4.1.1 中性质知，合取范式中每一个命题子句都有如下的形式：

$G_1 \vee G_2 \vee \cdots \vee G_m$

其中 $G_i(i=1,2,\cdots,m)$是命题文字。

定义 4.1.6 称命题公式的合取范式中的命题子句构成的集合为该命题公式的命题子句集。

定义 4.1.7 设 G 是一个命题公式，门槛 $\lambda\in[0,1]$，G 称为 λ-恒假，当且仅当对任意赋值 v，都有 $v(G)\leq \lambda$; G 称为 λ-恒真，当且仅当对任意赋值 v，都有 $v(G)\geq\lambda$。若存在赋值 v，使得 $v(G)>\lambda$，称命题公式 G 是 λ-可满足；若对任意赋值 v，使得 $v(G)<\lambda$，称命题公式 G 是 λ-不可满足。

定理 4.1.3 设 S 是命题公式 G 对应的命题子句集，$\lambda\in[0,1]$，则 G 是 λ-恒假当且仅当 S 是 λ-恒假。

证明：设 $G=G_1\wedge G_2\wedge\cdots\wedge G_m$，其中 $G_i(i=1,2,\ldots,m)$是命题子句，则 $S=\{G_1, G_2, \ldots, G_m\}$。

对任意赋值 v，$v(G)= v(G_1\wedge G_2\wedge\cdots\wedge G_m)$

$$=v(G_1)*v(G_2)*\ldots*v(G_m)$$

若 S 是 λ-恒假，则存在 $i\in\{1,2,\ldots,m\}$，使得 $v(G_i)\leq\lambda$，所以 $v(G)\leq\lambda$，即 G 是 λ-恒假。

反之，若 G 是 λ-恒假，则对任意赋值 v，有 $v(G)\leq\lambda$，故存在 $j\in\{1,2,\ldots,m\}$，使得 $v(G_j)\leq\lambda$，所以 S 是 λ-恒假。

由该定理知，判断命题公式 G 的 λ-恒假性，可转化为判定其命题子句集 S 的 λ-恒假性。

4.2 算子命题逻辑的 λ-归结

定义 4.2.1 称 $\lambda_1 p$ 与～$(\lambda_2 p)$为 λ-互补文字，$\lambda_1 p$ 和 $\lambda_2 p$ 或～$(\lambda_1 p)$和～$(\lambda_2 p)$为 λ-相似文字，若 $\lambda_1, \lambda_2 \geq\lambda$ 或 $\lambda_1, \lambda_2 \leq 1-\lambda$。称 $\lambda^* p$ 是 λ-无关文字，若 $1-\lambda\leq\lambda^*\leq\lambda(\lambda\geq 0.5)$，或 $\lambda\leq\lambda^*\leq 1-\lambda\ (\lambda\leq 0.5)$。

定义 4.2.2 设 G_1，G_2 是两个命题子句，且 $G_1=\lambda_1 p_1\vee G_1^*$，$G_2=\sim(\lambda_2 p_2)\vee G_2^*$，若 $\lambda_1 p_1$ 和～$(\lambda_2 p_2)$是 λ-互补文字，则称子句$(G_1^*-S_1)\vee(G_2^*-S_2)$为 G_1 和 G_2 的 λ-归结式，记为 $R_\lambda(G_1, G_2)$。其中 $S_1=\{\lambda^* p|\lambda^* p$ 是 G_1 中的命题文字，且 $\lambda^* p$ 与 $\lambda_1 p_1$ 是 λ-相似文字$\}$，$S_2=\{\sim(\lambda^* p)|\ \sim(\lambda^* p)$是 G_2 中的命题文字，且～$(\lambda^* p)$与～$(\lambda_2 p_2)$是 λ-相似文字$\}$，$G_1^*-S_1$ 表示在命题公式 G_1^*中去除 S_1 中元素后的命题子

句，$G_2^*-S_2$表示在命题公式G_2^*中去除S_2中元素后的命题子句。

定理 4.2.1 设G_1，G_2是两个命题子句，$R_\lambda(G_1，G_2)$为G_1和G_2的λ-归结式，若G_1和G_2为λ-恒真，则$R_\lambda(G_1，G_2)$为λ-恒真。

证明：不妨设$G_1=\lambda_1p_1\vee G_1^*$，$G_2=\sim(\lambda_2p_2)\vee G_2^*$，$\lambda_1p_1$和$\sim(\lambda_2p_2)$是λ-互补文字，而$G_1^*$与$G_2^*$不含与互补文字相似的文字，则$R_\lambda(G_1，G_2)= G_1^*\vee G_2^*$。若$G_1$和$G_2$为λ-恒真，则对任意赋值$v$，有$v(G_1)\geq\lambda$，$v(G_2)\geq\lambda$。若$R_\lambda(G_1，G_2)$不是λ-恒真，则存在赋值$\tilde{v}$，使得$\tilde{v}(R_\lambda(G_1，G_2))<\lambda$，即$\tilde{v}(G_1^*)\oplus\tilde{v}(G_2^*)<\lambda$，所以$\tilde{v}(G_1^*)<\lambda$，且$\tilde{v}(G_2^*)<\lambda$。若$\lambda_1\geq\lambda,\lambda_2\geq\lambda$，互补文字在赋值$\tilde{v}$下一个大于$\lambda$，一个小于$\lambda$，故在赋值$\tilde{v}$下，$G_1$与$G_2$不全大于$\lambda(\lambda\neq0.5)$，这与$G_1$和$G_2$为λ-恒真矛盾，$R_\lambda(G_1，G_2)$为λ-恒真。

定义 4.2.3 称由λ-无关文字组成的命题子句为λ-空子句，记为λ-□。

定义 4.2.3 设 S 是命题子句集，称有限子句列G_1，G_2，…，G_k为从 S 到G_k的λ-归结演绎，若$G_i\in S(i\leq k)$或$G_i=R_\lambda(G_j，G_h)(j，h\leq i)$。

定理 4.2.2 设 S 是命题子句集，任取 S 中命题子句的命题文字λ^*p，对 S 中的命题子句删除λ^*p及其λ-相似文字，再在剩下的命题子句中删除$\sim(\lambda^*p)$及其λ-相似文字，所得命题子句集记为S'。若 S 是λ-恒假，则S'是λ-恒假。

证明：不妨假定 S 中不含与λ^*p及$\sim(\lambda^*p)$的λ-相似文字，因为作λ-归结时，归结文字的λ-相似文字会被删除。S 中子句分为三类：含λ^*p的命题子句，含$\sim(\lambda^*p)$的命题子句，不含λ^*p与$\sim(\lambda^*p)$的命题子句。可设$S=(\lambda^*p\vee G_1)\wedge(\sim(\lambda^*p)\vee G_2)\wedge G_3$，其中$G_1$，$G_2$，$G_3$中不含$\lambda^*p$与$\sim(\lambda^*p)$，且具有合取范式的形式。按题设$S'=G_1\wedge G_2\wedge G_3$，若 S 是λ-恒假，而$S'$不是λ-恒假，则存在赋值$v$，使得$v(S')>\lambda$，即有

$$v(G_1\wedge G_2\wedge G_3)>\lambda$$

而

$$\begin{aligned}&v(G_1\wedge G_2\wedge G_3)\\=&v(G_1)*v(G_2)*v(G_3)\\=&\min\{v(G_1),v(G_2),v(G_3)\}\end{aligned}$$

所以$v(G_1)>\lambda$，$v(G_2)>\lambda$，$v(G_3)>\lambda$。

而
$$\begin{aligned}v(S)=&v((\lambda^*p\vee G_1)\wedge(\sim(\lambda^*p)\vee G_2)\wedge G_3)\\=&(v(\lambda^*p)\oplus v(G_1))*(v(\sim(\lambda^*p))\oplus v(G_2))*v(G_3)\end{aligned}$$

因为$v(\lambda *p)\oplus v(G_1)=\max\{v(\lambda *p), v(G_1)\}>\lambda$，

$v(\sim(\lambda *p)) \oplus v(G_2)=\max\{v(\sim((\lambda *p)), v(G_2)\}>\lambda$，

所以$v(S)>\lambda$，此与 S 是λ-恒假矛盾，故假定 S′ 不是λ-恒假不成立，故 S′ 是λ-恒假。

定理 4.2.3 设 S 是命题子句集，若 S 是λ-恒假，则存在从 S 推出λ-□或□的λ-归结演绎。

证明：假设 S 中每一个命题都不含λ-相似文字，M 是 S 中命题原子集，对 M 的元素数|M|用归纳法。

当|M|=1 时，因 S 是λ-恒假，所以可设 $S=\{G_1=\lambda_1 p, G_2=\sim(\lambda_1 p)\}$，显然 G_1，G_2，$R_\lambda(G_1, G_2)$是一个从 S 推出□的λ-归结演绎。

假设当$|M|<n(n\geq 2)$时，结论成立。

当$|M| = n$ 时，取 M 中任一原子命题λ*p，令 S_1 是如下命题子句集：将 S 中的所有的 λ*p 及其 λ-相似文字删除，然后在剩下的命题子句中删除命题文字～(λ*p)及其λ-相似文字。由定理 4.2.2 知，S_1 是λ-恒假，且所含命题原子的个数小于 n。根据归纳假设，存在从 S_1 推出λ-□或□的λ-归结演绎 D_1。

在 D_1 中，将 S_1 中所有不是 S 中的命题子句，通过添入文字λ*p 及其λ-相似文字而恢复成 S 中命题子句，于是得到从 S 推出λ-□或□的λ*p 的λ-归结演绎 D_2。然后在 D_2 中所有不是 S 中的命题子句，通过添入文字～(λ*p)及其λ-相似文字而恢复成 S 中命题子句，于是得到从 S 推出λ-□或□的～(λ*p)的λ-归结演绎 D_3，显然从 D_2 与 D_3 不难得到从 S 推出λ-□或□的λ-归结演绎 D。

定理 4.2.4 设 S 是命题子句集，若存在从 S 推出λ-□或□的λ-归结演绎，则 S 是λ-恒假。

证明：若命题子句集 S 不是λ-恒假，那么存在赋值v，使得$v(S)>\lambda$。设 $S=\{G_1, G_2, \cdots, G_m\}$，有 $S=G_1\wedge G_2\wedge\cdots\wedge G_m$，

则 $$\begin{aligned} v(S) &= v(G_1\wedge G_2\wedge\cdots\wedge G_m)\\ &=v(G_1)*v(G_2)*\cdots*v(G_m)\\ &=\min\{v(G_1), v(G_2), \cdots, v(G_m)\}>\lambda \end{aligned}$$

所以$v(G_i)>\lambda(i=1,2,\cdots,m)$，即 S 中命题子句都是λ-可满足的。

而从 S 推出λ-□或□的λ-归结演绎序列中，命题子句要么是 S 中命题子句，要么是从 S 中命题子句出发的λ-归结式，由定理 4.2.1 知，其λ-归结式是λ-可

满足的，从而 λ-□或□是 λ-可满足的，显然这是矛盾的，所以 S 是 λ-恒假。

定理 4.2.5 若 $\lambda_1 p$，$\lambda_2 p$，…，$\lambda_m p$ 是 λ-相似文字，若 $\lambda_i \neq 0.5$，则它们既不是 λ-恒假子句，又不是 λ-恒真子句。

证明：因为 $v(\lambda_i p)=\begin{cases}\lambda_i & p\text{被指定为}1\\ 1-\lambda_i & p\text{被指定为}0\end{cases}$ (i=1,2, …,m)， 故当 $\lambda_i \neq 0.5$ 时，$\lambda_1 p$，$\lambda_2 p$，…，$\lambda_m p$ 既不是 λ-恒假命题子句，又不是 λ-恒真命题子句。

若记 $\tilde{S}$ 为将 S 中的形如 λ*p 的命题文字在 $\lambda^* \geq \lambda$ 时用 p 代替，否则用～p 代替；将命题文字～(λ*p) 在 $\lambda^* \geq \lambda$ 时用～p 代替，否则用 p 代替，那么 $\tilde{S}$ 是经典逻辑中的基子句集，则有以下结论成立。

定理 4.2.6 算子命题逻辑中，命题子句集 S 是 λ-恒假的当且仅当经典逻辑中的基子句集 $\tilde{S}$ 是不可满足的。

证明：若将 S 中的命题文字 λ*p，当 p 指定为 1，$\lambda^* \geq \lambda$ 时，其赋值大于 λ，为 λ-可满足，与 p 在二值逻辑中的解释 1 对应；而在 $\lambda^* \leq \lambda$ 时，其赋值小于 λ，对该赋值为 λ-不可满足，与～p 在二值逻辑中的解释-1 对应。对于命题文字～(λ*p)有其相应的真值对应。所以算子命题逻辑中的命题子句集 S 是 λ-恒假的当且仅当经典逻辑中的基子句集 $\tilde{S}$ 是不可满足的。

该定理将命题子句的 λ-恒假性的判定转化为一般基子句集的不可满足性的判定，而一般基子句集的不可满足性是可以用矩阵归结或删除归结进行判定的，从而可以对命题子句的 λ-恒假性进行判定。

上述定理是从语义的角度分析了 λ-归结的完备性与可靠性。在下文中，我们欲直接从命题子句集 S 出发，从语法的角度，以 Petri 网为工具，讨论算子命题逻辑的 λ-归结的 Petri 网实现。

4.3 算子命题逻辑公式的 Petri 网模型

对于算子命题逻辑，它既不同于一阶逻辑，又有别于命题逻辑，因此它的 Petri 网模型必须重新定义。在给出 Petri 网模型之前，我们先讨论命题子句的另一种表达形式，以便于命题子句的 Petri 网表示。

4.3.1 命题公式的极简规则型命题子句集

定义 4.3.1 称命题公式 $G \rightarrow H$ 为规则型命题公式，若 G 或 H 是命题文字的合取式或析取式。

注记 4.1 在定义 4.3.1 中，允许命题公式 $G \rightarrow H$ 中的 G 和 H 之一可以不出现，此时无连接词"→"。

定理 4.3.1 在算子命题逻辑中，设 G 是一个命题子句，则存在规则型命题公式 G_1'，G_2'，…，G_m'，使得

$$G = G_1' \wedge G_2' \wedge \cdots \wedge G_m'$$

证明：由于 $\sim(\lambda p) = (1-\lambda)p$，所以可以假定命题子句 G 中不含逻辑连接词"～"。

设 h 是命题子句 G 中所含原子命题的个数，我们对 h 用数学归纳法。

(1) 当 $h=1$ 时，G 显然是一个规则型命题公式。

当 $h=2$ 时，设 $G=\lambda_1 p_1 \vee \lambda_2 p_2$，因为

$$G=\lambda_1 p_1 \vee \lambda_2 p_2 = \sim((1-\lambda_1)p_1) \vee \lambda_2 p_2 = (1-\lambda_1)p_1 \rightarrow \lambda_2 p_2$$

所以 G 是一个规则型命题公式。

(2) 假设结论在 $h<k$ 时成立，当 $h=k$ 时，设

$$G = \lambda_1 p_1 \vee \lambda_2 p_2 \vee \cdots \vee \lambda_{k-1} p_{k-1} \vee \lambda_k p_k$$

由假设知，存在命题公式 G_1 与 G_2，使得

$$G = (G_1 \rightarrow G_2) \vee \lambda_k p_k$$

其中 G_1 或 G_2 是合取范式或析取范式。

而

$$\begin{aligned} G &= (G_1 \rightarrow G_2) \vee \lambda_k p_k \\ &= (\sim G_1 \vee G_2) \vee \lambda_k p_k \\ &= \sim G_1 \vee (G_2 \vee \lambda_k p_k) \\ &= G_1 \rightarrow (G_2 \vee \lambda_k p_k) \end{aligned}$$

$G_2 \vee \lambda_k p_k$ 是命题公式，总可以化为合取范式，故当 $h=k$ 时结论成立。

由归纳假设知，结论成立。

定理 4.3.2 设 G，H，E 均为命题公式集 PL 中的命题公式，则下列结论成立：

(1) $G \rightarrow (H \wedge E) = (G \rightarrow H) \wedge (G \rightarrow E)$;

(2) $G\to(H\vee E)=(G\wedge\sim H)\to E$;

(3) $(G\vee H)\to E=(G\to E)\wedge(H\to E)$。

证明(1) 因为

$$
\begin{aligned}
&G\to(H\wedge E)\\
=&\sim G\vee(H\wedge E)\\
=&(\sim G\vee H)\wedge(\sim G\vee E)\\
=&(G\to H)\wedge(G\to E)
\end{aligned}
$$

所以 $G\to(H\wedge E)=(G\to H)\wedge(G\to E)$成立。

(2) 因为

$$
\begin{aligned}
&G\to(H\vee E)\\
=&\sim G\vee(H\vee E)\\
=&\sim(G\wedge\sim H)\vee E\\
=&(G\wedge\sim H)\to E
\end{aligned}
$$

所以 $G\to(H\vee E)=(G\wedge\sim H)\to E$ 成立。

(3) 因为

$$
\begin{aligned}
&(G\vee H)\to E\\
=&\sim(G\vee H)\vee E\\
=&(\sim G\wedge\sim H)\vee E\\
=&(\sim G\vee E)\wedge(\sim H\vee E)\\
=&(G\to E)\wedge(H\to E)
\end{aligned}
$$

所以$(G\vee H)\to E=(G\to E)\wedge(H\to E)$ 成立。

定义 4.3.2 称命题公式 $G\to H$ 为极简规则型命题公式，若 G 是命题原子的合取式，H 最多含一个命题原子。

定理 4.3.3 在算子命题逻辑中，每一个命题子句 G 都可以化为与之等价的极简规则型命题公式的合取。

证明：根据定理 4.3.1，存在规则型命题公式 G_1'，G_2'，…，G_m'，使得

$$G=G_1'\wedge G_2'\wedge\cdots\wedge G_m'$$

由定理 4.3.2，将命题子句 G 化为与之等价的极简规则型命题公式的合取。

定义 4.3.3 设 G 是命题子句，若存在极简规则型命题公式 G_1'，G_2'，…，G_m'，使得 $G=G_1'\wedge G_2'\wedge\cdots\wedge G_m'$，称命题公式集$\{G_1', G_2', \cdots, G_m'\}$为命题子句 G 的极简规则型命题子句集，记为 G_R。

4.3.2 算子命题逻辑公式的 Petri 网模型

现在，我们可以着手讨论命题子句的 Petri 网表示。由定理 4.3.3，每个命题子句可化为与之等价的极简规则型命题公式的合取，那么命题子句的 Petri 网表示就转化为如何表示极简规则型命题子句集，随之转化为如何表示极简规则型命题公式。而极简规则型命题公式只有两种形式，以含一个逻辑连接词“∧”为例为：$\lambda_1 p_1 \wedge \lambda_2 p_2$ 或 $\lambda_1 p_1 \wedge \lambda_2 p_2 \rightarrow \lambda^* q$。这两种极简规则型命题公式用 Petri 网分别表示为：

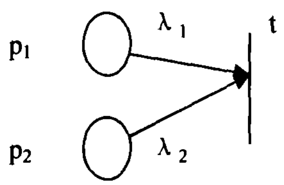


图 4-1 命题公式 $\lambda_1 p_1 \wedge \lambda_2 p_2$ 的 Petri 网模型

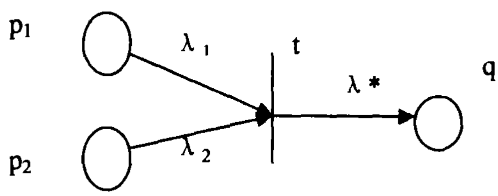


图 4-2 命题公式 $\lambda_1 p_1 \wedge \lambda_2 p_2 \rightarrow \lambda^* q$ 的 Petri 网模型

在上述 Petri 网模型中，将形如 λp 的命题原子中的 λ 置于与位置结点 p 相连的弧连接上，即作为连接权，这样的处理是恰当的，因为算子 λ 可认为是原子命题 p 的信任程度，反映到 Petri 网模型上，即是原子命题 p 对所在极简规则型命题公式的连接强度，由此给出命题子句的 Petri 网模型。

定义 4.3.4 命题子句的 Petri 网模型为五元组 OPN={P，T，F，W，M}，其中：

$P=\{p_1,p_2,\ldots,p_m\}$ 为位置结点集，每个位置结点表示命题子句中的原子命题，m 为命题子句中的原子命题的个数；

$T=\{t_1,t_2,\ldots,t_n\}$ 为转移结点集，每个转移结点对应一个极简规则型命题公式，n 为极简规则型命题公式的个数；

$F\subseteq(P\times T)\cup(T\times P)$ 为有向弧集；

W：$F\rightarrow[0,1]$ 将弧集映射到算子集上，每个取值与前面它相应的原子命题

构成规则中的命题原子；

M：$P \rightarrow N \cup \{0\}$将标识放于位置结点集中，记 M_0 为初始标识。

对于命题子句集 S 的 Petri 网模型，只需要将 S 中的每个命题子句的 Petri 网联结起来即可。

4.4 算子命题逻辑的 Petri 网推理算法

这一节，将根据命题子句集的 Petri 网模型，讨论两种算子命题逻辑的 Petri 网推理算法。

4.4.1 算子命题逻辑的 T-不变量推理算法

从定义 4.3.4 知，命题逻辑子句集的 Petri 网模型中，每个转移结点的后集最多只含一个元素(目标转移结点的后集为空)，如果所有的连接权都大于或等于 λ，则此时的 Petri 网有同二值逻辑中 Horn 基子句集的 Petri 网相似的功能。如果定义相应的关联矩阵和 T-不变量，那么可以利用命题子句集的 T-不变量来讨论命题子句集的 λ-恒假性，从而完成命题逻辑系统中的 λ-归结推理。

假定以下讨论的命题子句的 Petri 网模型的连接权都大于或等于 λ，且 λ ≥ 0.5 (λ ≤ 0.5 时讨论类似)。这种情况下，命题子句中不含有 λ-相似文字。

定义 4.4.1 矩阵 $C=[c_{ij}]$称为命题子句的 Petri 网模型的关联矩阵，如果

$$c_{ij}=\begin{cases}1 & \text{若} W(t_i,p_j)-W(p_j,t_i)>\lambda \\ -1 & \text{若} W(t_i,p_j)-W(p_j,t_i)<-(1-\lambda) \\ 0 & \text{其它}\end{cases},$$

其中$W(t_i,p_j)$,$W(p_j,t_i)$分别表示连接t_i到p_j以及p_j到t_i的连接权。

命题子句的 Petri 网模型中，关联矩阵 C 的 T-不变量的定义以及求 T-不变量的方法同于一般的 Petri 网模型中的定义 1.4.5 以及算法 1.1。

定义 4.4.2 若一个极简规则型命题公式中出现的算子都大于或等于λ，这样的命题公式称为 Horn 极简规则型命题公式，由 Horn 极简规则型命题公式构成的集合称为 Horn 型命题子句集。

定理 4.4.1 Horn 型命题子句集 S 的 Petri 网模型的关联矩阵 C 中存在 T-不变量$X>0$的充要条件是存在一个从 S 推出□的 λ-归结演绎。

证明：若存在 T-不变量$X>0$，根据关联矩阵的定义，算法 1.1 中 2.1)参加

运算的两行的线性组合的过程与两命题子句的λ-归结对应，循环的过程对应λ-归结演绎过程。存在 T-不变量$X>0$，表明能归结出空子句□，所以存在一个从 S 推出□的λ-归结演绎。

反之，若存在一个从 S 推出□的λ-归结演绎，那么选择λ-归结演绎中归结文字出现的先后顺序可构造命题子句集的关联矩阵 C，因为一个子句集的关联矩阵本身就不是唯一的。按照算法 1.1 中的循环过程，由于□对应的是算法 1.1 中 A 的位置有零行，那么算法 1.1 中 D 的行即为关联矩阵 C 的 T-不变量X，且$X>0$。

定理 4.4.2 Horn 型命题子句集 S 是λ-恒假的充要条件是 S 的 Petri 网模型的关联矩阵 C 中存在 T-不变量$X>0$

证明：若 S 是λ-恒假的， 则由定理 4.2.3 知，存在一个从 S 推出□的λ-归结演绎。又由定理4.4.1，S 的 Petri 网模型的关联矩阵 C 中存在 T-不变量$X>0$。

反之，若 S 的 Petri 网模型的关联矩阵 C 中存在 T-不变量$X>0$。由定理 4.4.1，存在一个从 S 推出□的λ-归结演绎。又由定理 4.2.4 知，命题子句集 S 是λ-恒假的。

例 4.4.1 设 S = {0.8A→0.1DR∨0.98CR, 0.75DR ∨0.8 CR →0.9BR，0.97 BR∧0.85A →0.85DR∨0.2 CR, 0.87A, 0.25DR}，判定 S 是否为 0.7-恒假 。

解：λ=0.7

$$C=\begin{array}{c}\\ t_1\\ t_2\\ t_3\\ t_4\\ t_5\\ t_6\end{array}\begin{array}{c}\begin{array}{cccc}A & DR & CR & BR\end{array}\\ \begin{bmatrix}1 & 0 & 0 & 0\\ -1 & 1 & -1 & -1\\ -1 & -1 & 1 & 0\\ 0 & 0 & -1 & 1\\ 0 & -1 & 0 & 1\\ 0 & -1 & 0 & 0\end{bmatrix}\end{array}$$

为了简化计算，直接对 C 应用算法 1.1 计算 T-不变量。

$$\longrightarrow\begin{array}{c}\\ t_1+t_2\\ t_1+t_3\\ t_4\\ t_5\\ t_6\end{array}\begin{array}{c}\begin{array}{cccc}A & DR & CR & BR\end{array}\\ \begin{bmatrix}0 & 1 & -1 & -1\\ 0 & -1 & 1 & 0\\ 0 & 0 & -1 & 1\\ 0 & -1 & 0 & 1\\ 0 & -1 & 0 & 0\end{bmatrix}\end{array}$$

$$\begin{array}{cc} & \begin{array}{cccc} A & DR & CR & BR \end{array} \\ \begin{array}{l} t_1+t_2 \\ t_1+t_3 \\ t_4 \\ t_5 \\ t_1+t_2+t_6 \end{array} & \longrightarrow \begin{bmatrix} 0 & 1 & -1 & -1 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 1 \\ 0 & -1 & 0 & 1 \\ 0 & 0 & -1 & -1 \end{bmatrix} \end{array}$$

该矩阵行的线性组合不可能有零行产生，所以矩阵 C 没有 T-不变量，故子句集 S 不是 0.7-恒假。

根据定理 4.4.2，T-不变量法只适用于对 Horn 型命题子句集 λ-恒假性的判定。对于一般的命题子句集，由定理 4.3.3，每一个命题子句都可以化为与之等价的极简规则型命题公式的合取，但是极简规则型命题公式不一定是 Horn 极简规则型命题公式，如命题公式 $0.97\ A \wedge 0.25B \rightarrow 0.85C(\lambda = 0.7$ 时)，这就限制了 T-不变量法的适用范围。为了解决这个问题，下面讨论删除推理算法。

4.4.2 算子命题逻辑的删除推理算法

在第 3 章，我们提出了一种直接利用一般子句集的 Petri 模型，通过删除模型的元素—位置、转移以及弧连接达到推理的目的。本节借用命题逻辑的 Petri 网模型的删除算法来完成算子命题逻辑的 Petri 网推理。

由定理 4.3.3，每一个命题子句都可以化为与之等价的极简规则型命题公式的合取。所以，这一节的讨论针对命题子句的极简规则型命题子句集，即对一般命题子句集进行讨论。

记*P 为位置结点 P 的前集，P*为位置结点 P 的后集。

算法 4.1

(1) 设 P 为一位置结点，且 $|{*P}| = k\ (k \geqslant 1)$, $|P{*}| = m\ (m \geqslant 1)$ ，不妨设 ${*P} = \{t_1, t_2, \ldots, t_k\}$，$P{*} = \{t_1', t_2', \ldots, t_m'\}$，若 $W(t_i,P)$与 $W(P, t_j')$同大于等于 λ，或同小于等于 $1-\lambda$，那么合并转移结点 t_i 与 t_j' 为结点 t_{ij}，同时删除结点 t_i 与 t_j'，以及弧(t_i, P) 与(P, t_j)，若此时位置 P 的前集与后集为空，这时删除该位置结点 P，最后将一个标识放入位置结点集$(t_{ij})*$的每一个元素中；

(2) 如果 $t{*} \cap {*t} \neq \Phi$，且对 $P \in t{*} \cap {*t}$，连接位置结点 P 的权同大于

等于 λ，或同小于等于 1-λ (即转移结点 t 表示的命题公式为 λ-恒真)，则删除转移结点 t 及其所有的弧连接；

(3) 如果$|{}^*P|=|P^*|=0$，则删除位置接点 P；

(4) 若$({}^*t_1 \cup t_1{}^*) \subset ({}^*t_2 \cup t_2{}^*)$，则删除转移结点 t_2 及其所有的弧连接。

定理 4.4.3 设 OPN′ 是由极简规则型命题子句集 S 的 Petri 网模型 OPN 经算法 4.1 作用后所得的 Petri 网模型，若 OPN′ 表示的命题子句集是 λ-恒假，则模型 OPN 表示的命题子句集是 λ-恒假。

证明：算法 4.1 中(1)中的删除过程对应的是命题子句的 λ-归结过程，删除的是 S 中的子句，新产生的是它们的 λ-归结式。若 OPN′ 表示的命题子句集是 λ-恒假，由定理 4.2.1 知，模型 OPN 表示的命题子句集是 λ-恒假。

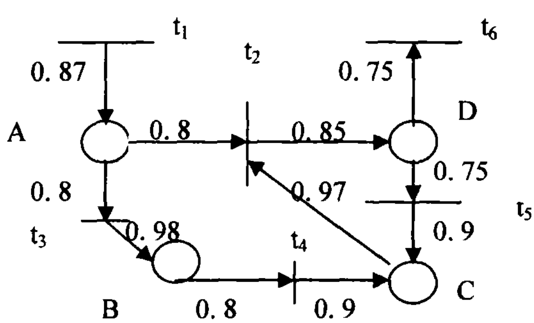
定理 4.4.4 极简规则型命题子句集 S 是 λ-恒假的充要条件是 S 的 Petri 网模型经删除算法 4.1 后可得空转移结点。

证明：(充分性) 若极简规则型命题子句集 S 的 Petri 网模型经删除算法 4.1 后可得空结点，由于空结点对应的是空子句，为 λ-恒假，由定理 4.4.3 知，S 是 λ-恒假。

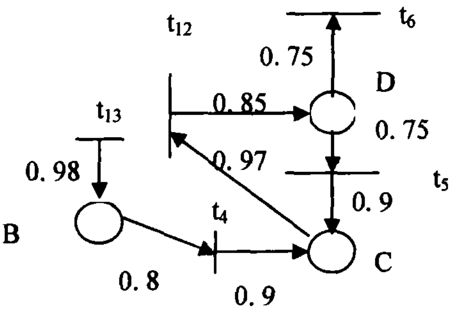
(必要性) 若极简规则型命题子句集 S 是 λ-恒假，由定理 4.2.3，存在从 S 推出□的 λ-归结演绎，而 λ-归结演绎推理过程即是反复算法 4.1(1)的过程，□对应空转移结点，所以 S 的 Petri 网模型经删除算法 4.1 后可得空转移结点。

例 4.4.2 设 S = {0.87A，0.8A ∧0.97C→0.85D，0.8A→0.98B，0.8B→0.9C，0.75D→0.9C，0.75D}，取 λ=0.7。

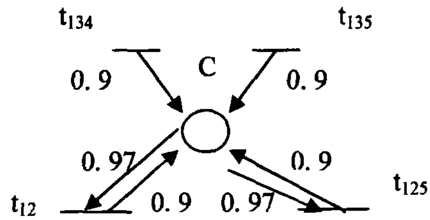
S 的 Petri 网模型为：



⇒



⇒ t_{136}



⇒ t_{136}

由于得到空转移接点 t_{136}， 所以 S 为 0.7-恒假。

第 5 章 算子模糊逻辑系统及其 Petri 网推理算法

上一章，讨论了算子命题逻辑系统。这一章，讨论更广泛的算子逻辑系统，即算子模糊逻辑系统(与文献[45]中称谓同，但定义不一样)。形如: X_1 is $P_{m_1}(\lambda_1)$，X_2 is $P_{m_2}(\lambda_2)$，…，X_k is $P_{m_k}(\lambda_k)$，then Y is $P_{m_{k+1}}(\lambda_{k+1})$的规则，将它表示为 $\lambda_1 P_{m_1}(X_1)\wedge\lambda_2 P_{m_2}(X_2)\wedge\cdots\wedge\lambda_k P_{m_k}(X_k)\to\lambda_{k+1}P_{m_{k+1}}(Y)$是合理的。算子模糊逻辑系统主要讨论这种表示的逻辑性质，以及 Petri 网表示和基于 Petri 网的归结推理。本章主要研究：

1. 算子模糊逻辑系统；
2. 算子模糊逻辑的 λ-归结；
3. 算子模糊逻辑的 Petri 网模型；
4. 基于 Petri 网模型的算子模糊逻辑的 λ-归结推理。

5.1 算子模糊逻辑系统 OFL

一、系统 OFL 的语言

1. 符号

系统 OFL 的符号包括下列七类：

i) 个体变元集：$V=\{x_i|i\in N\}$；

ii) 个体常元集：$C=\{c_k|k\in K\}$，其中K是一个指标集；

iii) 函数符集：$\{f_{m_j}|j\in J\}$，其中J为指标集，m_j称为函数f_{m_j}的元数，$m_j\in N^+$；

iv) 关系符集：$\{P_{m_i}|i\in I\}$，其中I为指标集，m_j称为函数P_{m_j}的元数，

$m_j \in N^+$；

v) 逻辑连接词：$\vee$，$\wedge$，$\sim$，$\rightarrow$；

vi) 量词：$\forall$，$\exists$；

vii) 技术性符号：(，)。

2. 项的构成

OFL 中的项集 T 是满足下列条件的最小集 T^*：

i) $V \cup C \subseteq \mathrm{T}^*$；

ii) 对于任意 $j \in J$，且 $t_1, t_2, \cdots, t_{m_j} \in \mathrm{T}^*$，则 $f_{m_j}(t_1, t_2, \cdots, t_{m_j}) \in \mathrm{T}^*$，其中 J 为指标集。

3. 算子模糊公式的构成

设([0，1]，*，⊕，′，0，1)是由 4.1 节定义的完备有余格。

定义 5.1.1 若 $P_{m_i}(x_1, x_2, \cdots, x_{m_i})$ 是 m_i 元谓词符号，$t_1, t_2, \cdots, t_{m_i}$ 是项，对任意 $\lambda \in [0, 1]$，称 $\lambda P_{m_i}(t_1, t_2, \cdots, t_{m_i})$ 是模糊原子。

模糊原子是 OFL 中最基本的元素。

算子模糊公式是被递归定义如下：

i) 模糊原子 $\lambda P_{m_i}(t_1, t_2, \cdots, t_{m_i})$ 是算子模糊公式；

ii) 若 G 是算子模糊公式，则～G 是算子模糊公式；

iii) 若 G，H 是算子模糊公式，则 G∧H，G∨H，G→H 是算子模糊公式；

iv) 若 G 是算子模糊公式，x 是 G 中自由变元，则 $(\forall x)$G 与 $(\exists x)$G 是算子模糊公式；

v) 所有公式都是有限次使用 i)- iv)生成的符号串。

定义 5.1.2 称 $\lambda P_{m_i}(t_1, t_2, \cdots, t_{m_i})$ 与～$\lambda(P_{m_i}(t_1, t_2, \cdots, t_{m_i}))$ 为算子模糊文字。

二、OFL 的解释

系统 OFL 的一个解释是如下形式的四元组 D：

$$D = \langle D, f_D, \{P_{iD}; i \in I\}, \{f_{jD}; j \in J\} \rangle$$

其中：

(i) D是一非空集合，称为解释域；

(ii) $f_D : V \cup C \to D$是一函数；

(iii) 对于任意$j \in J$，$f_{jD} : D^{m_j} \to D$是一m_j元函数；

(iv) 对于任意$i \in I$，$P_{iD} : D^{m_i} \to \{0,1\}$是一$m_i$元函数。

在解释D之下，每一算子模糊公式 F 对应[0，1]中一元素v(F)，称为算子模糊公式 F 在解释D之下的真值，其中：

(i) 若 F 为$\lambda P_{m_i}(t_1,t_2,\cdots,t_{m_i})$，则

$$v(\lambda P_{m_i}(t_1,t_2,\cdots,t_{m_i})) = \begin{cases} \lambda, & 若P_{iD}(D(t_1),D(t_2),\cdots,D(t_{m_i})) = 1 \\ 1-\lambda, & P_{iD}(D(t_1),D(t_2),\cdots,D(t_{m_i})) = 0 \end{cases},$$

其中$t_1,t_2,\cdots,t_{m_j}$是项。

(ii) 若 F 为算子模糊公式～G，则v(～G) = (v(G))$'$；

(iii) 若 F 为算子模糊公式 G∧H，则v(G∧H) = v(G) * v(H)；

(iv) 若 F 为算子模糊公式 G∨H，则v(G∨H) = v(G) ⊕ v(H)；

(v) 若 F 为算子模糊公式 G→H，则v(G→H) = v(～G∨H)；

(vi) 若 F 为算子模糊公式$(\forall x)$G，则$v((\forall x)\mathrm{G}) = \inf\limits_{x\in D}\{v(\mathrm{G})\}$；

(vii) 若 F 为算子模糊公式$(\exists x)$G，则$v((\exists x)\mathrm{G}) = \sup\limits_{x\in D}\{v(\mathrm{G})\}$。

定义 5.1.3 算子模糊公式 G 与 H 称为等价，如果对任意解释D的赋值v，都有v(G) = v(H)，记为 G=H。

定理 5.1.1 设 G，H，E 为算子模糊公式，不含自由变元，则下列结论成立：

(1) ～(G∧H) =～G∨～H， ～(G∨H) = ～G∧～H；

(2) G→H =～G∨H；

(3) G∧(H∨E) = (G∧H)∨(G∧E)，G∨(H∧E) = (G∨H)∧(G∨E)；

(4) ～(～G) = G。

证明：(1) 对任意解释D的赋值v，

v(～G∨～H)

$=\nu(\sim G)\oplus\nu(\sim H)$

$=(\nu(G))'\oplus(\nu(H))'$

$=(\nu(G)*\nu(H))'$

$=(\nu(G\wedge H))'$

$=\nu(\sim(G\wedge H))$

所以$\sim(G\wedge H)=\sim G\vee\sim H$，同理可证$\sim(G\vee H)=\sim G\wedge\sim H$成立。

(2) 由公式 G→H 的解释以及定义 5.1.3 可证。

(3) 对任意解释D的赋值ν，

$\nu(G\wedge(H\vee E))$

$=\nu(G)*\nu(H\vee E)$

$=\nu(G)*(\nu(H)\oplus\nu(E))$

$=(\nu(G)*\nu(H))\oplus(\nu(G)*\nu(E))$

$=\nu(G\wedge H)\oplus\nu(G\wedge E)$

$=\nu((G\wedge H)\vee(G\wedge E))$

所以$G\wedge(H\vee E)=(G\wedge H)\vee(G\wedge E)$。

同理可证$G\vee(H\wedge E)=(G\vee H)\wedge(G\vee E)$。

(4) 对任意解释D的赋值ν，

$\nu(\sim(\sim G))$

$=(\nu(\sim G))'$

$=1-\nu(\sim G)$

$=1-(1-\nu(G))$

$=\nu(G)$

所以$\sim(\sim G)=G$。

定理 5.1.2 设 G 是仅含有自由变元x的算子模糊公式，记为$G(x)$，H 中不含自由变元x，则

(1) $\sim(\forall x)G(x)=(\exists x)(\sim G(x))$，$\sim(\exists x)G(x)=(\forall x)(\sim G(x))$；

(2) $(\forall x)G(x)\vee H=(\forall x)(G(x)\vee H)$；

(3) $(\exists x)G(x)\vee H=(\exists x)(G(x)\vee H)$；

(4) $(\forall x)G(x)\wedge H=(\forall x)(G(x)\wedge H)$;

(5) $(\exists x)G(x)\wedge H=(\exists x)(G(x)\wedge H)$ 。

证明：(1) 对任意解释 D 的赋值 v，

因为 $v((\exists x)(\sim G(x)))$

$$=\sup_{x\in D}\{\ v(\ \sim G(x))\}$$

$$=\sup_{x\in D}\{(v(G(x)))'\}$$

$$=(\inf_{x\in D}\{\ v(G(x))\})'$$

$$=v(\sim(\forall x)G(x))$$

所以 $\sim(\forall x)G(x)=(\exists x)(\sim G(x))$。

同理可证：$\sim(\exists x)G(x)=(\forall x)(\sim G(x))$。

(2) 对任意解释 D 的赋值 v，

$$v((\forall x)G(x)\vee H)$$

$$=v((\forall x)G(x))\oplus v(H)$$

$$=\inf_{x\in D}\{\ v(G(x))\}\oplus v(H)$$

$$=\inf_{x\in D}\{\ v(G(x))\oplus v(H)\}$$

$$=v((\forall x)(G(x)\vee H))$$

所以 $(\exists x)G(x)\vee H=((\exists x)(G(x)\vee H)$ 。

同理可证结论(4)成立。

(3) 对任意解释 D 的赋值 v，

$$v((\exists x)G(x)\vee H)$$

$$= v((\exists x)G(x)) \oplus v(H)$$

$$= \sup_{x\in D}\{v(G(x))\} \oplus v(H)$$

$$= \sup_{x\in D}\{v(G(x)) \oplus v(H)\}$$

$$= v((\exists x)(G(x) \vee H))$$

所以 $(\exists x)G(x) \vee H = ((\exists x)(G(x) \vee H)$。

同理可证结论(5)成立。

定理 5.1.3 设 G 与 H 是仅含有自由变元 x 的算子模糊公式，记为 $G(x)$ 与 $H(x)$，于是有：

(1) $(\forall x)G(x) \wedge (\forall x)H(x) = (\forall x)(G(x) \wedge H(x))$;

(2) $(\exists x)G(x) \vee (\exists x)H(x) = (\exists x)((G(x) \vee H(x))$;

(3) $(\forall x)G(x) \vee (\forall x)H(x) = (\forall x)(\forall y)(G(x) \vee H(y))$;

(4) $(\exists x)G(x) \wedge (\exists x)H(x) = (\exists x)(\exists y)(G(x) \wedge H(y))$ 。

证明：(1) 对任意解释 D 的赋值 v，

$$v((\forall x)G(x) \wedge (\forall x)H(x))$$

$$= v((\forall x)G(x)) * v((\forall x)H(x))$$

$$= \inf_{x\in D}\{v(G(x))\} * \inf_{x\in D}\{v(H(x))\}$$

$$= \inf_{x\in D}(v(G(x)) * v(H(x)))$$

$$= v((\forall x)(G(x) \wedge H(x)))$$

所以 $(\forall x)G(x) \wedge (\forall x)H(x) = (\forall x)(G(x) \wedge H(x))$。

(2) 对任意解释 D 的赋值 v，

$$\begin{aligned} & v((\exists x)\mathrm{G}(x) \vee (\exists x)\mathrm{H}(x)) \\ = & v((\exists x)\mathrm{G}(x)) \oplus v((\exists x)\mathrm{H}(x)) \\ = & \sup_{x\in D}\{v(\mathrm{G}(x))\} \oplus \sup_{x\in D}\{v(\mathrm{H}(x))\} \\ = & \sup_{x\in D}(v(\mathrm{G}(x)) \oplus v(\mathrm{H}(x))) \\ = & v((\exists x)(\mathrm{G}(x) \vee \mathrm{H}(x))) \end{aligned}$$

所以 $(\exists x)\mathrm{G}(x) \vee (\exists x)\mathrm{H}(x) = (\exists x)(\mathrm{G}(x) \vee \mathrm{H}(x))$。

(3) 对任意解释 D 的赋值 v，

$$\begin{aligned} & v((\forall x)(\forall y)(\mathrm{G}(x) \vee \mathrm{H}(y))) \\ = & \inf_{x\in D}\inf_{y\in D} v(\mathrm{G}(x) \vee \mathrm{H}(y)) \\ = & \inf_{x\in D}\inf_{y\in D} v(\mathrm{G}(x)) \oplus v(\mathrm{H}(y)) \\ = & \inf_{x\in D} v(\mathrm{G}(x)) \oplus \inf_{y\in D} v(\mathrm{H}(y)) \\ = & \inf_{x\in D} v(\mathrm{G}(x)) \oplus \inf_{x\in D} v(\mathrm{H}(x)) \\ = & v((\forall x)\mathrm{G}(x) \vee (\forall x)\mathrm{H}(x)) \end{aligned}$$

所以 $(\forall x)\mathrm{G}(x) \vee (\forall x)\mathrm{H}(x) = (\forall x)(\forall y)(\mathrm{G}(x) \vee \mathrm{H}(y))$。

同理可证结论(4)成立。

定理 5.1.4 OFL 系统中任一算子模糊公式，都存在与之等价的前束范式。

证明：(1) 利用定理 5.1.1 中的(2)，可将算子模糊公式中的 $\to$ 消去，得只含～，$\vee$，$\wedge$ 的等价算子模糊公式；

(2) 利用定理 5.1.1 中的(1)(4)以及定理 5.1.2 中的(1)，将量词提到～的前面；

(3) 利用定理 5.1.2 与定理 5.1.3 将量词提到算子模糊公式的最左边。

对于一个前束范式，可用文献[45]中给出的方法，得此前束范式的 Skolem 标准型。再将其母式化成等价的合取范式，故合取范式中每一个算子模糊子句都有如下形式：

$G_1 \vee G_2 \vee \ldots \vee G_m$

其中 $G_i(i=1,2,\ldots,m)$是算子模糊文字。

定义 5.1.4 设 G 是一个算子模糊公式，将其化为 Skolem 标准型，设其母式为 M，M 给出的算子模糊子句集称为算子模糊公式 G 的算子模糊子句集。

定义 5.1.5 设 G 是一个算子模糊公式，门槛 $\lambda\in[0,1]$，称 G 为 λ-恒假(λ-不可满足的)，当且仅当对任意解释 D，都有 $v(G)\le\lambda$；G 称为 λ-恒真，当且仅当对任意解释 D，都有 $v(G)\ge\lambda$。

定理 5.1.5 设 S 是算子模糊公式 G 对应的算子模糊子句集，$\lambda\in[0,1]$，则 G 是 λ-恒假当且仅当 S 是 λ-恒假。

证明：由定理 5.1.4 知：公式 G 与一个前束范式等价，不妨设 G 的前束范式为：

$$G=(Q_1x_1)\cdots(Q_nx_n)M(x_1,\cdots,x_n)$$

设 (Q_rx_r) 是从左向右看第一个存在量词，令

$$G_1=(\forall x_1)\cdots(\forall x_{r-1})\cdots(Q_nx_n)M(x_1,\cdots,x_{r-1},f(x_1,\cdots,x_{r-1}),x_{r+1},\cdots,x_n)$$

其中 $f(x_1,\cdots,x_{r-1})$ 是对应于 $(\exists x_r)$ 的 Skolem 函数。

下证：G 是 λ-恒假当且仅当 G_1 是 λ-恒假。

若 G 是 λ-恒假，而 G_1 不是 λ-恒假，则存在解释 D，使得在解释 D 下的赋值 $v(G_1)>\lambda$。

G_1 的解释显然也是 G 的解释，于是

$$v(G_1)$$
$$=\inf_{(x_1,\cdots,x_{r-1})\in D^{r-1}} v((Q_{r+1}x_{r+1})\cdots(Q_nx_n)M(x_1,\cdots,x_{r-1},f(x_1,\cdots,x_{r-1}),x_{r+1},\cdots,x_n)),$$

$$v(G)=\inf_{(x_1,\cdots,x_{r-1})\in D^{r-1}} v((\exists x_r)(Q_{r+1}x_{r+1})\cdots(Q_nx_n)M(x_1,\cdots,x_{r-1},x_r,\cdots,x_n))$$

对任一组 $(x_1^0,\cdots,x_{r-1}^0)\in D^{r-1}$，有

$$v((\exists x_r)(Q_{r+1}x_{r+1})\cdots(Q_nx_n)M(x_1^{\ 0},\cdots,x_{r-1}^{\ 0},x_r,\cdots,x_n))$$

$$=\sup_{x_r\in D} v((Q_{r+1}x_{r+1})\cdots(Q_nx_n)M(x_1,\cdots,x_{r-1},x_r,\cdots,x_n))$$

$$\ge v((Q_{r+1}x_{r+1})\cdots(Q_nx_n)M(x_1{}^0,\cdots,x_{r-1}{}^0,f(x_1{}^0,\cdots,x_{r-1}{}^0),x_{r+1},\cdots,x_n))$$

所以$v(\mathrm{G})\ge v(\mathrm{G}_1)>\lambda$，此与 G 是$\lambda$-恒假矛盾。

反之，若 G_1 是λ-恒假的，因为对 G 的解释D，有

$$v(\mathrm{G})=\inf_{(x_1,\cdots,x_{r-1})\in D^{r-1}}\sup_{x_r} v((Q_{r+1}x_{r+1})\cdots(Q_nx_n)M(x_1,\cdots,x_{r-1},x_r,\cdots,x_n))$$

$$=\sup_{f(x_1,\cdots,x_{r-1})}\inf_{(x_1,\cdots,x_{r-1})\in D^{r-1}} v((Q_{r+1}x_{r+1})\cdots(Q_nx_n)M(x_1,\cdots,x_{r-1},f,x_{r+1},\cdots,x_n))$$

因为 G_1 是λ-恒假的，所以对 G_1 的解释$\widetilde{D}$，有

$$\widetilde{v}(\mathrm{G}_1)$$

$$=\inf_{(x_1,\cdots,x_{r-1})\in D^{r-1}}\widetilde{v}((Q_{r+1}x_{r+1})\cdots(Q_nx_n)M(x_1,\cdots,x_{r-1},f(x_1,\cdots,x_{r-1}),x_{r+1},\cdots,x_n))$$

由于上式对任意解释都成立，所以

$$\sup_{f(x_1,\cdots,x_{r-1})}\widetilde{v}(\mathrm{G}_1)\le\lambda$$

故$v(\mathrm{G})\le\lambda$。亦即，G 是λ-恒假。

设 G 中有 m 个存在量词，令

$\mathrm{G}_0=\mathrm{G}$

$\mathrm{G}_k=$ 将 G_{k-1} 中从左向右看第一个存在量词

用 Skolem 函数代替所得的公式

显然，G_m 是 G 的 Skolem 范式。

同理可证：G_k 是λ-恒假的当且仅当 G_{k-1} 是λ-恒假的。因此 G 是λ-恒假当且仅当 G 的 Skolem 范式是λ-恒假。

若将 Skolem 范式的母式 M 用一个子句集描述，那么这个子句集就是公式 G 的算子模糊子句集S，由上述分析可知：G 是λ-恒假当且仅当S是λ-恒假。

5.2 算子模糊逻辑的λ-归结

定义 5.2.1 称 $\lambda_1 P_{m_i}(t_1,t_2,\cdots,t_{m_i})$与$\sim(\lambda_2 P_{m_i}(t_1,t_2,\cdots,t_{m_i}))$为 λ-互补文字，$\lambda_1\ P_{m_i}(t_1,t_2,\cdots,t_{m_i})$ 和 $\lambda_2\ P_{m_i}(t_1,t_2,\cdots,t_{m_i})$ 或 $\sim(\lambda_1\ P_{m_i}(t_1,t_2,\cdots,t_{m_i}))$ 和 $\sim$

($\lambda_2 P_{m_i}(t_1,t_2,\cdots,t_{m_i})$)为 λ-相似文字，若 λ_1，$\lambda_2 \geq \lambda$ 或 λ_1，$\lambda_2 \leq \lambda$。称 $\lambda^* P_{m_i}(t_1,t_2,\cdots,t_{m_i})$ 是 λ-无关文字，若 $1-\lambda \leq \lambda^* \leq \lambda$ ($\lambda \geq 0.5$)，或 $\lambda \leq \lambda^* \leq 1-\lambda$ ($\lambda \leq 0.5$)，其中 $P_{m_i}(t_1,t_2,\cdots,t_{m_i})$ 为 m_i 元谓词，门槛 $\lambda \in [0,1]$。

定理 5.2.1 设 S 是算子模糊子句集，将 S 中 λ-无关文字删除后所得子句集记为 S^*，则 S 是 λ-恒假当且仅当 S^* 是 1-λ-恒假($\lambda \geq 0.5$)。

证明：当 $\lambda \geq 0.5$ 时，设 $S = G_1 \wedge G_2 \wedge \ldots \wedge G_m$，而 $S^* = G_1^* \wedge G_2^* \wedge \ldots \wedge G_m^*$，不妨设在算子模糊子句集 S 中只有 $\lambda^* P_{m_i}(t_1,t_2,\cdots,t_{m_i})$ 是 λ-无关文字(多个 λ-无关文字时类似可证)，则 $1-\lambda \leq \lambda^* \leq \lambda$，而对任意解释 D 的赋值 v，$v(\lambda^* P_{m_i}(t_1,t_2,\cdots,t_{m_i})) = \lambda^*$ 或 $1-\lambda^*$，所以 $1-\lambda \leq 1-\lambda^* \leq \lambda$。

若 S^* 是 1-λ-恒假，则 $v(G_1^* \wedge G_2^* \wedge \cdots \wedge G_m^*) \leq 1-\lambda$，$v(G_i^*) \leq \lambda \leq 1-\lambda (i=1,2,\cdots,m)$，若存在 $j \in \{1,2,\cdots,m\}$，使得 $G_j^* \vee \lambda^* P_{m_i}(t_1,t_2,\cdots,t_{m_i}) \in S$，则有 $v(G_j^* \vee \lambda^* P_{m_i}(t_1,t_2,\cdots,t_{m_i})) \leq \lambda$，此时 S 是 λ-恒假，若 S 中不存在 λ-无关文字，S 就是 S^*，显然 S 是 λ-恒假。

反之，若 S 是 λ-恒假，则对任意解释 D 的赋值 v，$v(G_1 \wedge G_2 \wedge \ldots \wedge G_m) \leq \lambda$，$v(G_k) \leq \lambda (k=1,2,\cdots,m)$，存在 $h \in \{1,2,\cdots,m\}$，使得 $G_h = G_h^* \vee \lambda^* P_{m_i}(t_1,t_2,\cdots,t_{m_i}) \in S$，而 $\lambda^* P_{m_i}(t_1,t_2,\cdots,t_{m_i})$ 是 S 中的 λ-无关文字，且设 G_h^* 中无其它 λ-无关文字，则 $G_h^* \in S^*$，那么 $v(G_h^*) \leq 1-\lambda$，所以 $v(S^*) \leq 1-\lambda$，S^* 是 $1-\lambda$-恒假。

定理 5.2.2 设 S 是算子模糊子句集，将 S 中 λ-无关文字删除后所得子句集为 S^*，S^* 是 1-λ-恒假的充要条件是 S^* 是 λ-恒假的($\lambda \geq 0.5$)。

证明：若 S^* 是 1-λ-恒假的，则存在 $G^* \in S^*$，使得 $v(G^*) \leq 1-\lambda \leq \lambda$，所以 S^* 是 λ-恒假的。

若S^*是λ-恒假的，则存在$G \in S^*$，使得$v(G) \le \lambda$，因为G中没有λ-无关文字，所以$v(G) \le 1-\lambda$，故S^*是$1-\lambda$-恒假的。

定理 5.2.3 设S是算子模糊子句集，将S中λ-无关文字删除后所得子句集为S^*，则S是λ-恒假当且仅当S^*是λ-恒假($\lambda \ge 0.5$)。

由定理 5.2.1 以及定理 5.2.2 容易证明该结论。

记$\widetilde{S}$为这样的子句集：将S^*中形如$\lambda^* P_{m_i}(t_1,t_2,\cdots,t_{m_i})$的算子模糊文字，在$\lambda^* \ge \lambda$时，用$P_{m_i}(t_1,t_2,\cdots,t_{m_i})$代替，在$\lambda^* \le 1\text{-}\lambda$时，用$\sim P_{m_i}(t_1,t_2,\cdots,t_{m_i})$代替；将$S^*$中形如$\sim\lambda^* P_{m_i}(t_1,t_2,\cdots,t_{m_i})$的算子模糊文字，在$\lambda^* \ge \lambda$时，用$\sim P_{m_i}(t_1,t_2,\cdots,t_{m_i})$代替，在$\lambda^* \le 1\text{-}\lambda$时，用$P_{m_i}(t_1,t_2,\cdots,t_{m_i})$代替。显然，$\widetilde{S}$是二值逻辑中的子句集。

定理 5.2.4 在算子模糊逻辑中，算子模糊子句集S^*是λ-恒假，当且仅当子句集$\widetilde{S}$在二值逻辑中恒假。

证明：不妨设$\lambda \ge 0.5$。

(充分性) 若子句集$\widetilde{S}$在二值逻辑中不可满足，则对任意解释D，$v(\widetilde{S})=0$成立。设$\widetilde{S}=\widetilde{G}_1 \wedge \widetilde{G}_2 \wedge \cdots \wedge \widetilde{G}_m$，则

$$v(\widetilde{S}) = v(\widetilde{G}_1 \wedge \widetilde{G}_2 \wedge \cdots \wedge \widetilde{G}_m) = v(\widetilde{G}_1) * v(\widetilde{G}_2) * \cdots * v(\widetilde{G}_m) = 0$$

故$v(\widetilde{G}_i)=0$ ($i \in \{1,2,\cdots,m\}$)。

不妨假定$\widetilde{G}_i$含两个谓词P_1，P_2。

a) 若$\widetilde{G}_i = P_1 \vee P_2$，因为$v(\widetilde{G}_i) = v(P_1 \vee P_2) = v(P_1) \oplus v(P_2) = 0$，所以$v(P_1)=0$，且$v(P_2)=0$。

将$\widetilde{G}_i$恢复成S^*中子句G_i，可设$G_i = \lambda_1 P_1 \vee \lambda_2 P_2$，则有$\lambda_1 \ge \lambda$，$\lambda_2 \ge \lambda$，且

$$\begin{aligned} v(G_i) &= v(\lambda_1 P_1 \vee \lambda_2 P_2) \\ &= v(\lambda_1 P_1) \oplus v(\lambda_2 P_2) \\ &= (1-\lambda_1) \oplus (1-\lambda_2) \\ &\le 1-\lambda < \lambda \end{aligned}$$

b) 若$\widetilde{G}_i = \sim P_1 \vee P_2$，因为$v(\widetilde{G}_i) = v(\sim P_1 \vee P_2) = v(\sim P_1) \oplus v(P_2) = 0$，所以$v(P_1) = 1$，且$v(P_2) = 0$。

将$\widetilde{G}_i$恢复成S^*中子句G_i，可设$G_i = \lambda_1 P_1 \vee \lambda_2 P_2$，则有$\lambda_1 \le 1-\lambda$，$\lambda_2 \ge \lambda$，若

$$\begin{aligned} v(G_i) &= v(\lambda_1 P_1 \vee \lambda_2 P_2) \\ &= v(\lambda_1 P_1) \oplus v(\lambda_2 P_2) \\ &= \lambda_1 \oplus (1-\lambda_2) \\ &\ge \lambda \end{aligned}$$

则$1-\lambda_2 \ge \lambda$，故$\lambda \le \lambda_2 \le 1-\lambda$，从而$\lambda \le 0.5$，矛盾，所以此时$v(G_i) \le \lambda$。

类似可证在$\widetilde{G}_i = P_1 \vee \sim P_2$时，$v(G_i) \le \lambda$成立。

c) 若$\widetilde{G}_i = \sim P_1 \vee \sim P_2$，因为$v(\widetilde{G}_i) = v(\sim P_1 \vee \sim P_2) = v(\sim P_1) \oplus v(\sim P_2) = 0$，所以$v(P_1) = 1$，且$v(P_2) = 1$。

将$\widetilde{G}_i$恢复成S^*中子句G_i，可设$G_i = \lambda_1 P_1 \vee \lambda_2 P_2$，则有$\lambda_1 \le 1-\lambda$，$\lambda_2 \le 1-\lambda$，且

$$\begin{aligned} v(G_i) &= v(\lambda_1 P_1 \vee \lambda_2 P_2) \\ &= v(\lambda_1 P_1) \oplus v(\lambda_2 P_2) \\ &= \lambda_1 \oplus \lambda_2 \\ &\le 1-\lambda \le \lambda \end{aligned}$$

即对任意解释D，存在S^*中子句G_i，使得$v(G_i) \le \lambda$。

设$S^* = G_1 \wedge G_2 \wedge \cdots \wedge G_m$，其中$G_i$为将$\widetilde{G}_i (i \in \{1,2,\cdots,m\})$恢复成的$S^*$中子

句，则有

$$\begin{aligned} v(S^*) &= v(G_1 \wedge G_2 \wedge \cdots \wedge G_m) \\ &= v(G_1) * v(G_2) * \cdots * v(G_m) \\ &\le \lambda \end{aligned}$$

所以S^*是λ-恒假。

(必要性) 若S^*是λ-恒假，则对任意解释D，$v(S^*) \le \lambda$。设$S^* = G_1 \wedge G_2 \wedge \cdots \wedge G_m$，则

$$\begin{aligned} v(S^*) &= v(G_1 \wedge G_2 \wedge \cdots \wedge G_m) \\ &= v(G_1) * v(G_2) * \cdots * v(G_m) \\ &\le \lambda \end{aligned}$$

所以存在$i \in \{1,2,\cdots,m\}$，使得$v(G_i) \le \lambda$。不妨设$G_i = \lambda_1 P_1 \vee \lambda_2 P_2$，其中$P_1$，$P_2$为谓词，则$v(\lambda_1 P_1) \le \lambda$，且$v(\lambda_2 P_2) \le \lambda$。

(1) 当$\lambda_1 \ge \lambda$，$\lambda_2 \ge \lambda$时，$\widetilde{G}_i = P_1 \vee P_2$。若$v(\widetilde{G}_i) = 1$，则$v(P_1) = 1$，且$v(P_2) = 1$，而$v(G_i) = v(\lambda_1 P_1 \vee \lambda_2 P_2) = v(\lambda_1 P_1) \oplus v(\lambda_2 P_2) = \lambda_1 \oplus \lambda_2 \ge \lambda$，此与$v(G_i) \le \lambda$矛盾，故$v(\widetilde{G}_i) = 0$。

(2) 当$\lambda_1 \ge \lambda$，$\lambda_2 \le \lambda$时，$\widetilde{G}_i = P_1 \vee \sim P_2$。若$v(\widetilde{G}_i) = 1$，则$v(P_1) = 1$，且$v(P_2) = 0$，而$v(G_i) = v(\lambda_1 P_1 \vee \lambda_2 P_2) = v(\lambda_1 P_1) \oplus v(\lambda_2 P_2) = \lambda_1 \oplus (1 - \lambda_2) \ge \lambda$，此与$v(G_i) \le \lambda$矛盾，故$v(\widetilde{G}_i) = 0$。

同理可证当$\lambda_1 \le \lambda$，$\lambda_2 \ge \lambda$时，$v(\widetilde{G}_i) = 0$ 成立。

(3) 当$\lambda_1 \le \lambda$，$\lambda_2 \le \lambda$时，$\widetilde{G}_i = \sim P_1 \vee \sim P_2$。若$v(\widetilde{G}_i) = 1$，则$v(P_1) = 0$，且$v(P_2) = 0$，而$v(G_i) = v(\lambda_1 P_1 \vee \lambda_2 P_2) = v(\lambda_1 P_1) \oplus v(\lambda_2 P_2) = (1 - \lambda_1) \oplus (1 - \lambda_2) \ge 1 - \lambda$

而G_i是S^*中子句，没有λ-无关文字，所以$v(G_i)\ge\lambda$，矛盾，故$v(\widetilde{G}_i)=0$成立。

而　　$\widetilde{S}=\widetilde{G}_1\wedge\widetilde{G}_2\wedge\cdots\wedge\widetilde{G}_m$，

所以　　$v(\widetilde{S})=v(\widetilde{G}_1\wedge\widetilde{G}_2\wedge\cdots\wedge\widetilde{G}_m)=v(\widetilde{G}_1)*v(\widetilde{G}_2)*\cdots*v(\widetilde{G}_m)=0$

故子句集$\widetilde{S}$在一阶逻辑中不可满足。

定理 5.2.5 在算子模糊逻辑中，子句集S是λ-恒假当且仅当子句集$\widetilde{S}$在一阶逻辑中不可满足。

证明：由于S是λ-恒假当且仅当S^*是1-λ-恒假($\lambda\ge 0.5$)，而S^*是λ-恒假当且仅当子句集$\widetilde{S}$在一阶逻辑中不可满足，所以由定理 5.2.2 有结论成立。

定义 5.2.2 设G_1,G_2是两个无公共变量的算子模糊子句，若$G_1=\lambda_1 P_1\vee G_1*$，$G_2=\sim(\lambda_2 P_2)\vee G_2*$，且$(\lambda_1 P_1)^\theta$与$(\sim(\lambda_2 P_2))^\theta$是λ-互补文字，其中$\theta$是最一般合一(mgu)，则称子句$((G_1*)^\theta-S_1)\vee((G_2*)^\theta-S_2)$为$G_1$和$G_2$的λ-归结式，记为$R_\lambda{}^\theta(G_1,G_2)$。其中$S_1=\{\lambda^* P|\lambda^* P$是$G_1$中的算子模糊文字，且$(\lambda^* P)^\theta$与$(\lambda_1 P_1)^\theta$是λ-相似文字$\}$，$S_2=\{\sim(\lambda^* P)|\sim(\lambda^* P)$是$G_2$中的算子模糊文字，且$\sim(\lambda^* P)^\theta$与$\sim(\lambda_2 P_2)^\theta$是λ-相似文字$\}$。

定理 5.2.6 设G_1,G_2是两个算子模糊子句，若G_1,G_2是λ-恒真，则$R_\lambda{}^\theta(G_1,G_2)$是λ-恒真，其中$\theta$是$G_1$与$G_2$的最一般合一。

证明：设$G_1=\lambda_1 P_1\vee G_1*$，$G_2=\sim(\lambda_2 P_2)\vee G_2*$，$(\lambda_1 P_1)^\theta$与$(\sim(\lambda_2 P_2))^\theta$是λ-互补文字，且设$G_1*$与$G_2*$中不含归结文字的$\lambda$-相似文字，则$R_\lambda{}^\theta(G_1,G_2)=(G_1*)^\theta\vee(G_2*)^\theta$。

若G_1，G_2是λ-恒真，则对任意解释D，赋值$v(G_1)\ge\lambda$，$v(G_2)\ge\lambda$。显然$v(G_1{}^\theta)\ge\lambda$，且$v(G_2{}^\theta)\ge\lambda$，即$v((\lambda_1 P_1)^\theta)\oplus v((G_1*)^\theta)\ge\lambda$与$v((\sim(\lambda_2 P_2))^\theta)\oplus v((G_2*)^\theta)\ge\lambda$成立。若$v((\lambda_1 P_1)^\theta)\ge\lambda$，则$v((\sim(\lambda_2 P_2))^\theta)\le\lambda$，故有

$v((G_2{*})^{\theta})\geq\lambda$，所以 $v((G_1{*})^{\theta}\vee(G_2{*})^{\theta})\geq\lambda$。若 $v((\lambda_1 P_1)^{\theta})\leq\lambda$，则有 $v((G_1{*})^{\theta})\geq\lambda$，所以 $v((G_1{*})^{\theta}\vee(G_2{*})^{\theta})\geq\lambda$ 成立，故 $R_\lambda{}^{\theta}(G_1, G_2)$ 是λ-恒真。

定义 5.2.3 称由λ-无关文字组成的算子模糊子句为λ-空子句，记为λ-□。

定义 5.2.4 设 S 是算子模糊子句集，称有限算子模糊子句列 G_1，G_2，…，G_k 为从 S 到 G_k 的λ-归结演绎，若 $G_i\in S$ $(i\leq k)$或 $G_i=R_\lambda{}^{\theta_i}(G_j, G_h)$ $(j, h\leq i)$。

定义 5.2.5 设 G 是算子模糊子句，若 G 中不含自由变元，则称 G 为算子模糊基子句，全部由算子模糊基子句构成的集合称为算子模糊基子句集。

显然，算子模糊基子句是算子命题逻辑中的子句，由定理 4.2.3 知，算子模糊基子句集 S 是λ-恒假的当且仅当存在从 S 推出□或λ-□的λ-归结演绎。

设 S 是算子模糊子句集，将 S 中λ-无关文字删除后所得子句集为 S^*，$\widetilde{S}$ 是按上述方法由 S 生成的一阶逻辑中的子句集。由一阶逻辑中的 H 域、基例等概念可定义算子模糊子句集 S^* 的相应概念，办法是将 $\widetilde{S}$ 中的 H 域定义为算子模糊子句集 S^* 的 H 域，$\widetilde{S}$ 中的基例加上相应的算子变成算子模糊逻辑中的算子模糊子句，从而定义算子模糊子句集 S^* 中算子模糊子句的基例。

定理 5.2.7(提升引理)若 G_1'，G_2' 分别是算子模糊子句 G_1，G_2 的基例，G' 是 G_1' 与 G_2' 的λ－归结式，则存在 G_1 与 G_2 的一个λ－归结式 G，使得 G' 是 G 的基例。

证明：由于在算子模糊逻辑中，子句集 S 的λ－恒假的判定，可转化为相应子句集 S^* 的λ-恒假的判定，所以可假定 G_1 与 G_2 中没有λ-无关文字。

如果 G_1'，G_2' 分别是算子模糊子句 G_1，G_2 的基例，G' 是 G_1' 与 G_2' 的λ-归结式。将 G_1'，G_2' 与 G' 中的形如 $\lambda^* P$ 的算子模糊文字在 $\lambda^*\geq\lambda$ 时，替换为 P，否则替换为 $\sim P$；而将形如 $\sim(\lambda^* P)$ 的算子模糊文字在 $\lambda^*\geq\lambda$ 时，替换为 $\sim P$，否则替换为 P。将替换后的子句集分别记为 $\widetilde{G}_1'$，$\widetilde{G}_2'$ 与 $\widetilde{G}'$，则 $\widetilde{G}'$

是$\widetilde{G}_1'$ 与$\widetilde{G}_2'$ 的二元归结式。由一阶逻辑中的提升引理知，存在一阶逻辑中的子句$\widetilde{G}_1$，$\widetilde{G}_2$与$\widetilde{G}$，使得$\widetilde{G}$是$\widetilde{G}_1$与$\widetilde{G}_2$的归结式，且$\widetilde{G}'$ 是$\widetilde{G}$的例。将$\widetilde{G}_1$，$\widetilde{G}_2$，$\widetilde{G}$以及$\widetilde{G}'$ 分别还原成算子模糊逻辑中的字句G_1，G_2，G以及G'，则G是G_1与G_2的$\lambda-$归结式，且G' 是G的基例。

定理 5.2.8 在算子模糊逻辑中，子句集S是 λ-恒假的当且仅当存在S的一个有限 λ-恒假的基例集S'。

证明：假定子句集S中没有 λ-无关文字，将S转换成一阶逻辑中的子句集$\widetilde{S}$，则由定理 5.2.5 知，S是 λ-恒假的当且仅当子句集$\widetilde{S}$在二值逻辑中不可满足，而$\widetilde{S}$在二值逻辑中不可满足当且仅当存在$\widetilde{S}$的一个有限不可满足的基例集$\widetilde{S}'$。在基例集$\widetilde{S}'$中加入相应算子构成算子模糊逻辑中子句集S'，则子句集S是 λ-恒假的当且仅当存在S的一个有限 λ-恒假的基例集S'。

定理 5.2.9 在算子模糊逻辑中，若子句集S是 λ-恒假的，则存在从S推出□或 λ-□的 λ-归结演绎。

证明：在算子模糊逻辑中，若子句集S是$\lambda-$恒假的，则存在S的一个有限 λ-恒假的基例集S'。根据定理 4.2.3，存在从S'推出□或 λ-□的 λ-归结演绎D'，根据提升引理，可将D' 提升成一个从S推出□或 λ-□的 λ-归结演绎。

5.3 算子模糊逻辑的 Petri 网模型

由定理 5.2.5 知，算子模糊逻辑系统中算子模糊子句集S的 λ-恒假性的判定可转化为对一阶逻辑中子句集$\widetilde{S}$的不可满足性的判定。而在 3.3 节中，讨论了一阶 Horn 子句集的删除归结原理，从而对那些能转换为一阶 Horn 子句集的算子模糊子句集是能进行 λ-恒假性的判定。对于不能转化为一阶 Horn 子句集的算子模糊子句集，该方法就失效了。这一节，我们试图直接从算子模糊公式的 Petri 网模型出发，探寻算子模糊逻辑的 Petri 网归结推理方法。

根据定理 5.2.3，设S是算子模糊子句集，将S中 λ-无关文字删除后所得子

句集为S^*，则S是 λ-恒假当且仅当S^*是 λ-恒假($\lambda\geq 0.5$)， 所以今后讨论的算子模糊子句集都不含λ-无关文字。

定理 5.3.1 若$\lambda_1 P_{m_1}(t_{11},t_{12},\cdots,t_{1m_1})$与$\lambda_2 P_{m_2}(t_{21},t_{22},\cdots,t_{2m_2})$是算子模糊文字，对于算子模糊子句集中的子句$\sim\lambda_1 P_{m_1}(t_{11},t_{12},\cdots,t_{1m_1})\vee\sim\lambda_2 P_{m_2}(t_{21},t_{22},\cdots,t_{2m_2})$，$\sim\lambda_1 P_{m_1}(t_{11},t_{12},\cdots,t_{1m_1})\vee\lambda_2 P_{m_2}(t_{21},t_{22},\cdots,t_{2m_2})$，$\sim(\sim\lambda_1 P_{m_1}(t_{11},t_{12},\cdots,t_{1m_1}))$，则下列结论成立：

(1) $\sim\lambda_1 P_{m_1}(t_{11},t_{12},\cdots,t_{1m_1})\vee\sim\lambda_2 P_{m_2}(t_{21},t_{22},\cdots,t_{2m_2})$

$=\sim(\lambda_1 P_{m_1}(t_{11},t_{12},\cdots,t_{1m_1})\wedge\lambda_2 P_{m_2}(t_{21},t_{22},\cdots,t_{2m_2}))$;

(2) $\sim\lambda_1 P_{m_1}(t_{11},t_{12},\cdots,t_{1m_1})\vee\lambda_2 P_{m_2}(t_{21},t_{22},\cdots,t_{2m_2})$

$=\lambda_1 P_{m_1}(t_{11},t_{12},\cdots,t_{1m_1})\to\lambda_2 P_{m_2}(t_{21},t_{22},\cdots,t_{2m_2})$;

(3) $\sim(\sim\lambda_1 P_{m_1}(t_{11},t_{12},\cdots,t_{1m_1}))=\lambda_1 P_{m_1}(t_{11},t_{12},\cdots,t_{1m_1})$。

证明：设$x_1,x_2,\cdots,x_n$是出现在$\lambda_1 P_{m_1}(t_{11},t_{12},\cdots,t_{1m_1})$与$\lambda_2 P_{m_2}(t_{21},t_{22},\cdots,t_{2m_2})$中的所有自由变元，受全称量词约束，对任意解释$D$的赋值$v$，

(1) $\inf\limits_{(x_1,x_2,\cdots x_n)\in D^n} v(\sim\lambda_1 P_{m_1}(t_{11},t_{12},\cdots,t_{1m_1})\vee\sim\lambda_2 P_{m_2}(t_{21},t_{22},\cdots,t_{2m_2}))$

$=\inf\limits_{(x_1,x_2,\cdots x_n)\in D^n}(v(\sim\lambda_1 P_{m_1}(t_{11},t_{12},\cdots,t_{1m_1}))\oplus v(\sim\lambda_2 P_{m_2}(t_{21},t_{22},\cdots,t_{2m_2})))$

$=\inf\limits_{(x_1,x_2,\cdots x_n)\in D^n}((v(\lambda_1 P_{m_1}(t_{11},t_{12},\cdots,t_{1m_1}))'\oplus(v(\lambda_2 P_{m_2}(t_{21},t_{22},\cdots,t_{2m_2}))')$

$=\inf\limits_{(x_1,x_2,\cdots x_n)\in D^n}(v(\lambda_1 P_{m_1}(t_{11},t_{12},\cdots,t_{1m_1}))*v(\lambda_2 P_{m_2}(t_{21},t_{22},\cdots,t_{2m_2})))'$

$=\inf\limits_{(x_1,x_2,\cdots x_n)\in D^n} v(\sim(\lambda_1 P_{m_1}(t_{11},t_{12},\cdots,t_{1m_1})\wedge\lambda_2 P_{m_2}(t_{21},t_{22},\cdots,t_{2m_2})))$

所以$\sim\lambda_1 P_{m_1}(t_{11},t_{12},\cdots,t_{1m_1})\vee\sim\lambda_2 P_{m_2}(t_{21},t_{22},\cdots,t_{2m_2})$

$=\sim(\lambda_1 P_{m_1}(t_{11},t_{12},\cdots,t_{1m_1})\wedge\lambda_2 P_{m_2}(t_{21},t_{22},\cdots,t_{2m_2}))$;

(2) 由于

$$\inf_{(x_1,x_2,\cdots x_n)\in D^n} v(\lambda_1 P_{m_1}(t_{11},t_{12},\cdots,t_{1m_1}) \to \lambda_2 P_{m_2}(t_{21},t_{22},\cdots,t_{2m_2}))$$

$$= \inf_{(x_1,x_2,\cdots x_n)\in D^n} v(\sim(\lambda_1 P_{m_1}(t_{11},t_{12},\cdots,t_{1m_1}))\vee \lambda_2 P_{m_2}(t_{21},t_{22},\cdots,t_{2m_2}))$$

所以

$$\sim \lambda_1 P_{m_1}(t_{11},t_{12},\cdots,t_{1m_1}) \vee \lambda_2 P_{m_2}(t_{21},t_{22},\cdots,t_{2m_2})$$

$$= \lambda_1 P_{m_1}(t_{11},t_{12},\cdots,t_{1m_1}) \to \lambda_2 P_{m_2}(t_{21},t_{22},\cdots,t_{2m_2});$$

(3) 因为

$$\inf_{(x_1,x_2,\cdots x_n)\in D^n} v(\sim(\sim \lambda_1 P_{m_1}(t_{11},t_{12},\cdots,t_{1m_1})))$$

$$= \inf_{(x_1,x_2,\cdots x_n)\in D^n} v(\lambda_1 P_{m_1}(t_{11},t_{12},\cdots,t_{1m_1})$$

所以$\sim(\sim \lambda_1 P_{m_1}(t_{11},t_{12},\cdots,t_{1m_1})) = \lambda_1 P_{m_1}(t_{11},t_{12},\cdots,t_{1m_1})$。

由定理 5.3.1 知，对算子模糊子句集中的子句可化为如下形式 (不失一般性，以含四个算子模糊文字和一个连接词“→”且连接词“→”前后各含两个算子模糊文字为例)：

(1) $$\begin{aligned}&\lambda_1 P_{m_1}(t_{11},t_{12},\cdots,t_{1m_1}) \wedge \lambda_2 P_{m_2}(t_{21},t_{22},\cdots,t_{2m_2})\\ &\to \lambda_3 P_{m_3}(t_{31},t_{32},\cdots,t_{3m_3}) \wedge \lambda_4 P_{m_4}(t_{41},t_{42},\cdots t_{4m_4});\end{aligned}$$

(2) $$\begin{aligned}&\lambda_1 P_{m_1}(t_{11},t_{12},\cdots,t_{1m_1}) \wedge \lambda_2 P_{m_2}(t_{21},t_{22},\cdots,t_{2m_2})\\ &\to \lambda_3 P_{m_3}(t_{31},t_{32},\cdots,t_{3m_3}) \vee \lambda_4 P_{m_4}(t_{41},t_{42},\cdots t_{4m_4});\end{aligned}$$

(3) $$\begin{aligned}&\lambda_1 P_{m_1}(t_{11},t_{12},\cdots,t_{1m_1}) \vee \lambda_2 P_{m_2}(t_{21},t_{22},\cdots,t_{2m_2})\\ &\to \lambda_3 P_{m_3}(t_{31},t_{32},\cdots,t_{3m_3}) \wedge \lambda_4 P_{m_4}(t_{41},t_{42},\cdots t_{4m_4});\end{aligned}$$

(4) $$\begin{aligned}&\lambda_1 P_{m_1}(t_{11},t_{12},\cdots,t_{1m_1}) \vee \lambda_2 P_{m_2}(t_{21},t_{22},\cdots,t_{2m_2})\\ &\to \lambda_3 P_{m_3}(t_{31},t_{32},\cdots,t_{3m_3}) \vee \lambda_4 P_{m_4}(t_{41},t_{42},\cdots t_{4m_4});\end{aligned}$$

其中，$\lambda_i P_{m_i}(t_{i1},t_{i2},\cdots,t_{im_i})(i=1,2,3,4)$ 为算子模糊文字，且每种形式都受全称量词约束，称以上四种形式的表达式为算子模糊规则型子句。

5.3.1 算子模糊 Horn 子句的 Petri 网模型

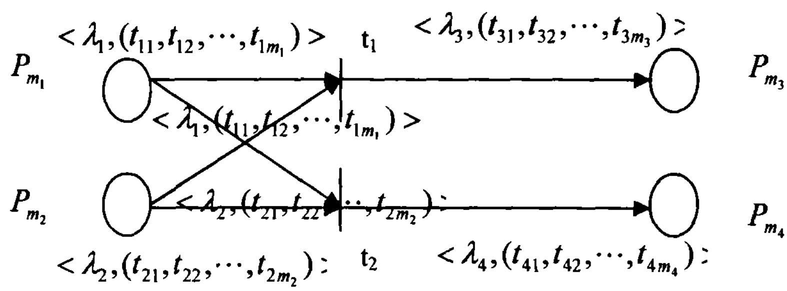
定义 5.3.1 称连接词“→”后最多有一个算子模糊原子的算子模糊规则型子句为算子模糊 Horn 子句。

本节只针对算子模糊 Horn 子句进行讨论。上述给定的四种算子模糊规则型子句中，(2)与(4)不能转化为算子模糊 Horn 子句，其余算子模糊规则型子句的 Petri 网模型如下：

模型一：因为

$$
\begin{aligned}
&\lambda_1 P_{m_1}(t_{11},t_{12},\cdots,t_{1m_1}) \wedge \lambda_2 P_{m_2}(t_{21},t_{22},\cdots,t_{2m_2}) \\
&\rightarrow \lambda_3 P_{m_3}(t_{31},t_{32},\cdots,t_{3m_{31}}) \wedge \lambda_4 P_{m_4}(t_{41},t_{42},\cdots t_{4m_4}) \\
&= (\lambda_1 P_{m_1}(t_{11},t_{12},\cdots,t_{1m_1}) \wedge \lambda_2 P_{m_2}(t_{21},t_{22},\cdots,t_{2m_2}) \rightarrow \lambda_3 P_{m_3}(t_{31},t_{32},\cdots,t_{3m_{31}})) \\
&\wedge (\lambda_1 P_{m_1}(t_{11},t_{12},\cdots,t_{1m_1}) \wedge \lambda_2 P_{m_2}(t_{21},t_{22},\cdots,t_{2m_2}) \rightarrow \lambda_4 P_{m_4}(t_{41},t_{42},\cdots t_{4m_4}))
\end{aligned}
$$

所以规则(1)的 Petri 网模型为：



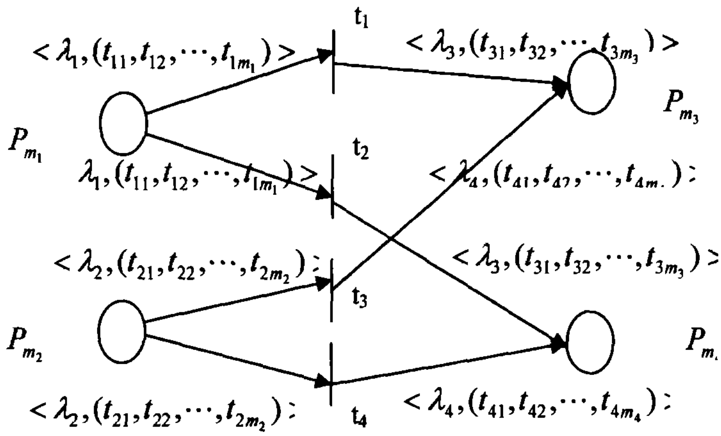
注：该模型用谓词符号 P_{m_i} 表示位置结点，而将算子及谓词的 m_i 元置于与谓词符号 P_{m_i} 相连接的弧上，共同作为符号连接权，这样的表示是为了在推理时易于替换。

模型二：因为

$$
\begin{aligned}
&\lambda_1 P_{m_1}(t_{11},t_{12},\cdots,t_{1m_1}) \vee \lambda_2 P_{m_2}(t_{21},t_{22},\cdots,t_{2m_2}) \\
&\rightarrow \lambda_3 P_{m_3}(t_{31},t_{32},\cdots,t_{3m_{31}}) \wedge \lambda_4 P_{m_4}(t_{41},t_{42},\cdots t_{4m_4}) \\
&= (\lambda_1 P_{m_1}(t_{11},t_{12},\cdots,t_{1m_1}) \rightarrow \lambda_3 P_{m_3}(t_{31},t_{32},\cdots,t_{3m_{31}})) \\
&\wedge (\lambda_2 P_{m_2}(t_{21},t_{22},\cdots,t_{2m_2}) \rightarrow \lambda_3 P_{m_3}(t_{31},t_{32},\cdots,t_{3m_{31}}))
\end{aligned}
$$

$$\wedge(\lambda_1 P_{m_1}(t_{11},t_{12},\cdots,t_{1m_1})\rightarrow\lambda_4 P_{m_4}(t_{41},t_{42},\cdots t_{4m_4}))$$
$$\wedge(\lambda_2 P_{m_2}(t_{21},t_{22},\cdots,t_{2m_2})\rightarrow\lambda_4 P_{m_4}(t_{41},t_{42},\cdots t_{4m_4}))$$

所以(3)的 Petri 网模型为：



从这两个算子模糊 Horn 子句的 Petri 网模型，定义一般算子模糊 Horn 子句集的 Petri 网模型如下：

定义 5.3.2 算子模糊 Horn 子句集 S 的 Petri 网模型是一个五元组 HFPN={P，T，F，W，M}，其中：

P：为位置集合，每个位置表示 S 中的一个谓词符号；

T：为变迁集合，每个变迁对应一个 S 中的算子模糊 Horn 子句；

F：为有向弧集；

W：F$\rightarrow f_s$，W 将弧集 F 映射到定义在项集上的符号集 f_s。任取 $P_{m_i}\in$P，t$\in$N，若(P_{m_i}，t) $\in$F，则 W(P_{m_i}，t) 是算子模糊原子 $\lambda P_{m_i}(t_1,t_2,\cdots,t_{m_i})$ 中的符号串 $<\lambda,(t_1,t_2,\cdots,t_{m_i})>$；

M：P$\rightarrow$N$\cup\{0\}$，M_0 为初始标识。

5.3.2　一般算子模糊子句的 Petri 网模型

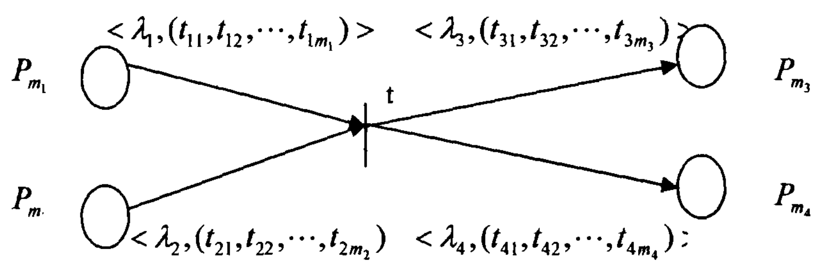
本节讨论一般算子模糊子句集的 Petri 网模型。四个算子模糊规则型子句的 Petri 网模型表示如下：

模型一：同 5.3.1 节中的模型一。

模型二：算子模糊规则型子句

$$\lambda_1 P_{m_1}(t_{11},t_{12},\cdots,t_{1m_1}) \wedge \lambda_2 P_{m_2}(t_{21},t_{22},\cdots,t_{2m_2})$$
$$\rightarrow \lambda_3 P_{m_3}(t_{31},t_{32},\cdots,t_{3m_{31}}) \vee \lambda_{m_4} P_{m_4}(t_{41},t_{42},\cdots t_{4m_4});$$

的 Petri 网摸如下：

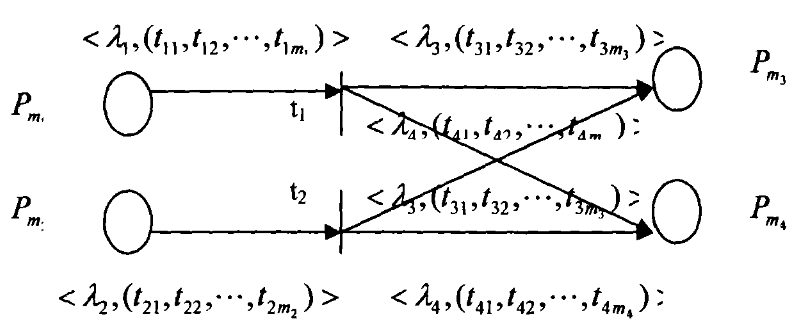


模型三：同 5.3.1 节中的模型三

模型四：因为

$$\lambda_1 P_{m_1}(t_{11},t_{12},\cdots,t_{1m_1}) \vee \lambda_2 P_{m_2}(t_{21},t_{22},\cdots,t_{2m_2})$$
$$\rightarrow \lambda_3 P_{m_3}(t_{31},t_{32},\cdots,t_{3m_{31}}) \vee \lambda_4 P_{m_4}(t_{41},t_{42},\cdots t_{4m_4})$$
$$=(\lambda_1 P_{m_1}(t_{11},t_{12},\cdots,t_{1m_1}) \rightarrow \lambda_3 P_{m_3}(t_{31},t_{32},\cdots,t_{3m_{31}}) \vee \lambda_4 P_{m_4}(t_{41},t_{42},\cdots t_{4m_4}))$$
$$\wedge(\lambda_2 P_{m_2}(t_{21},t_{22},\cdots,t_{2m_2}) \rightarrow \lambda_3 P_{m_3}(t_{31},t_{32},\cdots,t_{3m_{31}}) \vee \lambda_4 P_{m_4}(t_{41},t_{42},\cdots t_{4m_4}))$$

所以算子模糊规则型子句(4) 的 Petri 网模型如下：



定义 5.3.3 一般算子模糊子句集 S 的 Petri 网模型是一个五元组 FPN={P，T，F，W，M}，其中：

P：为位置集合，每个位置表示 S 中的一个谓词符号；

T：为变迁集合，每个变迁对应一个 S 中的规则型算子模糊子句；

F：为有向弧集；

W：F→f_s，W 将弧集 F 映射到定义在项集上的符号集f_s。任取P_{m_i}∈P，t∈N，若(P_{m_i}，t) ∈F，则 W(P_{m_i}，t) 是算子模糊原子$\lambda P_{m_i}(t_1,t_2,\cdots,t_{m_i})$中的符号串$<\lambda,(t_1,t_2,\cdots,t_{m_i})>$；

M：P→N∪{0}，M_0为初始标识。

定义 5.3.2 与定义 5.3.3 的本质区别在于一个变迁对应算子模糊 Horn 子句，一个对应规则型算子模糊子句。

5.4 算子模糊逻辑的 Petri 网推理算法

根据算子模糊子句集的上述两种 Petri 网模型，讨论算子模糊逻辑的两种 Petri 网推理算法。

5.4.1 算子模糊逻辑的 T-不变量推理算法

因为算子模糊 Horn 子句集的 Petri 网模型与一阶逻辑的 Horn 子句集的 Petri 网模型的区别仅仅是连接权发生了改变，即关联矩阵有所不同，如果重新定义关联矩阵，那么可以利用 Horn 子句集的 T-不变量来讨论算子模糊逻辑的 λ-恒假性，从而完成算子模糊 Horn 子句集的 λ-归结推理。

定义 5.4.1 称矩阵 C=[c_{ij}]为 HFPT 模型的关联矩阵，如果 c_{ij} = W(t_i,P_{m_j})-W(P_{m_j},t_i) 。

注：由于 W(t_i,P_{m_j})或 W(P_{m_j},t_i)中含符号串，所以 c_{ij} 只是形式上的记号。

对于算子模糊逻辑中的推理，合一算法是必须的。令 U(P_{m_j})表示位置P_{m_j}中的合一关系集，取 u(P)∈U(P_{m_j}), u(θ) = (w(t_i, P_{m_j})，W(P_{m_j},t_i), θ)表示 W(t_i, P_{m_j})与 W(P_{m_j},t_i) 可合一(指相应谓词P_{m_j}的项$(t_1,t_2,\cdots,t_{m_i})$可替换为相同)，最大合一算子是θ。为了实现归结推理，合一只在互补对(指符号串中相应谓词P_{m_j}的

项$(t_1,t_2,\cdots,t_{m_i})$可替换为相同，且串中算子λ^*大于或等于λ，符号串前一正一负)中进行。由于 HFPT 模型的特殊性，在判定 T-不变量时，与一阶逻辑有所不同，具体判定方法如下：

算法 5.1：C 是 FPT 模型的 m×n 关联矩阵，I_m是单位矩阵。

(1) A：= C；D：= I_m

(2) 重复 i = 1 到 i = n ：

2.1) 对矩阵 A 中第 i 列符号相反，且模糊算子都 $\lambda^* \geq \lambda$(或都 $\lambda^* \leq \lambda$)(即对应文字为 λ-互补文字)，且可以合一的任意两行，在矩阵[D/A]中经合一算法后进行相加运算，并将产生的新行加入[D/A]中。

2.2) 从[D/A]中消除 A 中第 i 列元素在 2.1)中参加运算的行。

A 中模糊算子都小于或等于 λ 所对应的 D 中的行即为关联矩阵 C 的 T-不变量。

定理 5.4.1 算子模糊 Horn 子句集 S 经 λ-归结演绎推出□的充要条件是 S 的 Petri 网模型中存在大于 0 的 T-不变量。

证明：若存在从 S∪G 推出□的 λ-归结演绎，则其激活转移结点的次数构成的向量即为满足结论的 T-不变量。

反之，若 S∪G 的 Petri 网模型中存在大于 0 的 T-不变量 T，由算法 5.1 知，存在从 S 推出□的 λ-归结演绎。

推论 5.4.1 若算子模糊 Horn 子句集 S 是 λ-恒假的，则 S 的 Petri 网模型中存在大于 0 的 T-不变量。

由定理 5.4.1 直接可证。

例 5.4.1[45]如果任何一个上届奥运会金牌得主都“很”可能是这届奥运金牌得主，已知某人是上届奥运银牌得主，证明他“很”可能是这届奥运金牌得主。

令下列谓词：

A(x)：x 是上届奥运金牌得主

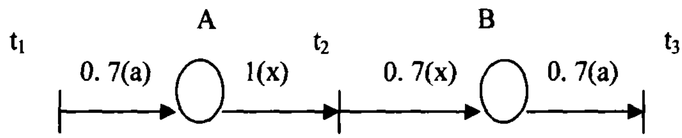
B(x)：x 是这届奥运金牌得主

假定：上届银牌得主记为 0.7A(x)，亦即是“很”可能的金牌得主。那么上述问题转化为问题：

已知：$\forall x(A(x)\rightarrow 0.7B(x))$, 0.7A(a),证明：0.7B(a)。

写成子句集：S=(0A(x)∨0.7B(x), 0.7A(a), 0.3B(a))，取门槛 λ=0.69，其 FPT

网如下：



$$\text{关联矩阵 C}\quad \begin{array}{c} \\ t_1 \\ t_2 \\ t_3 \end{array} \begin{array}{c} \begin{array}{cc} \quad A & \qquad B \end{array} \\ \begin{bmatrix} 0.7(a) & 0 \\ -1(x) & 0.7(x) \\ 0 & -0.7(a) \end{bmatrix} \end{array}$$

，T-不变量算法如下：

1)令 A：= C；D：= I_3

2)

$$[D/A] = \begin{array}{c} \\ t_1 \\ t_2 \\ t_3 \end{array} \begin{array}{c} \begin{array}{cc} \qquad\qquad\quad A & \qquad B \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 0.7(a) & 0 \\ 0 & 1 & 0 & -1(x) & 0.7(x) \\ 0 & 0 & 1 & 0 & -0.7(a) \end{bmatrix} \end{array}$$

$$\rightarrow \quad \begin{array}{c} \\ t_1+t_2(a/x) \\ t_3 \end{array} \begin{array}{c} \begin{array}{cc} \qquad\qquad\quad A & \qquad B \end{array} \\ \begin{bmatrix} 1 & 1 & 0 & -0.3(a) & 0.7(a) \\ 0 & 0 & 1 & 0 & -0.7(a) \end{bmatrix} \end{array}$$

$$\rightarrow \quad \begin{array}{c} \\ t_1+t_2(a/x)+t_3 \end{array} \begin{array}{c} \begin{array}{cc} \qquad\qquad\quad A & \qquad B \end{array} \\ \begin{bmatrix} 1 & 1 & 1 & -0.3(a) & 0(a) \end{bmatrix} \end{array}$$

由于 0.3 与 0 都小于门槛 0.69，所以 X=(1，1，1)为 T-不变量，且由于 X＞0，所以 0.7B(a)得证。

5.4.2　算子模糊逻辑的删除推理算法

这一节，讨论算子模糊逻辑的删除推理算法。

算法 5.2

(1)　设 P_{m_i} 是一个位置结点，且 $\left|{}^{*}P_{m_i}\right|$ = k (k≥1)，$\left|P_{m_i}{}^{*}\right|$= m (m≥1)，不妨设 ${}^{*}P_{m_i}$ = {t_1，t_2，…，t_k}，$P_{m_i}{}^{*}$ = { t_1'，t_2'，…，t_m' }，若 W(t_i, P_{m_i})与 W (P_{m_i}, t_j') 存在一般化合一算子 θ 可

合一，且λ_i与λ_j'同大于等于λ，或同小于等于$1-\lambda$，其中λ_i与λ_j'分别为$W(t_i,P_{m_i})$与$W(P_{m_i},t_j')$中的模糊算子，那么合并转移结点t_i与t_j'为接点t_{ij}，同时删除结点t_i与t_j'，弧(t_i,P_{m_i})与(P_{m_i},t_j')，若此时位置P_{m_i}的前集与后集为空，这时删除该位置结点P_{m_i}，然后对与P_{m_i}连接的所有权实施替换θ，最后将一个标识放入位置$(t_{ij})^*$；

(2) 如果 $t^*\cap {}^*t\neq\phi$(空集)，且任取$P_{m_i}\in t^*\cap {}^*t$，连接结点P_{m_i}的弧上的权同大于等于λ，或同小于等于 $1-\lambda$，则删除转移结点 t 及其所有的弧连接。

(3) 如果$\left|{}^*P_{m_i}\right|=\left|P_{m_i}{}^*\right|=0$，则删除位置接点$P_{m_i}$。

(4) 若$({}^*t_1\cup t_1{}^*)^{\theta}\subset({}^*t_2\cup t_2{}^*)$，则删除转移结点 t_2 及其所有的弧连接。

定理 5.4.2 设 FPN′ 是由算子模糊子句集 S 的 Petri 网模型 FPN 经算法 5.2 作用后得的 Petri 网模型，若 FPN′ 表示的算子模糊子句集是λ-恒假的，则 FPN 表示的算子模糊子句集是λ-恒假。

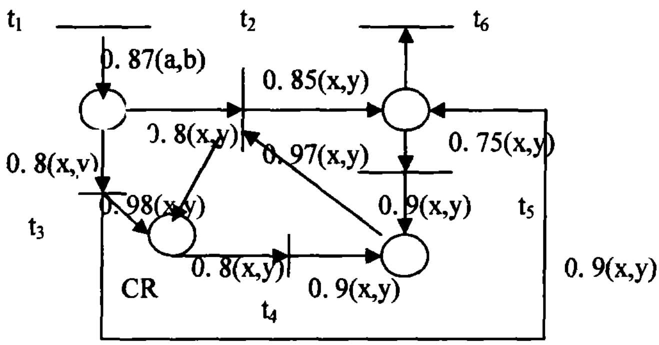
证明：若 FPN′ 表示的算子模糊子句集是λ-恒假，而 FPN 表示的算子模糊子句集是λ-可满足的，则存在赋值v，使得 FPN 中每个子句在这个赋值之下值大于或等于λ。而算法 5.2 作用一次，要么删除λ-恒真式，要么进行一次λ-归结，λ-归结式在 FPN′ 中，所以 FPN′ 中在赋值v之下是λ-可满足的，这与 FPN′ 表示的算子模糊子句集是λ-恒假矛盾，所以 FPN 表示的算子模糊子句集是λ-恒假。

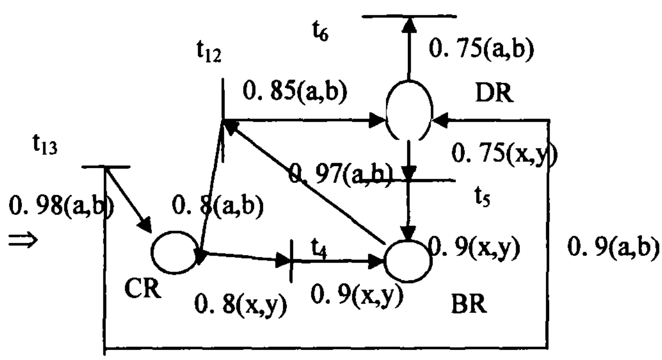
定理 5.4.3 若算子模糊子句集 S 是λ-恒假的，则 S 的一般型 Petri 网模型经删除算法 5.2 后可得空结点或表示λ-□的 Petri 网。

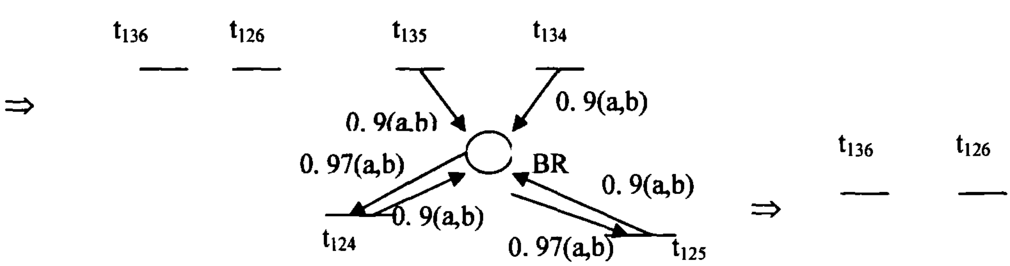
证明：算子模糊子句集 S 是λ-恒假的，则由定理 5.2.5 知：存在从S推出□或λ-□的λ-归结演绎，即 S 的一般型 Petri 网模型经删除算法 5.2 后可得空结点或表示λ-□的 Petri 网。

例 5.4.2 设 S = {0.8A (x,y) →0.9DR(x,y)∨0.98CR (x,y)， 0.75DR(x,y) ∨

0.8 CR(x,y) → 0.9BR(x,y)， 0.97 BR(x,y) ∧ 0.85A (x,y) → 0.85DR(x,y) ∨ 0.8 CR(x,y)，0.87A(a,b)，~0.75DR(a,b)}，取 $\lambda=0.7$，证明 S 为 0.7-不可满足。







由于得到空转移接点 t_{136} 与 t_{126}， 所以 S 为 0.7-不可满足。

第6章 格值命题逻辑系统中公式的 Petri 网模型及推理算法

子句形式的归结是基于归结的自动推理方法中常用的一个方法，该方法要求在归结前首先将公式转化为范式。基于经典逻辑和模糊逻辑系统的关于合取范式和析取范式的研究工作有很多，其上的工作大多是基于如下蕴涵：$G \to H = G' \vee H$。然而，由于格值命题逻辑系统 LP(X)中“→”连结词不能被格值命题逻辑系统中其它逻辑连结词表示，所以基于子句形式的归结推理是无法进行的。但是，Petri 网可以直接表示命题“G→H”，无需将“→”连结词进行转换，并且基于 Petri 网的归结推理算法直观而有效，这为基于格值命题逻辑系统 LP (X)中公式的 Petri 网模型表示及推理提供了可能性。关于 LP (X)上的推理算法，其工作主要有：徐扬[83]在讨论 LP (X)上的α-归结原理时提出了不可分解极简式(IESF)的概念，并在此基础之上给出了 LP (X)上的广义文字、广义合取范式等概念；王伟在文[86]中证明了任一 LP (X)中公式的广义合取范式的存在性；孟丹在文[87]中讨论了化 LP (X)中任一公式为准广义合取范式的方法。本章根据 LP(X)的性质，讨论 LP (X)中公式的 Petri 网模型以及在该模型之下的推理算法。主要研究：

1. 格值命题逻辑系统 LP (X)中公式的规则型范式及其子句集；
2. LP (X)中子句的 Petri 网模型以及公式的 Petri 网表示；
3. 四值非链的格值命题逻辑系统 $LP_4(X)$中公式的 Petri 网推理模型及推理算法；
4. 一般格值命题逻辑系统 LP(X)中公式的 Petri 网推理模型及推理算法。

6.1 格值命题逻辑系统 LP(X)中公式的规则型范式及其子句集

前两章，我们分别讨论了算子命题逻辑系统与算子模糊逻辑系统的 Petri 网推理算法。这一章，我们将基于 Petri 网的推理方法应用到格值命题逻辑系统的推理中。

定义 6.1.1 设 A→B 是格值命题逻辑系统 LP(X)的公式，称 A 是蕴涵算子

“→”的前项，B 是蕴涵算子“→”的后项。

由于 Petri 网不善于表达“∨”与“′”运算，故下面的步骤是根据格值命题逻辑系统中公式的性质，把公式化为统一的形式，以便 Petri 网的表示。

步骤一 删除算子“′”

(1) 由性质 $F\to G=G'\to F'$ 除去“→”两边的算子“′”；

(2) 根据 $F'=F\to O$ 除去剩余的算子“′”。

这样公式中只含有算子{∨，∧，→}。

步骤二 删除算子“∨”

(1) 由性质$(F\vee G)\to H=(F\to H)\wedge(G\to H)$除去蕴涵前项中的算子“∨”；

(2) 根据 $F\vee G=(F\to G)\to G$ 除去剩余的算子“∨”。

这样公式中只含有算子{∧，→}。

步骤三 删除部分算子“∧”

由性质 $F\to(G\wedge H)=(F\to G)\wedge(F\to H)$ 除去蕴涵后项中的算子“∧”。此时，公式中的算子“∧”要么是以合取式如 $F\wedge G$ 的形式单独出现，要么出现在蕴涵的前项。

为什么要保留蕴涵前项中的算子呢？因为形如$(F\wedge G)\to H$ 的公式在 Petri 网中很容易表示。

经过上述三个步骤，一个格值命题逻辑系统 LP(X)中的公式 G 具有较标准的形式：

$$G=G_1\wedge G_2\wedge\cdots\wedge G_n$$

称其为公式 G 的规则型范式，其中 $G_i(i=1,2,\ldots,n)$只含连接词“→”或“∧”，且连接词“∧”只出现在蕴涵的前项中。

定义 6.1.2 称只含连接词“→”或连接词“∧”，且连接词“∧”只出现在连接词“→”的前项的格值命题公式为规则型子句。

定理 6.1.1 每一个格值命题逻辑系统 LP(X)中的公式 G 都可以化为规则型子句的合取。即存在规则型子句 $G_i(i=1,2,\ldots,n)$，使得 $G=G_1\wedge G_2\wedge\cdots\wedge G_n$。

证明：直接由步骤一至步骤三可证明结论。

定义 6.1.3 若 $G\in LP(X)$，且 $G=G_1\wedge G_2\wedge\cdots\wedge G_n$，$G_i(i=1,2,\ldots,n)$为规则型子句，称集合$\{G_1, G_2, \cdots, G_n\}$为公式 G 的子句集，记为 S_G。

定义 6.1.4 设 G 为规则型子句，如果删除 G 中的任何命题常元、命题变元、任何命题常元或命题变元的合取项以及相应的蕴涵项得到的新的规则型子句

G^*都不与 G 等值，称 G 为极简规则型子句，简记为 ESRF(Extra Simple Ruled Formula)。

以下讨论的公式 G 的子句集都是指由极简规则型子句构成的集合。

定理 6.1.2 设 G 是格值命题逻辑系统 LP(X)中的一个公式，S_G是公式 G 的子句集，若对每一赋值v，存在 $G_i \in S_G$，使得v $(G_i) \le \alpha$，则 G 是α-恒假的。

证明：设 $G=G_1 \wedge G_2 \wedge \cdots \wedge G_n$，则$S_G=\{G_1, G_2, \ldots, G_n\}$，其中$G_i(i=1,2,\ldots,n)$为极简规则型子句。

若对每一赋值v，存在 $G_i \in S_G$，使得$v(G_i) \le \alpha$。又

$$v(G)=v(G_1 \wedge G_2 \wedge \cdots \wedge G_n)= v(G_1) \wedge v(G_2) \wedge \cdots \wedge v(G_n) \le \alpha$$

所以 G 是α-恒假的。

6.2 极简规则型子句的 Petri 网模型以及任意公式的 Petri 网模型

下面着手解决极简规则型子句的 Petri 网模型以及任意公式的 Petri 网模型。在此之前，先给出几个定义。

定义 6.2.1 设 G 是一个极简规则型子句，G 中所含运算符{∧，→}的个数记为$\ell(G)$称为 G 的元数，并称公式 G 为$\ell(G)$-ESRF(Extra Simple Rule Formula 的缩写形式)。

若 G 为$\ell(G)$-ESRF，记 G 中运算符∧的个数为$\ell_1(G)$，运算符→的个数为$\ell_2(G)$，显然$\ell(G)= \ell_1(G)+ \ell_2(G)$。

例 6.2.1 $\ell(x)=0$，$\ell(x \wedge y)=1$ ，$\ell(x \to (y \wedge z \to h))=3$，$\ell_1(x \to (y \wedge z \to h))=1$，$\ell_2(x \to (y \wedge z \to h))=2$，公式中的 x、y、z、h 均为命题常元或变元。

通常把常元公式看成 0- ESRF。

在 Petri 网表示经典命题逻辑公式的模型中[117]，常常用一个位置结点表示一个命题变元或常元，一个转移结点对应一个子句，这里我们仍然采用这样的模型方法。但是，对于元数较大的极简规则型子句，其组合形式较多，结构也相当复杂，不可能一一将它们的 Petri 模型表示出来。我们先对 n- ESRF 就 $n \le 3$ 的情况进行探讨，然后给出这类公式的一般化的模型表示。

6.2.1 n- ESRF($n \le 3$)的 Petri 网表示模型

我们约定：以下出现在公式中的小写字母均表示命题变元或命题常元或常

值公式。

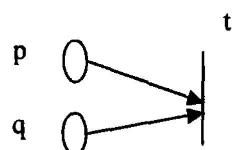
(1) n=0 时

0-ESRF 模式是单个的命题变元或命题常元或常值公式，只需一个位置结点就可以表示。如公式 p 的 Petri 网模型为：

图 6-1 公式 p 的 Petri 网模型

(2) n=1 时

1-ESRF 模式的公式只有两种：$p\wedge q$ 与 $p\rightarrow q$，Petri 网模型分别为：



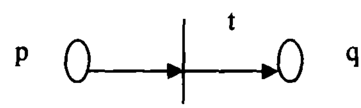


图 6-2 公式 $p\wedge q$ 的 Petri 网模型　图 6-3 公式 $p\rightarrow q$ 的 Petri 网模型

(3) n=2 时

2-ESRF 模式的公式类型：

a) $\ell_1=2$，$\ell_2=0$，即$(\wedge, \wedge)$型：$p\wedge q\wedge r$，公式的 Petri 网模型为：

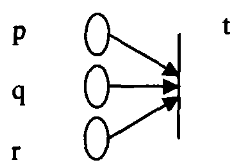


图 6-4 公式 $p\wedge q\wedge r$ 的 Petri 网模型

b) $\ell_1=\ell_2=1$，即$(\wedge, \rightarrow)$型：$(p\wedge q)\rightarrow r$，公式的 Petri 网模型为：

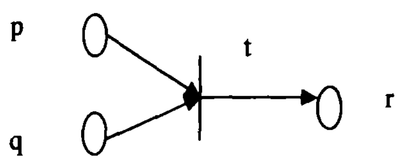


图 6-5 公式$(p\wedge q)\rightarrow r$ 的 Petri 网模型

c) $\ell_1=0$，$\ell_2=2$，即$(\rightarrow, \rightarrow)$型：$(p\rightarrow q)\rightarrow r$，$p\rightarrow(q\rightarrow r)$。

若将公式$(p\rightarrow q)\rightarrow r$ 看成一个规则，则 $p\rightarrow q$ 为前提，r 为结论。按照经典逻辑公式的 Petri 网表示，前提 $p\rightarrow q$ 表示为：

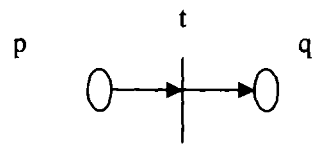


图 6-6 公式 p→q 的 Petri 网模型

而在公式(p→q)→r 中，p→q 作为一个整体出现的，在经典逻辑公式的表示中，不可分的命题即文字用一个位置结点表示。而在格值逻辑公式中，前提 p→q 是不可分的，为了表达公式的这种特性，我们将 p→q 的 Petri 网作为一个广义的位置结点表示为：

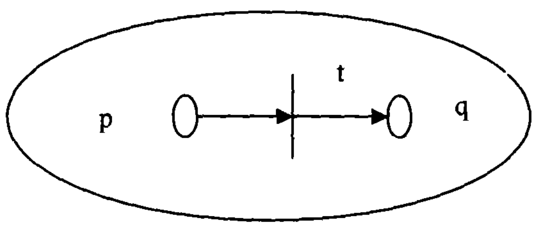
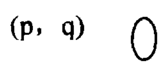


图 6-7 前提 p→q 的广义 Petri 网模型

为了简单起见，将上图简记为：



注：图中用向量(p，q)表示命题 p→q。

因此，公式(p→q)→r 的 Petri 网模型如下：

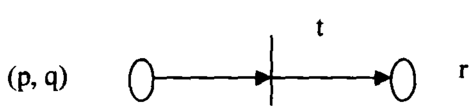


图 6-8 公式(p→q)→r 的 Petri 网模型

同理公式 p→(q→r)的 Petri 网模型为：

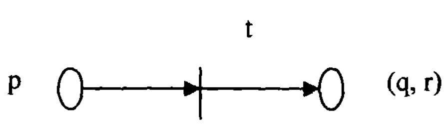


图 6-9 公式 p→(q→r)的 Petri 网模型

(4) n=3 时

3-ESRF 模式的公式类型有：

(a) ℓ_1=3，ℓ_2=0，(∧，∧，∧)型：p∧q∧r∧h，公式的 Petri 网模型为：

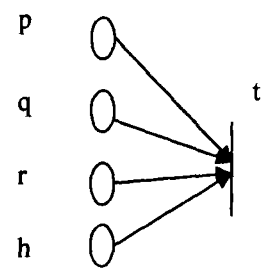


图 6-10 公式 p∧q∧r∧h 的 Petri 网模型

(b) ℓ_1=2，ℓ_2=1，(∧，∧，→)型：(p∧q∧r)→h，公式的 Petri 网模型为：

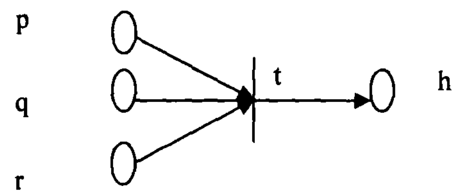


图 6-11 公式(p∧q∧ r)→h 的 Petri 网模型

(c) ℓ_1=1，ℓ_2=2，即(∧， → ， →)型： ①(p∧q)→(r→h)，②(p→(q∧ r))→h，③ p→((q∧ r)→h)，它们的 Petri 网模型分别为：

①(p∧q)→(r→h)

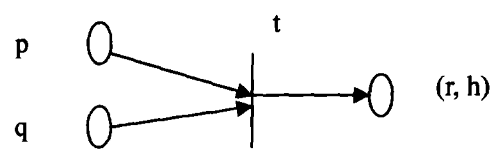


图 6-12 公式(p∧q)→(r→h)的 Petri 网模型

②(p→(q∧ r))→h = ((p→q) ∧(p→r)) →h

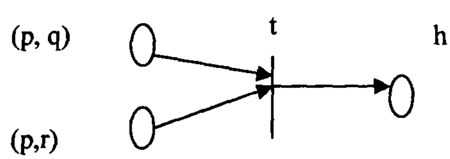


图 6-13 公式(p→(q∧ r))→h 的 Petri 网模型

③$p \to ((q \wedge r) \to h) = (q \wedge r) \to (p \to h)$

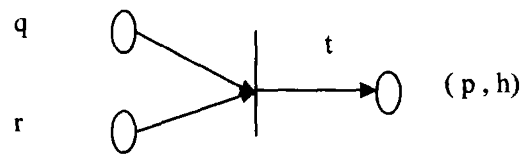


图 6-14 公式 $p \to ((q \wedge r) \to h)$的 Petri 网模型

(d)$\ell_1=0$，$\ell_2=3$，即(→，→，→)型：① $(p \to q) \to (r \to h)$，② $p \to (q \to (r \to h))$，③ $(p \to (q \to r)) \to h$，④ $p \to ((q \to r) \to h)$，⑤ $((p \to q) \to r) \to h$。下面分别就这 5 种公式，讨论它们的 Petri 网推理模型：

① $(p \to q) \to (r \to h)$

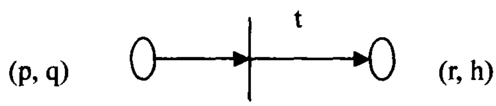


图 6-15 公式 $(p \to q) \to (r \to h)$的 Petri 网模型

② $p \to (q \to (r \to h))$

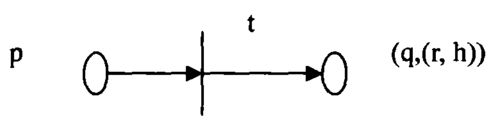


图 6-16 公式 $p \to (q \to (r \to h))$的 Petri 网模型

③ $(p \to (q \to r)) \to h$

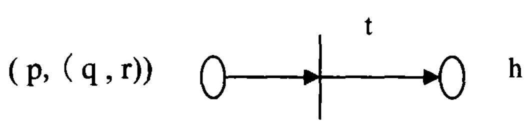


图 6-17 公式$(p \to (q \to r)) \to h$ 的 Petri 网模型

④ $p \to ((q \to r) \to h) = (q \to r) \to (p \to h)$，因此其模型同于①。

⑤ $((p \to q) \to r) \to h$

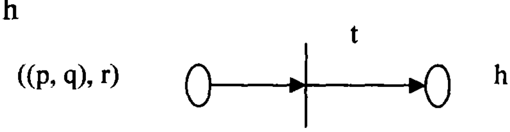


图 6-18 公式$((p \to q) \to r)) \to h$ 的 Petri 网模型

6.2.2 n- ESRF(n≥1)的 Petri 网表示模型

在上节中，我们给出了 n- ESRF(n≤3)的 Petri 网表示模型。由于广义位置结点的引入，使得每一 n- ESRF(n≤3)只对应一个转移结点，这样的表示与经典逻辑公式每一子句的表示是一致的，不同的仅仅是位置结点表示的含义：在经典逻辑公式的表示中，公式中的每一文字与一个位置结点对应；在格值命题逻辑系统中，位置结点不仅可以表示文字，而且可以表示不可分解的命题。

对于 n- ESRF(n>3)的 Petri 网表示模型，随着 n 的增大，这种模式的公式是不断增多，不能将这些公式一一表示出来。但是从 n- ESRF(n≤3)的 Petri 网表示模型，可以给出一般的 n- ESRF(n≥1)的表示方法如下：

定义 6.2.2 n- ESRF(n≥1)的 Petri 网模型是一个三元组(P，T，F)，其中：

$P=\{p_1,p_2,\ldots,p_m\}$为广义位置集合，每一位置表示 LP(X)中的一个不可分命题；

$T=\{t_1,t_2,\ldots,t_n\}$是变迁集合，每一变迁对应一个 ESRF；

$F\subseteq P\times T\cup T\times P$ 为有向弧集。

例 6.2.2 给出公式$((p\vee q)\rightarrow r)\rightarrow h$ 的 Petri 网模型。

由于$((p\vee q)\rightarrow r)\rightarrow h=((p\rightarrow r)\wedge(q\rightarrow r))\rightarrow h$，所以其 Petri 网模型为：

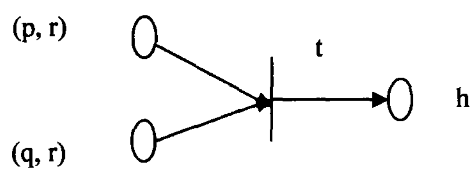


图 6-19 公式$((p\vee q)\rightarrow r)\rightarrow h$ 的 Petri 网模型

例 6.2.3 给出公式$(p\rightarrow q)\rightarrow(r\vee h)$的 Petri 网模型。

在格值命题逻辑中，$r\vee h=(r\rightarrow h)\rightarrow h$，这可以用一个广义结点表示，故公式$(p\rightarrow q)\rightarrow(r\vee h)$在格值命题逻辑中的 Petri 网模型为：

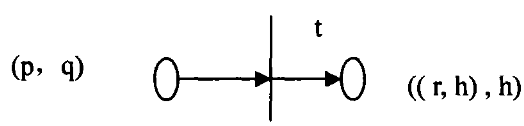


图 6-20 公式$(p\rightarrow q)\rightarrow(r\vee h)$的 Petri 网模型

6.2.3 一般公式的 Petri 网表示模型

对于 LP(X)中的一般公式，由于每一公式对应一个公式的子句集合即 n-ESRF 的集合，故公式的 Petri 网模型可以由 n- ESRF 的 Petri 网模型合成得到。

例 6.2.4 给出公式$((p\wedge q)\to r)\to h$ 的 Petri 网模型

由于$((p\wedge q)\to r)\to h=((p\to r)\vee(q\to r))\to h=((p\to r)\to h)\wedge((q\to r)\to h)$，

转化为$\ell_1=0,\ell_2=2$ 型的两个公式。其 Petri 网模型为：

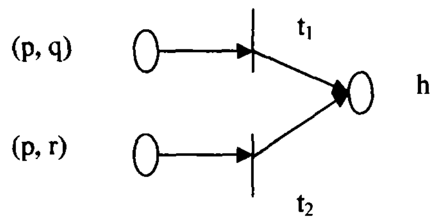


图 6-21 公式$((p\wedge q)\to r)\to h$ 的 Petri 网模型

若要表示一个公式集合的 Petri 网，只需将每一公式的 Petri 网合在一起就可以了。

6.3 四值非链格值命题逻辑系统 LP$_4$(X)的 Petri 网推理模型及算法

定义 6.3.1[87] 四元格 L$_4$(其中的"∧"与"∨"运算如 HASSE 图中所示)上分别定义如下的"′"和"→"运算，则(L$_4$，∨，∧，′，→，O，I)可构成一个四元格蕴涵代数。

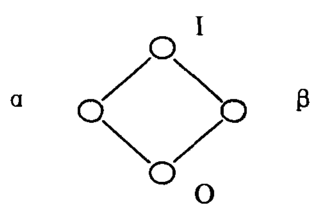


图 6-22 四元格 L$_4$

→	O	α	β	I
O	I	I	I	I
α	β	I	β	I
β	α	α	I	I
I	O	α	β	I

x	O	α	β	I
x′	I	β	α	O

基于 L_4 上的格值命题逻辑系统即为 $LP_4(X)$。

在四值非链格值命题逻辑系统 $LP_4(X)$中，由于其赋值格蕴涵代数为一个布尔代数，系统 $LP_4(X)$就有特殊的性质。对于任意的 p，$q\in LP_4(X)$有 $p\rightarrow q = p'\vee q$，且 $p\vee p' =I$，$p\wedge p' = O$。因此，$LP_4(X)$上的归结推理可在 I 的水平上进行，并且由于 $p\wedge p' = O$，因此其中的归结对的形式是互补文字。所以 $LP_4(X)$上的归结与经典的命题逻辑上的归结推理有许多类似的形式，不同的仅仅是赋值格中的不可比元的存在。对于一般格值命题逻辑系统上的讨论，将在下一节给出。

在 $LP_4(X)$中，由于 $p\rightarrow q = p'\vee q$，所以类似二值逻辑，有相应的合取范式与析取范式以及消去互补对的归结方法。如果保留运算符“→”与“∧”，消除运算符“′”与“∨”，那么也有规则型子句及其子句集以及 n- ESRF 等概念，这里不再重复。

$LP_4(X)$中公式的 Petri 网表示模型同上，下面定义其推理模型。

定义 6.3.2 四值非链格值逻辑系统 $LP_4(X)$的 Petri 网推理模型为五元组(P，T，F，W，M_0)，其中：

P，T，F 与定义 6.2.2 中同；

W：F → {1}为权函数；

M_0：P → N∪{0}为初始标识。

转移结点激活规则：若*t 中每一个位置结点(包含广义)至少含一个标识，则 t 可激活，并将一个标识转移到 t*的每一个结点中。

规定：源结点总是可激活的。

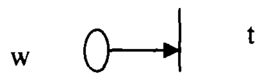
格值命题逻辑系统的 Petri 网推理模型与经典逻辑系统推理模型相比，不同的是广义位置结点的引入。在标识的转移时，如果将广义位置结点表示的命题当作一个广义文字，那么它的 Petri 网推理算法可以用我们给出的删除算法与 T-不变量法，因为此时的 Petri 网模型相当于 Horn 型的 Petri 网模型。以下我们考虑广义位置结点为一个子网，以及推理按照删除算法进行推理的情形。

由于公式中的每一原子并不与位置结点一一对应，在标识的转移过程中涉及子网的展开以及结点的合并等问题。

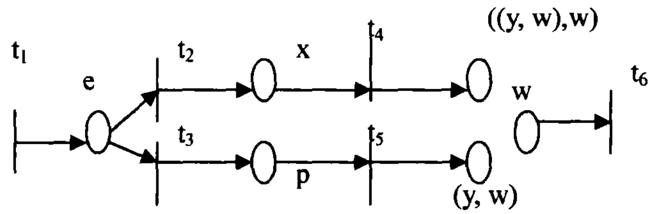
例 6.3.1 判定公式 $G=e\wedge(e\rightarrow x\wedge p)\wedge(x\rightarrow y\vee w)\wedge(p\rightarrow(y\rightarrow w))\wedge w'$ 是否

为 O-假的。

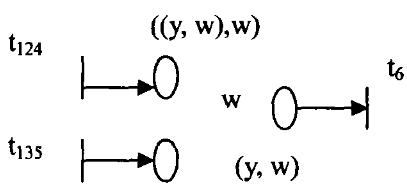
解：因为 $G=e\wedge(e\to x)\wedge(e\to p)\wedge(x\to((y\to w)\to w))\wedge(p\to(y\to w))\wedge(w\to O)$，所以 $S_G=\{e, e\to x, e\to p, x\to((y\to w)\to w), p\to(y\to w), w\to O\}$。如果 $w\to O=w'$ 仍表示为：



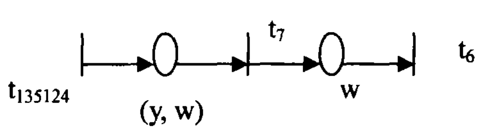
那么，S_G 的 Petri 网模型为：



t_1 是源结点，总是可以激活的，从而激活 t_2 与 t_3，进而 t_4 与 t_5 激活，根据删除算法，可得网络：



这是需要将广义结点((y，w)，w)展开，以及合并结点(y，w)得：



最后，再由删除算法可得空转移结点，即表明公式 G 是 O-假的。

6.4 格值命题逻辑系统中公式的 Petri 网模型及推理算法

在一般的格值命题逻辑系统 LP(X)(除 $LP_4(X)$)，$p\vee p'=I$，$p\wedge p'=O$ 一般不成立，而且在 I 的水平上归结推理意义不大，所以考虑在 α 水平上的归结。

在 LP(X)中，建立了语法推理规则 MP[160]：

$$\frac{\begin{array}{c}(p\to q,\ \alpha_1)\\(p,\alpha_2)\end{array}}{(q,\ \alpha_1\otimes\alpha_2)}$$

其中$\alpha_1\otimes\alpha_2=(\alpha_1\to\alpha_2{}')'$，即$\otimes$是格蕴涵代数L中关于→的伴随运算。

由此推理规则，我们给出LP(X)上MP归结式及(A,α)-归结演绎的定义及其性质。

定义6.4.1 设G，G→H是两个极简规则型子句，称H为G和G→H的MP归结式，记为MP(G，G→H)，求MP归结式的过程称为MP归结。

定义6.4.2[158]设C和G是两个极简规则型子句，R为一个推理规则，R(C,G)为C和G的推理结果。如果对LP(X)的任意赋值v，

$$v(R(C,G))\ \geq v(C)\ \otimes v(G)$$

则称R(C,G)为C和G的逻辑推论。

定理6.4.1 设G和G→H是两个极简规则型子句，则MP(G ,G→H) 是G和G→H的逻辑推论。

证明：因为对任意的g，h∈L，有

$$\begin{aligned}&[g\otimes(g\to h)]\to h\\&=h'\to[g\to(g\to h)']\\&=g\to[h'\to(g\to h)']\\&=g\to[(g\to h)\to h]\\&=g\to g\vee h=I\end{aligned}$$

从而对任意的LP(X)的赋值v，$v(H)\geq v(G)\ \otimes v(G\to H)$，即$v(R(G,G\to H))\geq v(G)\ \otimes v(G\to H)$，故MP(G, G→H) 是G和G→H的逻辑推论。

考虑映射A：LP(X)→L，A是LP(X)上的一个L-型模糊集，记为$A\in F_L(LP(X))$。对任意的p∈LP(X)，$A(p)$称为p的信度，$A(p)\in L$。

定义6.4.3 设S是极简规则型子句集，C是极简规则型子句，$\alpha\in L$，从S到C的(A,α)—归结演绎ϖ是如下形式的有穷序列：

(C_1, α_1)，(C_2, α_2)，…，(C_n, α_n)

其中 $C_n = C$，$\alpha_n \geq \alpha$。对任意的 $i(1 \leq i \leq n)$，$\alpha_i \in L$，且

(1) $C_i \in S$，$\alpha_i = A(C_i)$，或

(2) 存在 j，k<i，使 $C_i = MP(C_j, C_k)$，且 $\alpha_i = \alpha_j \otimes \alpha_k$。

记 ϖ 为 $S \mid\!\!-_{(A,\alpha)} C$，称 n 为 ϖ 的长度记为 $l(\varpi)$。特别地，称从 S 到 O 的 (A,α)-归结演绎为 (A,α)-反驳，记为 $S \mid\!\!-_{(A,\alpha)} O$。

这里的定义 6.4.3 类似于文献[159]中的定义 9.1.8，可视为一种无公理但附加了前提条件 S 和 A 的而对 C 的证明。

定义 6.4.4 设 $A \in F_L(LP(X))$，S 是极简规则型子句集，S 称为 A-可满足的，如果存在 LP(X)的赋值 v，使得对任意的 $C \in S$，$v(C) \geq A(C)$ 成立。否则称 S 为 A-不可满足。

定义 6.4.5 设 $A \in F_L(LP(X))$，S 是极简规则型子句集，C 是极简规则型子句，$\alpha \in L$，称 C 是在 A 下的一个 α-逻辑推论，记为 $S \models_\alpha^A C$，如果对 LP(X) 的任意赋值 v，当 S 是 A-可满足时，有 $v(C) \geq \alpha$。

定理 6.4.2 设 S 为极简规则型子句集，若 $A \in F_L(LP(X))$，$\alpha \in L$，且对任意 $C \in S$，$A(C) \geq \alpha$，则 $S \models_\alpha^A C$。

证明：LP(X)的任意赋值 v，当 v 是 A-满足 S 时，则对任意 $C \in S$，有 $v(C) \geq A(C) \geq \alpha$ 由定义 6.4.5 知，$S \models_\alpha^A C$。

定理 6.4.3(可靠性) 设 $A \in F_L(LP(X))$，S 是极简规则型子句集，C 是极简规则型子句，$\alpha \in L$，如果 $S \mid\!\!-_{(A,\alpha)} C$，则 $S \models_\alpha^A C$。

证明：设 (C_1,α_1), (C_2,α_2), ⋯ , (C_n,α_n)是一个从 S 到 C 的 (A,α)-归结演绎 ϖ，其中 $C_n=C$，$\alpha_n=\alpha$。下面将归纳法用于$l(\varpi)$证明：如果v是A-可满足S，则$v(C)\geq\alpha$。

(1) 若$l(\varpi)=1$，则$C_n=C$，因$A(C_n)=\alpha_n$，而赋值 v是A-满足S 的，所以$v(C)=v(C_n)\geq A(C_n)=\alpha_n=\alpha$。

(2) 设结论对于 $l(\varpi)<n$的演绎ϖ成立，考虑$l(\varpi)=n$。

若存在$i,j\,(i,j<n)$，使得$C_n=\mathrm{MP}(C_i,C_j)$且，$\alpha=\alpha_n=\alpha_i\otimes\alpha_j$，则由归纳假设，$v(C_i)\geq\alpha_i>\alpha, v(C_j)\geq\alpha_j\geq\alpha$，由定理 6.4.1 知：$v(C)=v(C_n)=v(\mathrm{MP}(C_i,C_j))\geq\alpha$成立。

推论 6.4.1 设$A\in F_L(LP(X))$，如果$S\mid\!\!-_{(A,\alpha)}O\ (\alpha>O)$，则$S$是$A-$不可满足的。

证明：因为 S $\mid\!\!-_{(A,\alpha)}O(\alpha>O)$，由可靠性定理知 S $\mid=^A_\alpha O$。如果 S 是 $A-$满足的，则存在赋值v，对任意的$C\in$S，$v(C)\geq A(C)$。所以有$v(O)\geq\alpha>O$，矛盾。故 S 是$A-$不可满足的。

定义 6.4.6 设 S 是极简规则型子句集，若对任意赋值v，都存在 $G_i\in S$，使得$v(G_i)\ \leq\alpha$，称子句集 S 是α一不可满足；若存在赋值v，对任意 $G_i\in S$，使得$v(G_i)\ \geq\alpha$，称子句集 S 是α一可满足。

定理 6.4.4 设 G 是系统 LP(X)中的一个公式，则 G 是α一恒假的充分必要条件是 G 的极简规则型子句集 S_G是α一不可满足。

证明：设 $G=G_1\wedge G_2\wedge\cdots\wedge G_n$，则$S_G=\{G_1,G_2,\ldots,G_n\}$，其中$G_i(i=1,2,\ldots,n)$为极简规则型子句。若 G 的极简规则型子句集 S_G是α一不可满足的，则若对任意赋值v，都存在 $G_j\in S$，使得$v(G_j)\ <\alpha$。又

$$v(G)=v(G_1\wedge G_2\wedge\cdots\wedge G_j\wedge\cdots\wedge G_n)$$

$$= v(G_1)\wedge v(G_2)\wedge\cdots\wedge v(G_j)\wedge\cdots\wedge v(G_n)<\alpha$$

所以 G 是α-恒假的。

反之，若 G 是α-恒假的，则对任意赋值v，$v(G)<\alpha$，而 $G=G_1\wedge G_2\wedge\cdots\wedge G_n$，所以有

$$v(G_1)\wedge v(G_2)\wedge\cdots\wedge v(G_i)\wedge\cdots\wedge v(G_n)<\alpha$$

故存在 $G_i\in S_G$，使得$v(G_i)\leq\alpha$，所以 S_G 是α－不可满足。

定理 6.4.5 设$A\in F_L(LP(X))$，如果 $S\mid\!-_{(A,\alpha)}O(\alpha>O)$，则 S 不是$\alpha$－可满足的。

证明：假设 S 是α－可满足的，则存在赋值v，对任意的 $G_i\in S$，$v(G_i)>\alpha$ 成立。若 $S\mid\!-_{(A,\alpha)}O(\alpha>O)$，则 O 要么是 S 中子句，要么是 MP 归结式，若为 S 中子句，则有$v(O)>\alpha>O$ 矛盾，若 O 是 MP 归结式，由于 MP 归结式是其母式的逻辑推论，故亦有$v(O)>\alpha>O$，矛盾，所以假设 S 是α－可满足的不成立，故 S 不是α－可满足的。

定理 6.4.6 设 S 是极简规则型子句集，$A\in F_L(LP(X))$，任取 $C\in S$，$A(C)\geq\alpha$，若 S 是A－可满足，则 S 是α－可满足。

证明：若 S 是A－可满足，因为对任意$C\in S$，$A(C)\geq\alpha$，故存在赋值v，使得$v(C)\geq A(C)\geq\alpha$，于是S是α-可满足。

类似可证：

定理 6.4.7 设 S 是极简规则型子句集，$A\in F_L(LP(X))$，任取 $C\in S$，$A(C)\leq\alpha$，若 S 是A－不可满足，则 S 是α－不可满足。

定义 6.4.7 若 S 是极简规则型子句集，且 S 中的每个极简规则型子句至多含一个蕴涵“→”，称子句集 S 为极简规则型 Horn 子句集。

定理 6.4.8 设 S 为极简规则型 Horn 子句集，若对任意 $C\in S$，$A(C)\geq\alpha$，则 S 是α－不可满足当且仅当 $S\mid\!-_{(A,\alpha)}O$。

证明：设 S 为极简规则型 Horn 子句集，则在极简规则型 Horn 子句上的 MP 归结形式上与二值逻辑中的两子句消去互补对的二元归结是对应的，极简

规则型 Horn 子句集 S 与二值逻辑中的 Horn 子句集对应，极简规则型 Horn 子句集 S 是α－不可满足与二值逻辑中 Horn 子句集是不可满足对应。S 中每个子句的信度都大于等于α，表示只要可以进行 MP 归结，那么 MP 归结式就可以加入(A,α)-归结演绎序列中。而二值逻辑中的 Horn 子句集是不可满足的当且仅当存在从 Horn 子句集到空子句的归结演绎，对应极简规则型 Horn 子句集 S，则为 S 是α－不可满足的当且仅当存在从 S 到 O 的(A,α)-归结演绎，所以有 $S \mid\!\!-_{(A,\alpha)} O$。

定义 6.4.8 一般格值逻辑系统 LP (X)的 Petri 网推理模型为七元组 LPN =(P，T，F，W，M_0，A，μ)，其中：

P，T，F 与定义 6.2.1 同；

W： F → {1}为权函数；

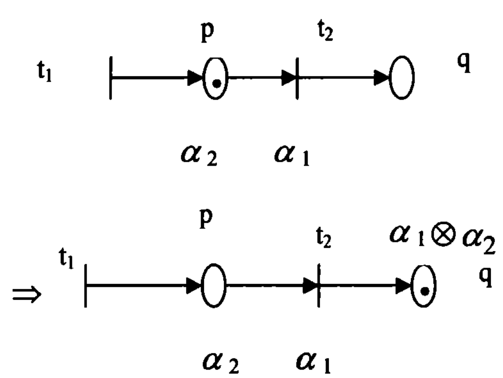
M_0： P →N∪{0}为初始标识；

A：T→L，$\forall t_i \in T, A(t_i)=\alpha_i$，表示每个子句的信度；

$\mu: P \to$L，$\forall p \in P, \mu(p)$表示位置 p 所表示的命题的信度。

定义 6.4.9(激活规则)若*t 中每一个位置结点(包含广义)至少含一个标识，α为阈值，且满足：$\forall p \in {}^*t, \mu(p) \ge \alpha$，则 t 可激活，p 失去一个标识，每个 t 的后集中元素获得一个标识，且$\forall q \in t^*, \mu(q) = \inf_{p \in {}^*t} \mu(p) \otimes A(t)$。

语法推理规则 MP 可表示为：



其中 t_1 是源变迁，总可以激活，那么位置结点 p 就有初始标识，且 $\mu(p)=\alpha_2$，若 $\alpha_2>\alpha$，那么 t_2 可以激活，且其后集 q 获得标识，且 $\mu(q)=\mu(p)\otimes A(t_1)=\alpha_1\otimes\alpha_2$。依此下去，可以得到标识序列，位置及标识构成$(A,\alpha)$-归结演绎的中间推理结果。在$\alpha$—自动归结推理中，将证明的命题的“逆”命题加入到子句集合中，需从子句集的 Petri 网模型给出所证命题在α—水平上的判定。

6.4.1 T-不变量推理算法

在 LP(X)系统的 Petri 网推理模型中，如果没有广义位置结点存在，那么其模型就完全同于模糊逻辑的 Petri 网模型[100]，可以用 T-不变量法进行算法设计与判定。如果将广义位置结点表示的命题作为一个广义文字，那么此时的模型又归为 Horn 型 Petri 网模型结构，同样用 T-不变量法判定。若将广义位置结点作为一个子网结构，这时就不能用 T-不变量法，需考虑其它的方法进行判定。本节的推理不涉及这种情况，仅考虑 n- ESRF($l_2\le 1$，至多有一个蕴涵→的极简规则型 Horn 子句，此时为 Horn 型 Petri 模型(每个转移结点的后集最多只有一个位置结点)。

LP(X)系统中子句集 S 的 Petri 网的 T-不变量求法与一般方法即算法 1.1 有所不同，不但要考虑子句的信度，而且需反映命题信度的传递过程，即流动情况。其计算方法推广如下：

算法 6.1 设 LPN 网有n个转移结点，m个位置结点，$C=(c_{ij})_{n\times m}$为 LPN 网的关联矩阵，I_n为n阶单位阵，$\alpha\in$ L 为阈值。

1) 令 A=C；D=I_n；

2) 重复 i=1 到 i=n：若 $A(t_i)<\alpha$ (假定第 i 行表示t_i对应的子句)，则将(D

¦ A)中第 i 行消除；

3) 设 k 为有标识的位置结点所在 A 中的列数，且在 A 中该列的非零元素所在行只有这个非零元素，若这样的列数不止一个，任意选择一个，转 4)，若没有这样的列，转 5)；

4) 对矩阵 A 中的第 k 列元素符号相反的任意两行，在矩阵(D ¦ A)中进行相加运算，并将产生的新行加入(D ¦ A)中，然后从(D ¦ A)中消除在 A 中第 i 列元素不为零的行，尔后消除第 i 列，并将位置结点的信度流动到获得标识的位置结点(标注在位置结点的后面)，转 3)；

5) 若 A 中某一行元素全为零，则 D 中该行的行向量即为 LPN 的 T-不变量。

定理 6.4.9 设 $S\cup\{G\to O\}$是极简规则型 Horn 子句集，则存在从 S 到 $G\to O$ 的(A,α)-归结演绎的充要条件是 $S\cup\{G\to O\}$的 LPN 网中存在 T－不变量 $X\geq 0$，且 $X(t_g)\neq 0$。其中 t_g 表示 $G\to O$ 的变迁，用一个目标变迁表示。

证明：由于 $S\cup\{G\to O\}$是极简规则型 Horn 子句集，所以它的 Petri 网为 Horn 型 LPN 网。且从初始标识回到初始标识时，变迁结点激活的次数构成 LPN 网的 T-不变量。而每次变迁结点的激活与一次 MP 归结对应，所以激活变迁序列即对应从 S 到目标子句 G 的(A,α)-归结演绎序列，故结论成立。

定理 6.4.10 设 $S\cup\{G\to O\}$是极简规则型 Horn 子句集，$S\cup(G\to O)$的 LPN 网中存在 T－不变量 $X\geq 0$，且 $X(t_g)\neq 0$ 的充要条件是存在从 S 到 O 的(A,α)-反驳，其中 t_g 表示 $G\to O$ 的变迁。

证明：由定理 6.4.9 知，若 $S\cup\{G\to O\}$的 LPN 网中存在 T－不变量 $X\geq 0$，且 $X(t_g)\neq 0$，则存在从 S 到 G 的(A,α)-归结演绎 ϖ，而在 ϖ 后加上序列$(G\to O,\ \beta_1)$，$(O,\ \beta_2)$，则是从 S 到 O 的(A,α)-反驳。

反之，若存在从 S 到 O 的(A,α)-反驳，由于是 Horn 子句集，故除子句 $G\to O$ 外，其它子句不含 O，于是存在从 S 到 G 的(A,α)-归结演绎。由定理 6.4.9 知，在 $S\cup(G\to O)$的 LPN 网中存在 T－不变量 $X\geq 0$，且 $X(t_g)\neq 0$。

定理 6.4.11 设 $S\cup\{G\to O\}$是 LP(X)系统中极简规则型 Horn 子句集。若 $S\cup(G\to O)$的 LPN 网中存在 T－不变量 $X\geq 0$，且 $X(t_g)\neq 0$，则 S 是 A－不可满足的。其中 t_g 是表示 $G\to O$ 的目标变迁。

证明：若 $S\cup(G\to O)$的 LPN 网中存在 T－不变量 $X\geq 0$，且 $X(t_g)\neq 0$ ，由定理 6.4.10 ，存在从 S 到 O 的(A,α)-反驳$(\alpha>O)$，由推论 6.4.1，得 S 是 A－不可满足的。

例 6.4.1 设 L=[0，1]，对于任意 $x,y\in[0,1]$，定义运算：

$$
\begin{aligned}
&x\wedge y=\min\{x,y\}\\
&x\vee y=\max\{x,y\}\\
&x'=1-x\\
&x\to y=1\wedge(1-x+y)
\end{aligned}
$$

则$(L,\wedge,\vee,',\to)$构成一个格蕴涵代数。

又设子句集 S={ t_1：P（μ_1=0.7）；t_2：$P\to A$（$\mu_2=0.9$）；t_3：$P\to B$（$\mu_3=0.86$）；t_4：$A\wedge B\to C$ （$\mu_4=0.75$）}。

目标变迁(子句) t_g： G= C(0.2)，$\alpha=0.2$ 为阈值。

$S\cup\{G\to O\}$的 Petri 网模型为：

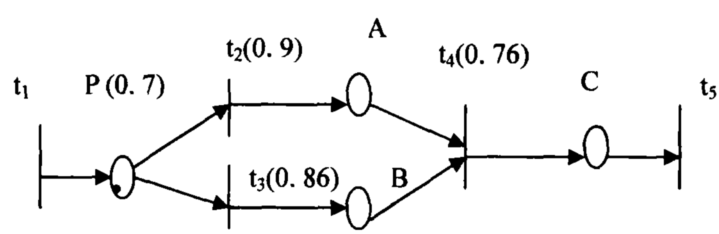


图 6-24 $S\cup\{G\to O\}$的 Petri 网模型

$$
(D\mid A)=\begin{array}{r}
\\
t_1(1)\\
t_2(0.9)\\
t_3(0.86)\\
t_4(0.75)\\
t_5(1)
\end{array}
\begin{array}{c}
\begin{array}{ccccc:cccc}
 & & & & & P(0.7) & A & B & C
\end{array}\\
\left[\begin{array}{ccccc:cccc}
1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0\\
0 & 1 & 0 & 0 & 0 & -1 & 1 & 0 & 0\\
0 & 0 & 1 & 0 & 0 & -1 & 0 & 1 & 0\\
0 & 0 & 0 & 1 & 0 & 0 & -1 & -1 & 1\\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1
\end{array}\right]
\end{array}
$$

$$\Downarrow k=1$$

$$\begin{array}{r|l} & \quad\quad\quad\quad\quad\quad\quad\quad\quad \text{A}(0.6)\ \text{B}(0.66)\ \text{C} \\ \begin{array}{c} t_2(0.9)+t_1(1) \\ t_3(0.86)+t_1(1) \\ t_4(0.75) \\ t_5(1) \end{array} & \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & \vdots & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & \vdots & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & \vdots & -1 & -1 & 1 \\ 0 & 0 & 0 & 0 & 1 & \vdots & 0 & 0 & -1 \end{bmatrix} \end{array}$$

$$\Downarrow k=2$$

$$\begin{array}{r|l} & \quad\quad\quad\quad\quad\quad\quad\quad \text{B}(0.66)\ \text{C} \\ \begin{array}{c} t_3(0.86)+t_1(1) \\ t_4(0.75)+t_2(0.9)+t_1(1) \\ t_5(1) \end{array} & \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & \vdots & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & \vdots & -1 & 1 \\ 0 & 0 & 0 & 0 & 1 & \vdots & 0 & -1 \end{bmatrix} \end{array}$$

$$\Downarrow k=3$$

$$\begin{array}{r|l} & \quad\quad\quad\quad\quad\quad\quad \text{C}(0.31) \\ \begin{array}{c} t_4(0.75)+t_2(0.9)+t_1(1)+t_3(0.86)+t_1(1) \\ t_5(1) \end{array} & \begin{bmatrix} 2 & 1 & 1 & 1 & 0 & \vdots & 1 \\ 0 & 0 & 0 & 0 & 1 & \vdots & -1 \end{bmatrix} \end{array}$$

$$\Downarrow k=4$$

$$\begin{array}{r|l} & \quad\quad\quad\quad\quad\quad \text{C}(0.31) \\ t_5(1)+t_4(0.75)+t_2(0.9)+t_1(1)+t_3(0.86)+t_1(1) & \begin{bmatrix} 2 & 1 & 1 & 1 & 1 & \vdots & 0 \end{bmatrix} \end{array}$$

从而有 T-不变量 X = (2，1，1，1，1)。由于 $X(t_g) = 1>0$，故 $S\cup G\rightarrow O$ 是 0.2-不可满足，即 C(0.2)成立。

分析：T-不变量表示从初始标识(1，0，0，0)开始，再回到初始标识，转移结点激活的次数。当激活结点t_1(源结点总可以被激活)，位置结点 P 获得一个标识，P 的信度为 0.7。这时，结点t_2与t_3的前集{P}都有标识，前集 P 的信度 0.7 大于阈值 0.2，故它们都被激活，从而 A 与 B 各获得一个标识，并且 A 的信度为 $\mu(P)\otimes A(t_2)=0.7\otimes 0.9=0.6$，B 的信度为 $\mu(P)\otimes A(t_3)=0.7\otimes 0.86=0.56$，这样结点$t_4$的前集{A，B}中元素都含有标识，

且信度都大于阈值 0.2，所以结点 t_4 被激活，位置结点 C 获得一个标识，信度为 $\inf\{0.6\otimes 0.75, 0.56\otimes 0.75\}=0.31$ 大于 0.2($\otimes$ 运算保序[158])，进而结点 t_5 被激活，t_5 为漏结点(sink transition, 后集为空)，标识被溢出，这时若再激活源结点，则可以计算出从初始标识(1，0，0，0)开始，再回到初始标识，结点 t_1 激活 2 次，t_2,t_3，t_4 与 t_5 各激活 1 次，故 T-不变量为 X=(2，1，1，1，1)。标识的转移过程可以用一个图(如图 6-25)来表示，这个图就是 Petri 网的可达图，可达图清晰展示了标识的各种可达情形。从初始标识回到初始标识有两条路径：$\sigma_1=t_1t_1t_2t_3t_4t_5$ 与 $\sigma_1=t_1t_1t_3t_2t_4t_5$，同样得 T-不变量 X = (2，1，1，1，1)。

对于 n- ESRF($l_2>1$)的子句，由于在表示该类型子句的 Petri 网模型时，引入了广义位置结点，所以从这种意义上来看，每一个 n-ESRF 都可以是 $l_2\le 1$ 情形，理论上可以解决其推理问题。但是广义位置结点(广义文字)的表示过于笼统，使得子句失去了很多信息，没有很好实现我们的初衷。如何克服这一问题呢？在格 H 蕴涵代数中，$x\to y=x'\vee y$ 以及 $x\to(y\to z)=x\wedge y\to z$ 成立[132]，如果赋值格为格 H 蕴涵代数，那么这样的格值命题逻辑系统就可以解决带任意多个蕴涵的 ESRF 了。因为由 $x\to(y\to z)=x\wedge y\to z$ 以及 $x\to y=x'\vee y$，可以将任意子句化为最多含一个蕴涵算子的子句。又 $x\vee x'=I, x\wedge x'=O$ 总成立，所以我们可以在 I 的水平上进行归结推理，如同在四值非链格值命题逻辑系统 $LP_4(X)$ 中推理那样。

对于一般的格值命题逻辑系统，情形相当复杂，若将广义位置结点看成子网，需考虑子网的合并与展开，这里不再详细讨论。

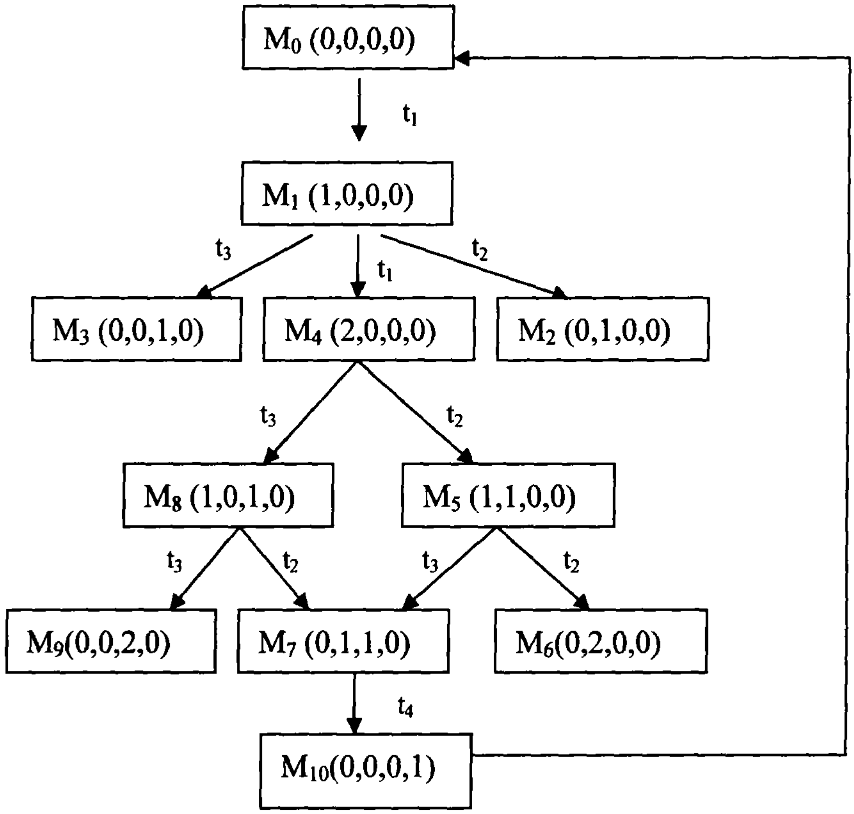


图 6-25 一个 Petri 网的可达图

6.4.2 可达性推理算法

T-不变量推理算法是从 Petri 网的结构出发，结合标识的转移过程，探讨子句集的性质。对于一般的 Petri 网，就不能根据 T-不变量进行判定。而且，即使是 Horn 型 Petri 网，有时 T-不变量也是不易求出。这时候，需要考虑其它的推理方法。

从例 6.4.1 的分析可以看到，可达图完全表征了状态转移过程。不仅由图可以计算出 T-不变量，而且不受 Petri 网模型的限制。由该图可以讨论一般情形的格值命题逻辑系统的推理算法。

下面介绍一些关于可达图的相关知识及可达性的性质，以此讨论 LPN 网的可达图及其推理算法。

定义 6.4.10[161](有向图)一个有向图是一个三元组 $D=<V,E,R>$。其中：V 是

结点集合；E 是有向弧集；R 是一个关系，R： $E \rightarrow V \times V$ 。

定义 6.4.11[161](可达图)一个 Petri 网的可达图是一个有向图 $G=(V,E,R)$，其中：$v \in V$ 表示一类可达的标识；$e \in E$ 表示从一类可达标识到另一类可达标识的有向弧；R 是一个转换关系，R： $E \rightarrow V \times V$ 。

定义 6.4.12[161](可达性问题)设 Petri 网 $PN=(P,T,F,W,M_0)$ 和一个标识 M，是否有 $M \in M_0[>$ ？($M_0[>$ 表示从 M_0 开始所有可达标识的集合)

理论上讲，可达性问题是可判定的[161,162]。而一个 Petri 网的可达图是唯一确定的[117]。可达图的优点是能反映一个网系统的动态行为和一些特征，特别地，对一个有界 Petri 网，它是一个准确刻画。

对于定义 6.4.8 以及定义 6.4.9 定义的 LPN 网以及激活规则，每个规则型 Horn 子句集的 LPN 网对应唯一可达图。又从 LP(X)系统中的 MP 规则的 Petri 网模型(图 6-23)可以发现，由于源变迁总可以被激活，从源变迁激活出发，可激活的转移结点对应的命题就是(A,α)-归结演绎序列的中间结果，因此有下列结论成立。

定理 6.4.12 在 LP(X)系统中，S 是极简规则型 Horn 子句集。若 S 的 LPN 网中目标变迁可激活，则 S 是α-不可满足的。

证明：若 S 的 LPN 网中目标变迁可激活，说明存在从 S 到 G→O 的(A,α)-归结演绎ϖ，而在ϖ后加上序列(O， β)，则是从 S 到 O 的(A,α)-归结演绎。而归结演绎序列中的每一个子句要么是 S 中子句，要么是 MP 归结式，若 O 是 S 中子句，则结论显然成立，若为一 MP 归结式，由于 MP 归结式是其亲本子句的逻辑推论，故对任一赋值v，在 S 中存在子句 G_i，使得$v(G_i)<\alpha$，所以 S 是α-不可满足的。

定理 6.4.13 设 S 是极简规则型 Horn 子句集，S 的 LPN 网中如果有标识的位置结点信度都大于α。则 LPN 网中存在从零标识到标识 $M_d(t_g)\neq 0$(t_g 为目标变迁)的转移系列的充分必要条件为 S 是α-不可满足的。

证明： S 的 LPN 网中，如果有标识的位置结点信度都大于α，那么只要

转移结点满足点火条件，结点可激活，这时的 LPN 网有一般 Horn 型 Petri 网的性质。

(必要性)若 LPN 网中存在从零标识到标识 $M_d(t_g) \neq 0$ (t_g 为目标变迁)的转移系列，说明目标变迁可激活，由定理 6.4.12 知：S 是 α-不可满足的。

(充分性) S 是 α-不可满足的，由定理 6.4.8 知：存在从 S 到 O 的 (A,α)-归结演绎 ϖ，反映到 LPN 网中表示目标变迁可激活，而 S 是 Horn 子句集，一定存在源变迁，故存在从零标识到标识 $M_d(t_g) \neq 0$ (t_g 为目标变迁)的转移系列。

以上结论是针对 Horn 型子句集，而一般的子句集有如下结论成立。

定理 6.4.14 在 LP(X)系统中，若 S 是极简规则型子句集，S 的 LPN 网中，有标识的位置结点信度都大于 α。如果目标变迁可激活，则子句集 S 是 α-不可满足的。

证明：在 LP(X)系统中，若 S 是极简规则型子句集，S 的 LPN 网中，如果有标识的位置结点信度都大于 α，那么此时的 LPN 网相当于二值逻辑中子句集的 Petri 网。

若目标变迁可激活，则表明 O 可由 S 经 (A,α)-演绎推出，由 MP 归结的性质知，极简规则型子句集 S 是 α-不可满足的。

关于标识可达的判定有很多，下面介绍一种由关联矩阵计算的方法[117]。

关联矩阵的第 i 行表示激活变迁 i 后各位置结点标识为变化情况。若设 A 为关联矩阵，记 M_k 为一个 $m \times 1$ 的列矩阵，它的第 i 个分量是 $M_k(p_i)$，即位置 p_i 处的标识个数。当变迁实施序列逐一点火时，标记 M_k 是第 k 次点火一个 t_j 从另一个标记 M_{k-1} 变换的结果。于是有矩阵状态方程：

$$M_k = M_{k-1} + A^T u_k, \text{k} = 1,2,\ldots$$

其中 u_k 为 $n \times 1$ 列矩阵，除了第 i 个分量为 1 外，其余每一个分量均为 0。

假定 M_d 从 M_0 可达，点火序列为 $u_1, u_2, \cdots, u_d$，由状态方程有

$$M_d = M_0 + A^T \sum_{k=1}^{d} u_k$$

可记为

$$A^T x = \Delta M$$

其中$\Delta M = M_d - M_0$，$x = \sum_{k=1}^{d} u_k$为点火记数向量，其第 j 分量表示变迁t_j从初始标记M_0变换到M_d的过程中必须点火的次数。

定理 6.4.15[117]方程$A^T x = \Delta M$有解的充要条件为$\Delta M^T y = 0$，其中y为$Ay = 0$的解。

设$r = R(A)$，令$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$，$A_{12}$为$r$阶可逆方阵，则$Ay = 0$的解为$(m-r)\times m$矩阵$B_f = \left[I_\mu \;\vdots\; -A_{11}^T (A_{12}^T)^{-1}\right]$的$m-r$个行，其中$I_\mu$为$\mu = m-r$阶单位阵。从而有$AB_f^T = 0$，所以$\Delta M^T y = 0$等价于$B_f \Delta M = 0$，故有如下可达的必要条件成立。

定理 6.4.16[117]若M_d从M_0可达，则$B_f \Delta M = 0$，其中$\Delta M = M_d - M_0$，$B_f = \left[I_\mu \;\vdots\; -A_{11}^T (A_{12}^T)^{-1}\right]$。

定理 6.4.17[117]如果$\Delta M = B_f^T z$，z是非零的$\mu \times 1$列矩阵，其中$\Delta M = M_d - M_0$，$B_f = \left[I_\mu \;\vdots\; -A_{11}^T (A_{12}^T)^{-1}\right]$，则$M_d$不能由$M_0 (\neq M_d)$可达。

上述结论是对一般 Petri 网成立，而在 LPN 网中，标识的转移除了一般 Petri 的激活条件外，还另外附加了条件，即激活变迁的前集中每一个元素的信度必须大于α(阈值)。在一个 LPN 网中，如果有标识的位置结点信度都大于α，则上述两定理对 LPN 网也成立。

定理 6.4.18 LPN 网中，如果有标识的位置结点信度都大于α。若M_d从M_0

可达，则$B_f\Delta M=0$。其中$\Delta M=M_d-M_0$，$B_f=\left[I_\mu \;\vdots\; -A_{11}^T(A_{12}^T)^{-1}\right]$。

定理 6.4.19 LPN 网中，如果有标识的位置结点信度都大于α，且$\Delta M=B_f^Tz$，z是非零的$\mu\times1$列矩阵，其中$\Delta M=M_d-M_0$，$B_f=\left[I_\mu \;\vdots\; -A_{11}^T(A_{12}^T)^{-1}\right]$，则$M_d$不能由$M_0(\neq M_d)$可达。

推论 6.4.2 若 S 是极简规则型子句集，S 的 LPN 网中有标识的位置结点信度都大于α。若存在非零向量z，使得$M_d=B_f^Tz$，其中$B_f=\left[I_\mu \;\vdots\; -A_{11}^T(A_{12}^T)^{-1}\right]$，则$M_d$不能从$M_0$=0 可达。

由上述定理结合定理 6.4.13 与定理 6.4.18 可得到以下结论：

定理 6.4.20 设 S 是 Horn 型子句集，S 的 LPN 网中，如果有标识的位置结点信度都大于α。若 S 是α-不可满足的，则存在M_d ($M_d(t_g)\neq0$，t_g为目标变迁)，满足$B_fM_d=0$。

证明：定理6.4.13知：因S是α-不可满足的，则存在M_d满足$M_d(t_g)\neq0$(t_g为目标变迁)从M_0=0 可达。又由定理 6.4.18 知：$B_f\Delta M=0$，其中$\Delta M=M_d-M_0$，$B_f=\left[I_\mu \;\vdots\; -A_{11}^T(A_{12}^T)^{-1}\right]$。

定理 6.4.21 设 S 是极简规则型 Horn 子句集，S 的 LPN 网中有标识的位置结点信度都大于α。若{M_d | $M_d(t_g)\neq0$，t_g为目标变迁}={M_d | $\Delta M=B_f^Tz$，$\Delta M=M_d-M_0$，z是非零的$\mu\times1$列向量}，则 S 是α-可满足的(α为阈值)。

证明：若{M_d | $M_d(t_g)\neq0$，t_g为目标变迁}={M_d | $M_d=B_f^Tz$，z是非零的$\mu\times1$列矩阵}，说明M_d ($M_d(t_g)\neq0$，t_g为目标变迁)不能由初始标识M_0可达，由定理 6.4.13 知，S 是α-可满足的(α为阈值)。

结论与展望

本章对全文的工作及有待于进一步研究的问题给以总结。

结　论

在人工智能研究的众多课题中，自动推理一直是一个十分活跃的中心课题之一。对经典逻辑以及基于经典逻辑的推理，人们进行了大量的、卓有成效的研究，而非经典逻辑以及基于非经典逻辑的不确定推理的研究正处于快速发展阶段。本文针对这一课题，进行了一些研究，建立了一些理论与方法。

本文基于图分析技术 Petri 网，主要对算子命题逻辑系统、算子模糊逻辑系统以及格值命题逻辑系统 LP(X)中的自动推理进行了系统深入的研究，取得了如下成果。

1. 根据 T-不变量判别法的思想，并结合归结策略，提出了多种矩阵归结推理方法，并证明了算法的完备性和有效性。

2. 删除归结是结合 Petri 网中标识的流动规则以及归结原理而给出的推理方法。它使得我们可以直接从 Petri 网的结构入手，进行自动归结推理，适合一般子句集的归结推理。文中利用删除归结原理对 Horn 基子句集、一般基子句集以及一阶 Horn 子句集的自动推理进行了研究。并比较了 T-不变量法与删除归结算法的优缺点，得到了有意义的结论。

3. 提出了算子命题逻辑系统，定义了系统中的λ－恒假、λ－归结以及λ－归结演绎等概念，讨论了它们的逻辑的性质。通过将命题公式表示为极简规则型子句集，给出了算子命题子句集的 Petri 网模型。讨论了算子命题逻辑的 T-不变量推理算法和删除归结推理算法，证明了推理算法的完备性。

4. 提出了算子模糊逻辑系统，定义了算子模糊逻辑中的λ－归结以及λ－归结演绎，证明了算子模糊逻辑中的提升引理以及λ－归结的完备性。在讨论算子模糊逻辑的逻辑性质的基础上，给出了算子模糊逻辑公式的两种 Petri 网模型即算子模糊 Horn 子句和一般算子模糊子句的 Petri 网模型，并分别讨论了基于两种模型的推理算法－T-不变量推理算法与删除归结推理算法，证明了推理算法的完备性。

5. 给出了格值命题逻辑系统 LP(X)中子句的规则型范式以及规则型子句集。在极简规则型子句上定义了 MP 归结式、(A,α)—归结演绎、A-不可满足以及α-逻辑推论等基本概念。根据 MP 归结式定义(A,α)—归结演绎能将归结自动推理与不确定性推理建立一定的联系。并证明了语义推导与语法推导的可靠性与弱完备性。在此基础上对 LP(X)中公式的 Petri 网模型进行了研究，提出了广义位置结点的概念，进一步对 LP(X)系统中基于 Petri 网模型的自动推理进行了判定。讨论了 LP(X)系统的 Petri 网模型上的 T-不变量推理，证明了推理算法的完备性。特别地，讨论了 LP(X)系统中的可达性推理算法，得到了有意义的结论。

以上的研究，为进一步研究多值逻辑系统的自动推理提供了新的思想与方法。

展　望

本文以 Petri 网为工具，利用 Petri 网的分析技术，研究了算子命题逻辑、算子模糊逻辑以及格值命题逻辑系统的 Petri 网推理算法。讨论了算法的可靠性与完备性，得到了一些重要的研究结果。但仍然有与本文研究直接有关的许多重要工作尚待进行。

作者认为，下述与本文有关的工作尚需进一步的研究：

1. 在第二章中，我们针对基子句集，讨论了几种矩阵归结算法。在对子句的矩阵表示时，若原子出现的次序不一样，是否会影响归结的效率？是否存在最好的次序，使得归结的效率最高？这些问题值得探讨。

2. 如何将算子命题逻辑中的结论推广到命题赋值域为[0,1]上？若在算子模糊逻辑中，将函数映射到区间[0,1]，结论又会如何变化？

3. 格值命题逻辑系统 LP(X)中的公式的 Petri 网模型是否存在同构？如何定义？性质又如何？

4. 关于格值逻命题辑系统 LP(X)中的 Petri 网自动推理的进一步研究。

5. 关于格值一阶逻辑系统 LF(X)中的 Petri 网自动推理的研究。

6. 如何利用 Petri 网的其它分析技术，对一般的模糊逻辑系统的归结推理进行研究？

7. Petri 网推理与神经网络推理有许多共同之处，能否将两者结合，提出新的推理方法？

8. 对推理方法，特别是基于多值逻辑的计算机实现方法的研究。

尽管这些问题能够在本文研究工作的基础上逐步推进、展开进行研究，并有望得到满意的解决，但仍然需要大量的努力和探索，这也是作者进一步开展研究工作的重点。作者相信，本文提出的理论和方法以及进一步要做的工作，将会在人工智能领域中有着广阔的应用前景。

致 谢

衷心感谢我的导师徐扬教授。本文是在导师的悉心指导下完成的。多年来，导师在学业上对作者严格要求，在生活上导师给予了我无微不至的关心和照顾，花费了大量的心血。在导师的严格要求和关心爱护下，我的学业有了很大的进步，才能完成这篇博士论文，对此我牢记于心。导师渊博的知识、严谨的治学态度、无私的奉献精神和忘我的工作热情使我受益匪浅，永远激励我拼搏进取、奋发向上，是我终生学习的榜样。借论文完成之际，谨表对导师崇高的敬意和诚挚的感谢！

衷心感谢智能控制开发中心的宋振明教授、黄天民教授、秦克云教授、赵海良教授、吴建乐副教授以及吴德珍和关效东老师。在论文的写作过程中，他们给过我热心的指点和生活上无私的帮助。

感谢智能开发中心的刘军、马骏、裴峥、王伟、孟丹、缪志农、李文江、邱小平、杜福银等每一位博士对我学术上的帮助。

感谢阮达教授多年来在智能控制开发中心的讲学，从中我学到了丰富的知识，也学到了许多科研方法。

感谢我的亲人以及工作单位的领导和同事在生活上和学习上给予的关心、照顾和支持，使我能安心学习，顺利完成学业。

本文工作部分得到国家自然科学基金支持，特致以感谢。

参考文献

[1] Swain, Anjan Kumar, Morris, Alan S. A Novel Hybrid Evolutionary Programming Method for Function Optimization. Proceedings of the IEEE Conference on Evolutionary Computation, ICEC, 2000(1): 699-705.

[2] L.Wos, J.A.Robinson, D.F.Carson. Efficiency and completeness of the set of support strategy in theorem proving. J.ACM, 1965, Vol12(4): 536-541.

[3] A.P.Dempter. Upper and lower probabilisties induced by a multi-valued mapping. Ann. Mathematical Statistics, 1967, Vol 38: 325-339.

[4] Shafer, G. A Mathematical Theory of Evidence. Princeton, Nj: Princeton University Press, 1979.

[5] Judea Pearl. Fusion, propagation and structuring in belief networks. Artificial Intelligence, 1986, Vol 29:241-288.

[6] Judea Pearl. Belief networks revisited. Artificial Intelligence, 1993, Vol 59: 49-56.

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

[8] Mccarthy, J. Circumscription –a form of non-monotonic reasoning. Artificial Intelligence, 1980, Vol 13:27-39.

[9] Mcdermott, D. Non-monotonic logic I. Artificial Intelligence, 1980, Vol 13:41-72.

[10] L.A.Zadeh. Fuzzy Sets. Information and Control, 1965, Vol 8: 338-353.

[11] Yeung, S.Tsang, E.C.C. Weighted fuzzy production rules. Fuzzy Sets and Systems, 1997, Vol88(30): 299-313.

[12] Baldwin, J.F. Fuzzy logic and fuzzy reasoning. Internat. J.Man-Machine Stud., 1979, Vol 11: 456-480.

[13] Mizumoto, M. Extended fuzzy reasoning. Approximate Reasoning in Expert Systems, North-Holland, Amsterdam, 1985: 67-76.

[14] Tsukamoto,T. An approach to fuzzy reasoning method. Advances in Fuzzy Set Theory and Application, North-Holland, Amsterdam, 1979: 137-149.

[15] Yager, R.R. Using approximate reasoning to represent default knowledge. Artificial Intelligence, 1987, Vol 31 : 99-112.

[16] Dubois, D, Prade, H. Fuzzy relation equations and causal reasoning. Fuzzy Sets and Systems, 1995, Vol.75: 119-134.

[17] 张文修, 梁怡. 不确定性推理, 西安交通大学出版社, 西安, 1996.

[18] Gorzalczany, M.B. A method of inference in uncertainty reasoning based on interval-valued fuzzy sets. Fuzzy Sets and Systems, 1987, Vol.21: 1-17.

[19] Yuan, B,Pan , Y, Wu, W. On normal from based interval-valued fuzzy sets and their applications to uncertainty reasoning. Internat. J. Gen. System, 1995, Vol.23: 241-254.

[20] Chen, S.M. New methodology to fuzzy reasoning for rule-based expert systems. Cybernetics and Systems, 1995, Vol.26: 237-263.

[21] Chen, S.M. Bidirectional uncertainty reasoning based on interval-valued fuzzy sets. Fuzzy Sets and Systems, 1997, Vol.91: 339 - 353.

[22] Hsiao, W.H, Jong, W.T, Chen, S.M, Lee, C.H. Interval-valued bi-directional uncertainty reasoning techniques for rule-based systems. Proc. of the 14th National Conference on Defense Management, Taipei, Taiwan, 1996: 915-927.

[23] Turksen, I.B. Interval valued fuzzy sets based on normal forms. Fuzzy Sets and Systems, 1999, Vol.20: 191-210.

[24] Takagi, T, Imura, A, Ushida, H, Yamaguchi, T. Multi-layered reasoning by means of conceptual fuzzy sets. Internat.J.Intelligent Systems, 1996, Vol11: 97-111.

[25] Raha,S, Ray,K.S. Uncertainty reasoning with time. Fuzzy Sets and Systems, 1996, Vol107(1) : 59-79.

[26] Raha,S, Ray,K.S. Reasoning with vague default. Fuzzy Sets and Systems, 1997, Vol91 (3): 327-338.

[27] Raha,S, Ray,K.S. Reasoning with vague truth. Fuzzy Sets and Systems, 1999, Vol105 (3) : 385-399.

[28] Emami, M.R. A unified paramrterized formulation of reasoning in fuzzy modeling and control. Fuzzy Sets and Systems, 1999, Vol.108: 59-81.

[29] Kuhu, P.Nikhil, R.P. A neural-fuzzy system for inference. Internat. J. Intelligent Systems, 1999, Vol.14: 1155-1182.

[30] 王国俊. 模糊推理的全蕴涵三 I 算法. 中国科学 E, 1999, Vol.29(1): 43-53.

[31] 宋士吉, 吴澄. 模糊推理的反三 I 算法. 中国科学 E , 2002, Vol.32 (2): 230-246.

[32] Chen,S.M. Bidirectional uncertainty reasoning for rule-based Systems using interval-valued fuzzy sets. Fuzzy Sets and Systems, 1997, Vol.117: 279-296.

[33] Yager, R.R. Uncertainty reasoning and conflict resolution. Internat.J. Uncertainty Reason, 2000, Vol.25: 15-42.

[34] Niittymaki,J, Turunen, E. Traffic signal control on similarity logic reasoning. Fuzzy Sets and Systems, 2003, Vol.133 (1): 109-131.

[35] 徐扬, 乔全喜, 陈超平, 秦克云. 不确定性推理. 西南交通大学出版社, 成都, 1994.

[36] Yang Xu, E.E. Kerre, Rua n, D., Song, Z.M. Fuzzy Reasoning Based on the Extension Principle. Int. J. of Intelligent Systems, 2001, Vol16(4): 469-495.

[37] Yang Xu, Jun Liu, Da Ruan, LI, W.J. Fuzzy Reasoning Based on Generalized Fuzzy If-Then Rule. Int. J. of Intelligent Systems, 2002, Vol17 (10): 977-1006.

[38] Xu,Y., Ruan,D., Liu,J. Approximate reasoning based on lattice-valued prepositional logic L_{vpl}. Fuzzy Sets Theory and Applications, Kluwer Academic Publishers, 2000: 81-105.

[39] Pavelka,J. On fuzzy logic I: Many-valued rules of inference, II: Enriched residuated lattices and semantics of prepositional calculi, III: Semantical completeness of some many-valued propositional calculi. Zeitschr.F. Math.Logik und Grundlagend, Math., 1979: 25, 45-52,119-134,447-464.

[40] Novak,V., Perfilieva, I., Mojckojr, j. Mathematical Principles of Fuzzy Logic. Kluwer, 1999.

[41] Robinson, J.A. The generalized resolution principle. Machine Intelligence, 1968, Vol.3: 77-94.

[42] J.R.Slagle. Automatic theorem proving with renamable and semantic resolution. J.ACM, 1967, Vol 14(4): 687-697.

[43] D.W.Loveland. A linear format for resolution, Proc. IRIA Symp. Automatic Demonstration, Springer-Verlag, New York, 1970: 147-162.

[44] D.Luckham. Refinements in resolution theory. Proc. IRIA Symp. Automatic Demonstration, Versailes, France, Springer-Verlag, New York, 1970: 163-190.

[45] 刘叙华. 基于归结方法的自动推理. 科学出版社, 1994.

[46] 陆汝钤. 人工智能(下). 科学出版社, 1996.

[47] C.L.Chang. The unit proof and the input proof in theorem proving. J.ACM, 1970, Vol 17(4): 698 - 707.

[48] L.Henschen, L.Wos. Unit Rafutations and Horn Sets. J.ACM, 1974, Vol 21(4): 590 - 605.

[49] 陆汝钤. Horn 集的消解原理. 中国科学(A 辑), 1981, Vol 7: 896 - 903.

[50] 陆汝钤. 强有序输入消解原理. 中国科学(A 辑), 1981, Vol8: 1035 - 1042.

[51] 刘叙华, 欧阳继红. 自动定理证明中 RUE-NRF 单元, 输入和锁演绎. 吉林大学学报, 1989, Vol 2: 60 - 66.

[52] 刘叙华. Horn 集上的输入半锁归结法. 科学通报, 1985, Vol16: 1201 - 1202.

[53] 刘叙华, 杨凤杰. λ-Horn 集上的 λ-单元锁归结. 吉林大学学报, 1989, Vol 3:34 - 41.

[54] 刘叙华. 广义模糊逻辑和锁语义归结原理. 计算机学报, 1980, Vol2: 97 - 111.

[55] 刘叙华, 狭义模糊逻辑中的锁语义归结, 吉林大学学报, 1980, 4: 56 - 60.

[56] 刘叙华, 肖红. 算子模糊逻辑和 λ-归结方法. 计算机学报, 1989, Vol2: 81 - 91.

[57] 刘叙华, 安直. 算子模糊逻辑及其归结推理的改进. 计算机学报, 1990, Vol12: 890 - 899.

[58] 刘叙华, 司徒芊. 广义λ-调解. 计算机学报, 1995, Vol 18(2): 135 - 140.

[59] 刘叙华, 欧阳丹彤. 广义 Horn 集. 软件学报, 1995, Vol 6(4): 248 - 256.

[60] 刘瑞胜,孙吉贵,刘叙华. 带有约束的缺省逻辑. 吉林大学自然科学学报, 1996, Vol 3: 1- 4.

[61] C.G. Morgan. Resolution for many-valued logics. Logique et Analyse, 1976, Vol 19 (74-76): 311-339.

[62] E. Orlowska. The resolution principle for ω^{+}-valued logic. Fundament Information II, 1978, Vol 1:1-15.

[63] E. Orlowska, S. Wierzchon. Mechnical reasoning in fuzzy logics. Logique et Analyse, 1985, Vol 110-111:193-207.

[64] P.H. Schmitt. Computational aspects of three-valued logic. J.H.Siekmann (Ed.), Proceedings of the Eighth International Conference on Automated Deduction, Springer, LNCS, 1986:190-198.

[65] H.A. Blair, V.S. Subrahamnian. Paraconsistent logic programming. Theoretical computer Science, 1989, Vol 68: 135-154.

[66] N.C.A. da Costa. Automated theorem proving in paraconsistent logics: theory and implementation. Proceedings of the 10th International Conference on Automated Deduction, Germany,Lecture Notes in Artificial Intelligence, Springer, 1990, 449:72-86.

[67] M. Kifer. On the expressive power of annotated logic program. Proceedings of the 1989 North American Conference on Logic Programming, Cleveland, OH, 1069-1089.

[68] M. Kifer, E. Lozinskii, RI. A logic for reasoning with inconsistency. IEEE of the International Symposium on Multiple-Valued Logic, 1991: 253-262.

[69]M. Kifer, V.S. Subrahamnian. Theory of generalized annotated logic programming and its application. Journal of Logic Programming, 1992, Vol 12:

335-367.

[70] J.J. Lu, L.J. Henschen. The completeness of gp-resolution for annotated logics. Information Processing Letters, 1992: 135-140.

[71] J.J. Lu, N.V. Murray, E. Rosenthal. Signed formulas and annotated logics. Proceedings of the 23th IEEE International Symposium on Multiple-Valued Logic, 1993: 48-53.

[72] N.V. Murray, E. Rosenthal. Resolution and path dissolution in multiple-valued logics. Proceedings of the Sixth International Symposium on methodologies for Intelligent Systems, Charlotte, Lecture Notes in Artificial intelligence, 1991, 542: 570-579.

[73] Z. Stachniak. The resolution rule：an algebraic perspective. Proceedings of Algebraic Logic and Universal Algebra in Computer Science Conference, Springer LNCS 425, Heidelberg, 1988: 227-242.

[74] Z. Stachniak. Resolution Proof Systems: An Algebraic Theory. Kluwer Academic Publishers, 1996.

[75] R.C.T. Lee, C.L. Chang. Some Properties of Fuzzy Logic. Information and Control, 1971, Vol 19: 417-431.

[76] R.C.T. Lee. Fuzzy Logic and the Resolution Principle. J. ACM., 1972, Vol 19: 109-119.

[77] M. Mukaidono. On some properties of fuzzy logic. Syst. Comput. Control, 1975, Vol 6(2): 36 - 43.

[78] M. Mukaidono. The necessary and sufficient condition for fuzzy logic function. Proc. of 9th ISMVL, Beth, 1979.

[79] M. Mukaidono. Fuzzy Inference of Resolution Style. Fuzzy Sets and Possibility Theory, Pergamon Press, New York, 1982: 224-231.

[80] J.P. Benejam. La methode de Beth pour la construction de models enlogique a valeurs reeles. Proc. of Extended Abstracts of the Inter. Conf. On Information Proceeding and Management of Uncertainty in Knowledge-Based Systems, Paris,

Frances, 1986: 393-396.

[81] Farinas del Cerro, L. Resolution modal logic, Logics and Models of concurrent Systems Springer-Verlag, Heidelberg, 1985: 27-55.

[82] 陈永义. 不精确推理中的归结方法. 模糊推理的原理和方法论文集, 1990: 41 - 46.

[83] Yang Xu, Da Ruan, E.E. Kerre, Jun Liu. α – Resolution principle based on lattice-valued propositional logic LP(X). Int. J. Information Sciences, 2000, Vol 130: 195-223.

[84] Yang Xu, Da Ruan, Etienne E. Kerre, Jun Liu. α – Resolution principle based on first-order lattice-valued logic LF(X). Int. J. Information Sciences, 2001, Vol 132: 221-239.

[85] 马骏. 基于格蕴涵代数的格值逻辑系统及其自动推理的研究. 西南交通大学博士研究生学位论文, 2002.

[86] 王伟. 格值命题系统 LP(X)中基于α－归结原理的自动推理的研究. 西南交通大学博士研究生学位论文, 2003.

[87] 孟丹. 基于格蕴涵代数的格值逻辑系统的归结自动推理研究. 西南交通大学博士研究生学位论文, 2004.

[88] J Peterson. Petri net theory and the modeling of systems. Prentic-hall, Engle-wood Cliffs, NJ, 1981.

[89] T Murata. Petri nets: Properties analysis and applications. Proc of the IEEE., 1989, Vol77(4):541-580.

[90] Y.S.Ksynchronous systems. Theorent Comput. Sci., 1977, Vol5:25-50.

[91] A Finkel. The minimal coverability graph for Petri net. LNCS, 1993, Vol 674: 210-243.

[92] T Agerwala, Y Choed-Amphai. A syntheais rule for concurrent systems, Pro 5th Design Automation Conf, LasVegas NV, 1978: 305-311.

[93] G Berthelot, Reductions of net and parallel programs, LNCS, 1980, Vol84:277-290.

[94] Lin C. Logical inference of Horn clauses in Petri net models. IEEE Trans on Knowledge and Data Engineering, 1993, Vol 5(3):416-425.

[95] 林闯. Petri 网用于 Horn 子句的逻辑推论. 软件学报, 1993, Vol 4(4): 32-37.

[96] T Murata. A Petri net model for reasoning in the presence of inconsistency. IEEE Trans on Knowledge and Data Engineering, 1991, Vol 3(3):281-292.

[97] 丁彩虹, 黄文虎, 姜兴渭. Petri 网在基于规则系统一致性问题检查中的应用. 机械强度,1999, Vol 9: 170 - 173.

[98] Lin C, Murata T. A Petri net model for nonmonotonic reasoning based on annotated logic program. IEEE Trans on Fundamentals, 1994, VolE77-A(10): 1579-1587.

[99] Peterka G, Murata T. Proof procedure and answer extraction in Petri net model of logic programs. IEEE Trans on software Engineering, 1989, Vol 15(2): 209-217.

[100] 何新贵. 模糊 Petri 网. 计算机学报, 1994,Vol.17: 946-950.

[101] C.G.Looney, A.R.Alfize. Logical control via Boolean rhle matrix transformation. IEEE Trans. Systens, Man, Cybernetics, 1987,Vol.17(6): 1077-1082.

[102] C.G.Looney. Fuzzy Petri nets for rule-based decision-marking. IEEE Trans. Systens, Man, Cybernetics, 1988, Vol18(1): 178-183.

[103] S.M.Chen. Knowledge representation using fuzzy Petri nets. IEEE Trans. Knowledge and Data Eng., 1990, Vol2(3): 311-319,.

[104] H.Scarpelli, F.Gomide, and R.Yager. A reasoning algorithm for high-level fuzzy petri nets. IEEE Trans. Fuzzy Systems, 1996, Vol4(3): 282-294.

[105] A.Konar and A,K.Mandal. Uncertainty management in expert systems using fuzzy Petri nets. IEEE Trans. Knowledge and Data Eng., 1996, Vol8(1): 96-105.

[106] T.Cao et al. Task sequence planning using fuzzy Petri nets. IEEE Trans. Systems, Man, Cybernetics, 1995, Vol 25(5): 755-768.

[107] A.j.Bugarin , S.Barro. Fuzzy reasoning supported by Petri nets. IEEE Trans,

Systems, 1994, Vol.2(2): 135-149.

[108] S.Ahson. Petri net models of fuzzy neural networks. IEEE Trans. Systems, Man, Cybernetics, 1995, Vol25(6): 926-932.

[109] Tommy W.S.Chow, Jin-yan Li. Higher-order Petri net models based on artificial neural networks. Artificial Intelligence , 1997, Vol 92: 289-300.

[110] Witold Pedrycz. Generalized fuzzy Petri nets as pattern classifiers. Pattern Recognition Letters 20, 1999:1489-1498.

[111] 陆维明, 林闯. Petri 网研究: 机遇与挑战. 计算机学报, 1994, Vol 21(4), 1-4.

[112] Freym, G. and Litz, L. Verification and Validation of Control Algorithms by Coupling of Interpreted Petri Nets. Proc. of IEEE SMC'98, San Diego, 1998, Vol1:7-12.

[113] Schmidt, M. A Complexity Measure Based on Selection and Nesting. ACM SIGMETRICS-Performance E valuation Review, V13, No.1 ,1985.

[114] Rechenberg, P. Ein neues MaB Fur Die Softwaretechnische KOMPLEXITAL Von Programmen. Informatik Forschung und Entwicklung, V1 Springer, New York,1986:26-37.

[115] Jalotc P. An integrated approach to software engineering. 2nd Ed. Springer, New York, 1997.

[116] 袁从义, Petri 网原理. 北京：清华大学出版社, 2000.

[117] Tadao Murata. Petri Nets: Properties, Analysis and application, Proceedings of the IEE, 1989, Vol77(4):541-580.

[118] Freym G., Litz, L. A measure for transparency in net based control algorithms Proc. of the IEEE SMC'99, October 12-15, 1999, Tokyom Vol 3:887-892.

[119] Lewism R.W. Programming industrial control systems using IEC 1131-3 IEEE Publishing, London, United Kongdom, 1998.

[120] IEC. International standard 1131 A: Programmable logic controllers. Part 3: Languages, 1992.

[121] Zusem H. A framework for software measurement. Berlin, New York, 1998.

[122] Martinez J, Silva M. A simple and fast algorithm to obtain all invariants of generalized Petri nets. Informatik-Fachbrichte 52, 1982: 301-303.

[123] C.A.Petri. Communication with Automata. New York: Griffiss Air Force Base, Tech, Rep.RADC-TR-65-377, Vol1, Suppl.1, 1966.

[124] Xu, Yang, Liu, Jun, Song, Zhenming, Qin, Keyun. On semantics of L-valued first-order logic L_{vfl}. International Journal of General Systems, 2000, Vol 29(1): 53-79.

[125] Xu, Yang, Da, Ruan, Liu, Jun. Progress and prospect in lattice-valued logic systems based on lattice implication algebras. Applied Computational Intelligence - Proceedings of the 6th International FLINS Conference, Applied Computational Intelligence - Proceedings of the 6th International FLINS Conference, 2004: 29-34.

[126] Xu, Yang, Liu, Jun, Ruan, Da, Lee, Tsu-Tian. On the consistency of rule bases based on lattice-valued first-order logic LF(X). International Journal of Intelligent Systems, 2006, Vol 21(4): 399-424.

[127] Liu, Jun, Song, Zhenming, Qin, Keyun. Resolution procedure based on a fuzzy logic. IEEE International Conference on Fuzzy Systems, 2000, Vol1: 191-196.

[128] Liu, Jun, Ruan, Da, Xu, Yang, Song, Zhenming. A resolution-like strategy based on a lattice-valued logic. IEEE Transactions on Fuzzy Systems, 2003,Vol 11(4): 560-567.

[129] Ma, Jun, Li, Wen J., Xu, Yang, Song, Zhen M. A model for handling linguistic terms in the framework of lattice-valued logic LF(X). Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics, Vol 2, 2004 IEEE International Conference on Systems, Man and Cybernetics, SMC 2004: 1504-1509.

[130] Ma, J. Chen, S., Xu, Y. Fuzzy logic from the viewpoint of machine

intelligence. Fuzzy Sets and Systems, 2006, Vol 157(5): Mar 1, What is Fuzzy Logic, 628-634.

[131] Qin, Keyun, Pei, Zheng. On the relation between filters and deductive rules. Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, 2003, Vol 5: 4463-4467.

[132] Meng, Dan. Resolution principle based on six lattice-valued proposition logic LP $_6$(X). International Conference on Machine Learning and Cybernetics, 2003, Vol 1: 508-512.

[133] Meng, Dan , Xu, Yang. α-resolution principle based on an intermediate element lattice-valued prepositional logic. Proceedings of the Joint Conference on Information Sciences, Vol6, Proceedings of the 6th Joint Conference on Information Sciences, JCIS 2002: 89-91.

[134] Meng, Dan, Qiu, Xiaoping. Resolution principle based on finite chain lattice-valued proposition logic FCLP(X). IEEE International Conference on Fuzzy Systems, 2003, Vol 1: 61-66.

[135] Li, Yongming. A categorical approach to lattice-valued fuzzy automata, Fuzzy Sets and Systems, 2006, Vol 157(6): 855-864.

[136] Li, Yongming. Fuzzy finite automata and fuzzy regular expressions with membership values in lattice-ordered monoids. Fuzzy Sets and Systems, 2005,Vol 156(1): 68-92.

[137] Li, Wen-Jiang, Xu, Yang. Semantics of lattice-valued tense propositional logic system. Journal of Southwest Jiaotong University, 2004,Vol 39(5): 691-695.

[138] Li, Wenjiang, Xu, Yang. Resolution method on lattice-valued tense propositional logic. Applied Computational Intelligence - Proceedings of the 6th International FLINS Conference, Applied Computational Intelligence - Proceedings of the 6th International FLINS Conference, 2004: 87-92.

[139] Chen, Shuwei, Xu, Yang. Uncertainty reasoning based on lattice-valued

first-order logic L_{vfl}. Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics, Vol 3, 2004 IEEE International Conference on Systems, Man and Cybernetics, SMC 2004: 2237-2242.

[140] Chen, Shuwei, Xu, Yang, Ma, Jun. A linguistic truth-valued uncertainty reasoning model based on lattice-valued logic, Lecture Notes in Artificial Intelligence (Subseries of Lecture Notes in Computer Science). Vol3613, PART I, Fuzzy Systems and Knowledge Discovery: Second International Conference, FSKD 2005. Proceedings, 2005: 276-284.

[141] Guller, Dusan. Binary resolution over Boolean lattices. Fuzzy Sets and Systems, 2006, Vol 157(15): 2100-2127.

[142] Sofronie-Stokkermans,V. Representation theorems and the semantics of (semi)lattice-based logics Proceedings of The International Symposium on Multiple-Valued Logic, 2001: 125-134.

[143] Sofronie-Stokkermans,V. Automated theorem proving by resolution for finitely-valued logics basen on distributive lattices with operators. Mutiple-valued Logic-An Interna. J., 2001, Vol 6(3): 289-344.

[144] Lu, James J. , Murray, Neil V., Rosenthal, Erik. Deduction and search strategies for regular multiple-valued logics. Journal of Multiple-Valued Logic and Soft Computing, 2005, Vol 11(3-4): 375-406.

[145] Cser, Laszlo, Farago, B., Krexner, G., Sharkov, I., Torok, Gy. Atomic resolution neutron holography (principles and realization) Physica B: Condensed Matter, 2004, Vol350(4): 113-119.

[146] Hans Kleine Buning. Resolution for Quantified Boolean Formulas. Information and Computation 117, 1995:12-18 .

[147] 刘清，黄兆华. G-逻辑及其归结推理. 计算机学报，2004, Vol27(7): 865-873.

[148] Lin, Chuang, Qu, Yang. Temporal inference of workflow systems based on time petri nets: Quantitative and qualitative analysis. International Journal of

Intelligent Systems, 2004, Vol 19(5): 417-442.

[149] Liu, Ting,Lin, Chuang. Linear temporal inference of workflow management system based on timed Petri net models. Tien Tzu Hsueh Pao/Acta Electronica Sinica, 2002, Vol 30(2): 245-248.

[150] Li, Xiaoou, Yu, Wen, Lara-Rosano. Dynamic knowledge inference and learning under adaptive fuzzy Petri net framework. IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews, 2000, Vol 30(4): 442-450.

[151] Looney, Carl G. , Liang, Lily R. Inference via Fuzzy Belief Petri Nets. Proceedings of the International Conference on Tools with Artificial Intelligence, 2003: 510-514.

[152] Egorov, A.F. , Shajkin, A.N. Logical modeling under uncertainty based on fuzzy interval Petri nets. Izvestiya Akademii Nauk. Teoriya i Sistemy Upravleniya, No2, 2002: 134-139(Language: Russian).

[153] Lee, Jonathan , Liu, Kevin F.R. Modeling uncertainty reasoning with possibilistic Petri nets. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 2003, Vol 33(2): 214-224.

[154] Aura, Tuomas, Lilius, Johan. A causal semantics for time Petri nets. Theoretical Computer Science, 2000, Vol 243(1-2): 409-447.

[155] Liang, Lily R., Looney, Carl G. Decisionmaking with fuzzy-belief-state-based reasoning. Proceedings of the 2004 IEEE International Conference on Information Reuse and Integration, IRI-2004, Proceedings of the 2004 IEEE International Conference on Information Reuse and Integration, IRI-2004, 2004: 553-558.

[156] Shyi-Ming Chen. Fuzzy backward reasoning using fuzzy Petri nets. Systems, Man and Cybernetics, Part B, IEEE Transactions onVol30, Issue 6, Dec. 2000: 846 – 856.

[157] Shyi-Ming Chen. Weighted fuzzy reasoning using weighted fuzzy Petri nets.

Knowledge and Data Engineering, IEEE Transactions onVol14, Issue 2, March-April 2002: 386 - 397 .

[158] 刘军. 基于格蕴涵代数的格值逻辑系统及格值归结原的研究. 西南交通大学博士学位论文, 1998.

[159] Yang Xu, Da Ruan, Keyun Qin, Jun Liu. Lattice-Valued Logic. Springer-Verlag Berlin Heidelberg New York, 2003.

[160] 秦克云, 徐扬, 宋振明. 基于系统 L(X)的几种近似推理. 模糊系统与数学, 1998, Vol12(2): 55-60.

[161] E.W.Mayr. An algorithm for the general Petri net reachability problem. SIAM,J.Comput., 1984, Vol.13(3): 441-460.

[162] S.R.Kosaraju. Decidability of reachability in vector addition system. Proc. 14th Annual ACM Symp. Theory Computing, San Francisco, May 5-7, 1982: 267-281.

攻读博士学位期间的论文及科研情况

一、论文

1. Shifen Xia, Ming Qing, Tian-min Huang, Yang Xu. Neural network resolution on Horn clause set. Proceedings of the Second International on Machine Learning and Cybernetics, Xi'an 2-5 November 2003: 1682-1686(EI 检索，检索号：0412807141).
2. Shifen Xia , Huang Tian-min, Yang Xu . Neural network resolution on clause set. 模糊系统与数学, 2004, Vol18(2): 77-82.
3. Shifen Xia , Shu-xia Ma, Yang Xu . 并行单元归结. 四川师范大学学报，2004,Vol127(5): 501-504.
4. 夏世芬, 徐扬. 一种矩阵归结方法. 模糊系统与数学, 2005, Vol19(2): 44-48.
5. Shifen Xia , Ming Qing, Yang Xu . Parallel Unit Resolution. Proceedings of 2003 National Conference on Artificial Intelligence (CAAI-10), Guangzhou, 2003,11: 250-130.
6. 夏世芬. 模糊控制模糊加强学习算法. 第九界全国多值逻辑与模糊逻辑学术会议论文集, 四川, 成都, 2000: 167-171.
7. 夏世芬, 徐扬. 一种算子命题逻辑系统及其 T-不变量推理算法. 模糊系统与数学, 2006(已接收, 待发表).
8. 夏世芬, 徐扬. 一种算子模糊逻辑系统及其 Petri 网推理算法. 模糊系统与数学, 2006(已接收,待发表).
9. 夏世芬, 徐扬. 一种算子模糊逻辑系统. 四川师范大学学报, 2006(已接收,待发表).
10. Shifen Xia , Yang Xu . MP resolution-based automated reasoning method for LP(X). The Journal of Fuzzy Mathematics, 2007(待发表).
11. Zhi-nong Miao, Shifen Xia . State space method for Fuzzy Control system description. IEEE SMC'2004 Conference Proceedings, October 10-13, The Hague, The Netherlands, 2004: 382-386.

12. Zhi-nong Miao, Shifen Xia , Yang Xu . Fuzzy controller using multiple rule bases. Proceedings of the Third International Conference on Machine Learning and Cybernetics, Shanghai, 26-29 August 2004: 1093-1098.

13. Ming Qing, Hai-liang Zhao, Shifen Xia , Xue-Fang Wang. Contruction king of fuzzy systems based on neural networks techniques. Proceedings of the Second International on Machine Learning and Cybernetics, Xi'an 2-5 November 2003:2629-2633(EI 检索，检索号：0412802336).

二、科研情况

1. 国家自然科学基金项目：智能控制中的信息优化与决策方法研究(69874033),1999.1-2001.2, 主研；
2. 国家自然科学基金项目：基于格值逻辑的不确定性推理(60074014),2001.1-2003.12, 参与；
3. 国家自然科学基金项目：基于格值逻辑的语言真值归结自动推理研究(60474022), 2005.1-2007.12, 主研；
4. 校基金项目：基于格蕴涵代数的格值逻辑系统中 MP 归结自动推理的研究（2006B09），2006.9-2008.9，主持。