

# Simplifying Set of Rules and Judging Relative Properties

Yishun Zhang

College of Computer and Information Engineering, Zhejiang Gongshang University, Hangzhou Zhejiang  
Email: zys64@sina.com

Received: Sep. 6<sup>th</sup>, 2016; accepted: Sep. 22<sup>nd</sup>, 2016; published: Sep. 29<sup>th</sup>, 2016

Copyright © 2016 by author and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

---

## Abstract

Determining an equivalent minimal form for a given set of rules is important to build a rule base that consists of the key part of an expert system. Using logic algebra for solving the problem, this paper first transforms the given set of rules into a logic function, then simplifies the logic function by the means of logic algebra and finally convert the result function into a desired simplest set of rules in which each rule is independent. Through looking up special items in all prime implicates obtained in the above procedure, the completeness and contradictoriness of a set of rules can be judged out directly. This method can also be used to detect other problems in rule base without rule-based reasoning.

## Keywords

Expert System, Production Rule, Logic Algebra, Prime Implication

---

# 规则集的化简及相关性质的判定

张亦舜

浙江工商大学计算机与信息工程学院, 浙江 杭州  
Email: zys64@sina.com

收稿日期: 2016年9月6日; 录用日期: 2016年9月22日; 发布日期: 2016年9月29日

---

## 摘要

任给一个规则集, 确定其等价的最简规则集, 此理论问题的解决对实际构建专家系统的核心组成部分规

则库有重要意义。本文运用逻辑代数的基本理论和方法,将给定规则集对应于一个逻辑函数,化简得到最简逻辑函数,其对应所求的最简规则集,其中每条规则具有独立性。通过查找化简过程保留的所有质蕴含中的特定项,还能直接判断规则集的完备性和矛盾性。这种方法无需利用规则进行推理,可以统一运用于检测规则库的其它多种问题。

## 关键词

专家系统,产生式规则,逻辑代数,质蕴含

## 1. 引言

专家系统目前已实际应用在许多领域。知识库是其体系结构的核心部分。产生式规则表示知识简单、自然、有效、便于推理,被普遍采用。由产生式规则构成的知识库也称规则库。

规则主要通过提取,分析,归纳专家的具体领域知识获得。规则的正确简明及在规则库中合理有序的组织存储,保证了专家系统基于规则进行推理的可靠高效。

对于规则库中的大量规则,如果孤立地评判,也许都无可置疑,但作为一个有机的整体,因为规则彼此间复杂紧密的关联,却难免产生各种问题,如冗余,矛盾,循环推理链等等,从而导致专家系统的推理效率降低甚至彻底失效。

这些问题的定义简单明了,但要正确完全地据此做出检测判断却烦琐而困难,往往需要利用规则作大量的有针对性的推理,从推理的过程和结果中做出判别。已提出的各种方法[1]-[6],基于不同领域的理论知识,如经典逻辑[7],Petri网[8][9],图论[10][11]等,虽在问题的转换、表示及相应处理上各有特点,但实质上都只是复制模拟了这些推理过程,因此难以取得切实的突破和改进。

本文探讨一种基于逻辑代数理论处理这些问题的新途径。它无需烦琐的推理,而是利用逻辑代数中质蕴含,最小项等基本概念的特殊性质,通过考察特定的质蕴含和最小项即可直接对各种问题存在与否做出判定。

本文仅作理论上的初步讨论,不针对具体规则库。规则库可简单看作规则的集合,称作规则集。为了讨论方便,我们只考虑最简单的一种规则形式:即命题逻辑中的条件命题。

不妨将规则集以及基于其上的推理作为一个公理系统,集中的每条规则相当于一条公理。数学上要求一个公理系统具有完备性,一致性,独立性。完备性指可以推导出公理系统所描述领域的所有定理,一致性要求不能推导出互相矛盾的结论,独立性指每条公理不能由其它的公理推导出来。作为公理系统的规则集也应该满足这三个条件,体现为规则集中规则所表示知识的完整性,不存在冗余的规则及矛盾的规则或推理规则链。

规则库中的众多问题大多源自于上述完备性,一致性,独立性或显或隐,或多或少的缺失,本文主要讨论利用逻辑代数理论化简规则集及判定规则集的上述性质。

## 2. 预备知识

本节简略介绍所涉及的逻辑代数和规则集的主要概念、记号和结论。详细深入的内容,请参阅[12][13]。

### 2.1. 逻辑代数

逻辑代数用字母代表变量,逻辑变量及逻辑代数的运算结果只有0和1两个取值。0和1不表示数量的大小,只表示对立的两种逻辑状态。

逻辑代数中基本运算类型有三种：与、或、非(也称“反”)。

变量  $A, B$  之间的“与”运算表示为  $AB$ ，即“积”的形式。多个变量的“与”构成一个与项，如  $ABC$ 。

变量  $A, B$  之间的“或”运算表示为  $A+B$ ，即“和”的形式。多个变量的“或”构成一个或项，如  $A+B+C$ 。

“非”是单目运算，变量  $A$  的“非”运算表示为  $\bar{A}$ 。

若逻辑变量  $A_1, A_2, \dots, A_n$  的取值确定以后，变量  $Y$  的值也唯一地确定了，那么就称  $Y$  是  $A_1, A_2, \dots, A_n$  的逻辑函数，记作  $Y = F(A_1, A_2, \dots, A_n)$ 。

对于  $n$  个变量的逻辑函数，如果  $P$  是一个含有  $n$  个变量的与项，每个变量都以原变量或反变量的形式出现一次，且仅出现一次，则称  $P$  是  $n$  个变量的一个最小项，也称标准积。因为每个变量都以原变量和反变量两种可能的形式出现，所以  $n$  个变量可构成  $2^n$  个最小项。

两个相同变量的逻辑函数，如果对于各种不同的变量取值组合，它们的结果都相同，则称这两个逻辑函数等价。

每个逻辑函数都可化为等价的与或式，即“积之和”的形式。每个逻辑函数有且仅有一个等价的由最小项之和构成的与或式，即“标准积之和”，因此等价的逻辑函数有相同的“标准积之和”。如设有三个变量  $A, B, C$ ，与逻辑函数  $A\bar{B} + B\bar{C}$  等价的“标准积之和”为：

$$A\bar{B}\bar{C} + A\bar{B}C + AB\bar{C} + \bar{A}BC \quad (1)$$

利用等价式  $X\bar{A} + XA \Leftrightarrow X$ ，可以化简逻辑函数。在逻辑函数的“标准积之和”中，每个最小项以及所有化简过程中得到的与项都称为该函数的蕴含项。不能再简化的蕴含项称为质蕴含项，如(1)式中  $A\bar{B}, B\bar{C}, A\bar{C}$  为质蕴含项。

化简与或式形式的逻辑函数得到等价的最简与或式，所谓最简指式中没有冗余的与项且每个与项都是质蕴含项。与某个特定逻辑函数等价的最简与或式可能有多个。

## 2.2. 规则集

将规则看作命题逻辑中的条件命题，则基于一个规则集的推理等同于命题逻辑中给定一组命题公式及利用推理规则进行的推理。特别之处在于这组命题公式即规则集中的每条规则有统一的形式

$$X \rightarrow Y$$

上式含义是若  $X$ ，则  $Y$ 。其中  $X$  是条件，称为规则左部， $Y$  是结论，称为右部。单个条件和结论统称为单个命题，一般用字母表中起始的大写字母表示。复合命题由单个命题用逻辑联结词联结而成，一般用字母表中结尾的大写字母表示。逻辑联结词主要有否定( $\neg$ )，析取( $\vee$ )，合取( $\wedge$ )。

用析取符号将单个命题或单个命题的否定联结起来的式子称作析取式。

一个规则集中，只出现于规则左部，不出现于任何规则右部的字母代表初始条件。

若由一组命题公式  $F$  推导出命题公式  $S$ ，则称  $F$  蕴含  $S$ 。

运用等价关系  $X \rightarrow Y \Leftrightarrow \neg X \vee Y$  及其它运算法则，每个规则可以等价转化为若干个析取式的合取形式。每个析取式可等价还原为被原规则蕴含的规则，称为原子规则，如：

$$(A \vee B) \rightarrow C \Leftrightarrow \neg(A \vee B) \vee C \Leftrightarrow (\neg A \wedge \neg B) \vee C \Leftrightarrow (\neg A \vee C) \wedge (\neg B \vee C) \Leftrightarrow (A \rightarrow C) \wedge (B \rightarrow C)$$

$(A \vee B) \rightarrow C$  分解为两条原子规则  $A \rightarrow C$  和  $B \rightarrow C$ 。一个析取式还原后的原子规则可以不同，但都是等价的，如  $\neg A \vee B \Leftrightarrow A \rightarrow B \Leftrightarrow \neg B \rightarrow \neg A$ 。若不重复计数等价的规则，一个析取式与一个原子规则一一对应。

一条规则等价于若干原子规则的集合，任意规则集皆可等价转化为由原子规则组成的集合，这样的集合称为原子规则集。由一个规则集推导出的任何一个原子规则称作规则集的蕴含规则。

一个原子规则集满足下列两个条件称为最简规则集。

- (1) 每条规则没有多余的条件和结论。
- (2) 不存在冗余规则。即可由规则集中其它规则导出的规则。

如规则集  $\{A \rightarrow B, A \wedge B \rightarrow C \vee D, A \wedge B \rightarrow C \vee \neg D\}$  中，由第一条规则，可知第二，三条规则中的条件  $B$  是多余的，其中结论  $D$  和  $\neg D$  也多余，故可简化为  $\{A \rightarrow B, A \rightarrow C\}$ 。规则集  $\{A \rightarrow B, B \rightarrow C, A \rightarrow C\}$  中， $A \rightarrow C$  可由  $A \rightarrow B, B \rightarrow C$  导出，故是冗余规则。

### 3. 等价关系

命题逻辑主要讨论命题公式的构造和推理，逻辑代数侧重研究逻辑函数的构造和化简。虽然符号表示有所不同，但命题逻辑理论与逻辑代数理论在基本定义，定理，运算法则上是等价或同构的。

命题逻辑中的“合取”，“析取”，“否定”分别对应于逻辑代数中的“与”、“或”、“非”。

命题逻辑中有等价关系  $(X \wedge \neg A) \vee (X \wedge A) \Leftrightarrow X \wedge (A \vee \neg A) \Leftrightarrow X$  对应于逻辑代数中用于化简的等价式  $X\bar{A} + XA \Leftrightarrow X$ 。

根据这种等价对应，容易写出一个原子规则集对应的逻辑函数：即每个原子规则对应一个或式，各或式由与运算联结而成的或与式。

为便于书写和处理，以下采用或与式的对偶式与或式形式，此时每个原子规则集对应的逻辑函数是一个与或式，称作规则集的对应与或式，每个原子规则对应式中的一个与式。如规则  $A \rightarrow B$  对应  $A\bar{B}$ 。

由命题逻辑与逻辑代数的等价对应关系，可得到以下重要结论：

**对应定理：**一个规则集的对应与或式中任意一个蕴含项必对应此规则集的一个蕴含规则，反之，规则集的一个蕴含规则必对应此规则集对应与或式中的一个蕴含项，蕴含规则与蕴含项之间存在一一对应。

### 4. 最简规则集

本节研究最简规则集的构造，进而讨论规则集的独立性、完备性和矛盾性的检测判定方法。

#### 4.1. 构造最简规则集

给定一个原子规则集，确定其等价最简规则集显然是一个化简问题，可在逻辑代数领域中求解。

首先由规则集写出对应与或式。运用逻辑代数的方法化简对应与或式。化简过程中存储保留所有最小项和质蕴含。将结果与或式中的每个与项还原为对应的规则便得到最简规则集。

关于化简与或式的实际步骤，因篇幅较长，此不赘述，具体可参阅[12]。依据上节最后的结论，有以下推论。

对应与或式化简前后等价，故对应的前后两个规则集必等价。

化简后的与或式中每个与项都是质蕴含，不可能再简化，故对应的原子规则也不可能再简化，即不可能存在多余的条件或结论。如前例规则集：

$$\{A \rightarrow B, A \wedge B \rightarrow C \vee D, A \wedge B \rightarrow C \vee \neg D\}$$

对应与或式为  $A\bar{B} + ABC\bar{D} + ABCD$ ，化简得  $A\bar{B} + AC$ ，还原得  $\{A \rightarrow B, A \rightarrow C\}$ ，为最简规则集。

#### 4.2. 规则集的独立性

化简结果的每个与项都不可能从其它与项中导出，故对应的原子规则也不可能从其它规则中导出，

是非冗余的。因此，此过程得到的是最简规则集中的每条规则都是独立的。

### 4.3. 规则集的完整性

规则集的不完整指当存在应该推出某结论的条件时，却推不出这一结论，或者推出错误的结论。

利用化简过程求得并保留的所有质蕴含，可简单判定规则集是否完整。

设有前提条件  $X$ ， $X$  不含多余条件，应该推出结论  $Y$ 。不妨假设  $X\bar{Y}$  构成一个与项，否则可分解为若干与项再分别处理。若能由  $X$  推出  $Y$ ，则  $X \rightarrow Y$  是规则集的蕴含规则，依据对应定理和假设， $X\bar{Y}$  是规则集对应与或式的一个蕴含项且是质蕴含，因此只要在所有质蕴含中查看是否存在质蕴含  $X\bar{Y}$  即可判定  $X$  能否推出  $Y$ 。

如有规则集  $\{A \rightarrow B \vee C, A \wedge C \rightarrow B, B \wedge D \rightarrow C, B \rightarrow A \vee D, A \wedge B \rightarrow D\}$ ，判断由条件  $A$  是否可推出结论  $C$  和  $D$ 。对应与或式为  $A\bar{B}\bar{C} + AC\bar{B} + BDC + B\bar{A}\bar{D} + AB\bar{D}$ ，所有质蕴含为  $A\bar{B}, \bar{B}\bar{C}, B\bar{D}, A\bar{C}, A\bar{D}$ ，故由条件  $A$  可推出结论  $C$  和  $D$ 。

### 4.4. 规则集的矛盾性

两条规则或规则链在相同的条件下得到互斥的结论，则称它们是矛盾的。

设条件  $X$  既能推出  $Y$ ，也能推出  $\neg Y$ ，结论矛盾。若规则集不存在循环推理链，那么由某些初始条件  $S$  出发一定可推导出  $X$ ，因此从  $S$  便能推出同样矛盾的结论，故  $S\bar{Y}$  和  $S\bar{Y}$  是规则集对应与或式的蕴含项， $S\bar{Y} + S\bar{Y} = S$  也是蕴含项，若不是质蕴含，则存在质蕴含  $P \subset S$ ，由此若在保留的所有质蕴含中存在仅由初始条件构成的质蕴含，则可断定规则集中存在矛盾。

如有规则集  $\{A \rightarrow B, B \rightarrow C, B \rightarrow \bar{C}\}$ ，其对应与或式为  $A\bar{B} + B\bar{C} + BC$ ，所有质蕴含为  $A, B$ ，初始条件  $A$  构成了一个质蕴含，故规则集存在矛盾。

## 5. 结论

本文研究规则集的化简，通过将规则集转化为对应的逻辑函数，此问题等价于对应逻辑函数的化简。专家系统中所有规则的集合可作为一个公理系统，应该具备完备性，无矛盾性和独立性，通过检查特定的质蕴含，无需利用规则作烦琐的推理，即可以直接判定一个规则集是否满足这些性质。

本文提出的方法还可以运用于检测规则库的其它问题，如冗余，循环规则链等，后续文章中将对对此进一步研究。虽然此法避免了大量的推理，但需要利用各种质蕴含和最小项，实际应用中，质蕴含和最小项的数量可能很庞大，因此如何组织存储所有质蕴含和最小项，以便于查找，也是一个有重要研究价值的问题。

## 参考文献 (References)

- [1] 陈世福, 潘金贵, 徐殿祥. 产生式知识库一致性和冗余性检查[J]. 计算机学报, 1992, 15(9): 670-675.
- [2] 刘书家, 孙名松. 知识库维护技术的研究[J]. 哈尔滨理工大学学报, 1997(1): 33-36.
- [3] 应晶, 吴朝晖. 知识库的一致性问题 and 检查方法[J]. 计算机科学, 1991(2): 63-67.
- [4] 宗成庆, 陈肇雄, 黄河燕. 规则库冗余性控制策略的研究[J]. 软件学报, 1997, 8(1): 1-6.
- [5] 孙运传, 别荣芳. 产生式规则库的求精研究[J]. 北京师范大学学报(自然科学版), 2003, 39(4): 435-443.
- [6] Knauf, R., Philippow, I. and Gonzalez, A.J. (2000) Towards Validation and Refinement of Rule-Based Systems. *Journal of Experimental & Theoretical Artificial Intelligence*, **12**, 421-431. <http://dx.doi.org/10.1080/095281300454801>
- [7] 王永庆. 人工智能原理与方法[M]. 西安: 西安交通大学出版社, 1998.
- [8] 栾尚敏, 戴国忠. 命题规则知识库更新的一种代数方法[J]. 中国科学 E 辑: 信息科学, 2008, 38(2): 177-194.

- 
- [9] 姜浩, 罗军舟, 方宁生. 一种基于有色 Petri 网的知识库验证方法[J]. 东南大学学报, 2000, 30(1): 77-83.
- [10] Ramaswamy, M., Sarkar, S. and Chen, Y.-S. (1997) Using Directed Hypergraphs to Verify Rule-Based Expert Systems. *IEEE Transactions on Knowledge and Data Engineering*, **9**, 221-237. <http://dx.doi.org/10.1109/69.591448>
- [11] 孙伟, 郭莉, 高天一, 等. 一种基于有向超图的规则库冗余及环路检测方法[J]. 大连理工大学学报, 2008, 48(1): 74-78.
- [12] 陈光梦. 数字逻辑基础[M]. 上海: 复旦大学出版社, 2007.
- [13] 左孝凌. 离散数学[M]. 上海: 上海科学技术文献出版社, 1982: 2-80.

**期刊投稿者将享受如下服务:**

1. 投稿前咨询服务 (QQ、微信、邮箱皆可)
2. 为您匹配最合适的期刊
3. 24 小时以内解答您的所有疑问
4. 友好的在线投稿界面
5. 专业的同行评审
6. 知网检索
7. 全网络覆盖式推广您的研究

投稿请点击: <http://www.hanspub.org/Submission.aspx>

期刊邮箱: [csa@hanspub.org](mailto:csa@hanspub.org)