

Creating a Reasoning Chain Determinately

Yishun Zhang

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

Received: Sep. 5th, 2016; accepted: Sep. 22nd, 2016; published: Sep. 29th, 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

Reasoning is a research focus of artificial intelligence. Uncertainty in reasoning process generally produces many redundant reasoning branches. Based on production rule commonly used in expert system, this paper proposes a method to build the reasoning chain determinately. The method uses the theory of logic algebra. First we establish correspondence between a set of production rules and a logical function, and then prove that rules in a reasoning chain corresponding to a special minimal subset of the prime implication set that consists of the logical function, finally give out the basic algorithm to determine the subset. A sound reasoning chain is formed by arranging rules corresponding to prime implications in subset according to the logical order.

Keywords

Expert System, Reasoning, Production Rule, Logic Algebra, Prime Implication

推理规则链的确定性构建

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5日; 录用日期: 2016年9月22日; 发布日期: 2016年9月29日

摘要

推理是人工智能领域研究的重点。推理过程一般是非确定的, 会产生许多冗余的推理分支。本文针对专

家系统中常用的产生式规则，提出了一种确定性构造推理链的方法。方法运用逻辑代数理论，首先确立了规则集与逻辑函数的对应关系，证明推理规则链中的所有规则对应构成逻辑函数质蕴含集中的一个特定最小子集，给出了确定该子集的基本算法。子集中质蕴含对应的规则按逻辑顺序排列即构成了一条合理的推理规则链。

关键词

专家系统，推理，规则，逻辑代数，质蕴含

1. 引言

推理是人工智能领域的一个研究重点。无论是建立在经典逻辑基础上的结论确定的推理，如自然演绎推理，归结演绎推理[1]，还是基于模糊性的知识导致结论不确定的推理[2]，如近似推理[3]，模糊推理[4] [5]，正向推理[6]，反向推理[7]，推理过程都是非确定性的，需要试探前行，难免会产生各种多余的推理分支，影响推理效率。

本文提出的基于逻辑代数理论构建推理链的方法，构建过程不必模拟实际的推理过程，不会产生冗余的推理分支，因此能确定性地构建从初始条件到最终结论的所有推理链。本文仅在理论上针对产生式规则集做了初步探讨，且采用产生式规则(以下简称规则)最简单直观的一种类型或形式，记为 $P \rightarrow Q$ ，意思是“若P则Q”。

此方法首先建立规则集与逻辑函数的对应。运用逻辑代数方法化简对应逻辑函数，最简逻辑函数对应最简规则集。化简过程中，构造两个集合P1和P2，P1是构成最简逻辑函数的所有质蕴含，对应最简规则集，P2是所有包含初始条件和最终结论的质蕴含，对应从初始条件出发运用最简规则集中的规则可推导出的最终结论。给定初始条件，在P2中找出仅包含给定初始条件和最终结论的质蕴含p，再在P1中确定一个覆盖p的最简子集，一个最简子集对应的规则即是一条最简推理链上使用的所有规则，最后将这些规则按逻辑顺序排列便得到了一条合理的最简推理规则链，最简子集可能有多个，对应多条最简推理规则链。

2. 预备知识

2.1节简略列出了涉及到的逻辑代数的有关基本知识，详细内容可参考有关教材[8]。2.2讨论了规则集与逻辑函数的关系。

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，如真与假、对与错。

定义1 逻辑代数中基本运算类型有三种：与(\cdot)、或($+$)、非($-$)，定义见表1。其他的运算皆可等价地转化为这三种运算。

变量 x, y 之间的“与”运算表示为 $x \cdot y$ ， \cdot 通常省略，简写为 xy ，即“积”的形式。多个变量的“与”构成一个与项，如 xyz 。

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

“非”是单目运算，变量 x 的“非”运算表示为 \bar{x} 。称 x 与 \bar{x} 为互反的。

定义2 若逻辑变量 x_1, x_2, \dots, x_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 x_1, x_2, \dots, x_n 的逻辑函数，记作 $y = f(x_1, x_2, \dots, x_n)$ 。

Table 1. Definition of and, or, not
表 1. 与, 或, 非定义

x	y	xy	$x+y$	\bar{x}
0	0	0	0	1
0	1	0	1	1
1	0	0	1	0
1	1	1	1	0

定义 3 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。

定义 4 两个相同变量的逻辑函数 f, g , 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, 记作啊 $f \Leftrightarrow g$ 。

定理 1 常用等价式有:

$$x(y+z) \Leftrightarrow xy+xz \quad (1)$$

$$x+x \Leftrightarrow x+xy \Leftrightarrow x \quad (2)$$

$$x \Leftrightarrow x(y+\bar{y}) \Leftrightarrow xy+x\bar{y} \quad (3)$$

$$\overline{x+y} \Leftrightarrow \bar{x}\bar{y} \quad (4)$$

$$\overline{xy} \Leftrightarrow \bar{x}+\bar{y} \quad (5)$$

依据等价式(1) (4) (5)每个逻辑函数都可化为等价的与或式, 即“积之和”的形式。以下讨论默认逻辑函数具有此种形式。由(3), 每个与项可化为等价的最小项之和, 这些最小项被与项和逻辑函数所包含或覆盖, 因此“积之和”可化为等价的最小项之和, 即“标准积之和”。

定理 2 每个逻辑函数的“标准积之和”有且仅有一个。等价的逻辑函数有相同的“标准积之和”。如设有三个变量 x, y, z , 与逻辑函数 $x\bar{y}+y\bar{z}$ 等价的“标准积之和”为:

$$x\bar{y}(z+\bar{z})+y\bar{z}(x+\bar{x}) \Leftrightarrow x\bar{y}z+x\bar{y}\bar{z}+xy\bar{z}+\bar{x}y\bar{z}$$

反之, 由(3)两个仅有一个变量互反, 其余变量相同的与项可消去互反变量, 合并为一个与项。

定义 5 逻辑函数的“标准积之和”中, 每个最小项以及所有运用(3)合并化简得到的与项都称为该函数的蕴含项。不能再合并化简的蕴含项称为质蕴含项, 如上式的质蕴含项为 $x\bar{y}z, y\bar{z}, \bar{x}\bar{y}z$ 。

定理 3 两个包含互反变量的蕴含项没有共同的最小项。两个蕴含项共同包含的蕴含项是这两个蕴含项相与并消去重复。如: $x\bar{y}z, x\bar{y}\bar{v}$, 其共同蕴含项为 $x\bar{y}z\bar{v}$ 。

定理 4 一个蕴含项 $x_1x_2\cdots x_n$ 除去其覆盖的一个蕴含项 $x_1x_2\cdots x_nx_{n+1}x_{n+2}\cdots x_m$ 后, 剩余部分写成最简逻辑函数为 $x_1x_2\cdots x_n(\overline{x_{n+1}+x_{n+2}+\cdots+x_m}) \Leftrightarrow x_1x_2\cdots x_n\overline{x_{n+1}}+x_1x_2\cdots x_n\overline{x_{n+2}}+\cdots+x_1x_2\cdots x_n\overline{x_m}$

上面每个 $x_i (1 \leq i \leq m)$ 表示不同变量或变量的反。

定义 6 最简逻辑函数指函数式中没有冗余的与项且每个与项都是质蕴含项。

化简给定逻辑函数可得到等价的最简逻辑函数, 等价的最简逻辑函数可能有多组。

定义 7 一个最简逻辑函数的蕴含项包含的所有最小项若只被最简逻辑函数中的一个质蕴含覆盖, 称此蕴含项为单属蕴含项。

定理 5 若逻辑函数的一个最小项是单属蕴含项, 则包含它的质蕴含必出现在最简逻辑函数中。

定义 8 将逻辑函数 f 中的 $+$ 换成 \cdot , \cdot 换成 $+$, \bar{x} 换成 x , x 换成 \bar{x} 得到的式子 f^* 称为 f 的对偶, 显然 f 与 f^* 互为对偶。

由等价式(4) (5), 对偶函数之间存在等价关系 $\bar{f} \Leftrightarrow f^*$ 。因此一个逻辑函数可方便地转换成其对偶函数来处理。

2.2. 规则与规则集

形如 $x \rightarrow y$ 的规则, 左部 x 为条件, 右部 y 为结论。其运算定义等价于 $\bar{x} + y$, 见表 2, 为了书写和处理方便, 以下讨论采用 $\bar{x} + y$ 的对偶项 $x\bar{y}$ 。

定义 9 类似 $(x + y) \rightarrow vw$ 的规则, 因有等价关系 $(x + y)\overline{vw} \Leftrightarrow (x + y)(\bar{v} + \bar{w}) \Leftrightarrow x\bar{v} + x\bar{w} + y\bar{v} + y\bar{w}$, 可以将其等价分解为几条规则, 使得每条规则的左部是一个与项, 右部是一个或项。这种形式的规则称为原子规则。

原子规则和与项之间存在一一对应。若非特别声明, 以下出现的规则皆是原子规则。

一条规则可等价地写成不同形式, 但它们都对应同一个与项。如 $xy \rightarrow z$ 与 $x \rightarrow (\bar{y} + z)$, $y \rightarrow (\bar{x} + z)$, $\bar{z} \rightarrow (\bar{x} + \bar{y})$, $x\bar{z} \rightarrow \bar{y}$, $y\bar{z} \rightarrow \bar{x}$, 等相互等价, 都对应与项 $xy\bar{z}$ 。

定义 10 一个推理系统如果基于一个规则集, 则集中不出现在任何规则右部的条件称为初始条件, 不出现在任何规则左部的结论称为最终结论。

定义 11 给定一个规则集, 将其中每条规则对应的与项用或运算连接得到的式子称为规则集的逻辑函数。

定理 6 规则集 A 的逻辑函数的任一蕴含项皆对应一条被 A 蕴含的规则, 即能够由 A 推导出的规则。

证明: 设集中有规则 $R: x_1x_2 \cdots x_n \rightarrow y_1 + y_2 + \cdots + y_m$, 其对应的与项为 $x_1x_2 \cdots x_n\bar{y}_1\bar{y}_2 \cdots \bar{y}_m$, 此与项包含的每个最小项必含有 $x_1x_2 \cdots x_n\bar{y}_1\bar{y}_2 \cdots \bar{y}_m$, 设 $x_1x_2 \cdots x_nz_1z_2 \cdots z_r\bar{y}_1\bar{y}_2 \cdots \bar{y}_m$ 是其中任意一个最小项, 解释为规则 $R_1: x_1x_2 \cdots x_nz_1z_2 \cdots z_r \rightarrow y_1 + y_2 + \cdots + y_m$, 显然由规则 R 可以导出规则 R_1 , 故 A 的逻辑函数包含的所有最小项皆对应一条被 A 蕴含的规则。

逻辑函数的一个蕴含项是从最小项开始, 逐次利用等价式(3)合并两个与项得到的。由 $xy\bar{z} + x\bar{y}\bar{z} \Leftrightarrow x\bar{z}$, 解释为规则 $xy \rightarrow z$ 和 $x\bar{y} \rightarrow z$ 等价于规则 $x \rightarrow z$, 此显然成立。故 A 的逻辑函数的任意蕴含项皆对应一条被 A 蕴含的规则。

定义 12 一个规则集中若不存在冗余规则且每条规则最简(没有多余的条件和结论), 则称其为最简规则集。

定理 7 最简规则集的逻辑函数是最简逻辑函数。

证明: 设最简规则集 R_s 的逻辑函数 f 不是最简的, 则或者:

1) f 中有冗余与项 p , f 除去 p 后, p 仍然是 f 的蕴含项, 由定理 6, R_s 中除去 p 对应的规则 r_p 后, r_p 仍可由规则集推导出来。故 R_s 不是最简规则集。

2) f 中有非质蕴含项 q , q 对应规则 r , q 化简为质蕴含 q' , 由定理 6, q' 对应的规则 r' 可由 R_s 导出。

Table 2. Definition of \rightarrow & equivalent form

表 2. \rightarrow 的定义及等价式

x	y	$x \rightarrow y$	$\bar{x} + y$
0	0	1	1
0	1	1	1
1	0	0	0
1	1	1	1

R_s 中除去 r , 加入 r' , 得到规则集 R'_s , 显然 R'_s 可推出 r , 因此 R_s 与 R'_s 等价, 但 R_s 中的 r 不是最简规则, 故 R_s 不是最简规则集。

综上, f 是最简逻辑函数。

规则集的化简对应规则集逻辑函数 f 的化简。逻辑函数的化简常运用 Quine-McCluskey 方法。该方法先求出 f 所有的质蕴含, 再从其中确定构成最简逻辑函数的质蕴含。在化简过程求出的所有质蕴含中, 设确定构成最简逻辑函数的质蕴含集合记为 P_1 , 仅包含初始条件和最终结论的质蕴含集合记为 P_2 。

3. 构建推理链

有最简规则集 R_s , 若初始条件 $X = \{x_1, x_2, \dots, x_n\}$ 恰好可推出最终结论 $Y = y_1 + y_2 + \dots + y_m$, 则规则集的逻辑函数包含质蕴含 $x_1 x_2 \dots x_n \bar{y}_1 \bar{y}_2 \dots \bar{y}_m$, 简写为 $X\bar{Y}$, $X\bar{Y}$ 属于 P_2 。若初始条件不充分, 则可从 P_2 中找出初始条件与 X 相近的质蕴含, 并判断能否补充缺失的初始条件。在实际应用中, 就此可以有针对性地提问, 补充事实和条件。

如何构造从 X 到 Y 的推理链? 由定理 6, 推理链上的规则对应的质蕴含一定覆盖了 $X\bar{Y}$, 据此在 P_1 中先确定这些质蕴含, 再将质蕴含对应的规则按逻辑顺序先后排列便构成了推理规则链。下面给出基本算法。

1) 确定 P_1 中每个质蕴含包含的所有最小项。

2) 构造两张表格: 单属蕴含项表, 记录 P_1 中每个质蕴含包含的所有单属蕴含项。质蕴含表, 记录规则集的逻辑函数的每个非单属最小项及 P_1 中包含它的所有质蕴含。

3) 初始化, U 是已确定的质蕴含的集合, 初始为空。 V 是 $X\bar{Y}$ 包含的尚未被 U 中质蕴含所覆盖的蕴含项集合, 初始为 $X\bar{Y}$ 。

4) 扫描单属蕴含项表, 确定所有这样的质蕴含, 其单属蕴含项与 $X\bar{Y}$ 包含有相同的最小项。这些质蕴含对应的规则必出现在推理规则链中。因此将这些质蕴含加入 U 中。修改 V , 从 $X\bar{Y}$ 删去被 U 中质蕴含包含的蕴含项。

5) 经过步骤 4, V 中的蕴含项包含的所有最小项都被 P_1 中的两个或两个以上质蕴含所覆盖。查询质蕴含表, 对 V 中每个最小项将包含它的所有质蕴含相或, 最后将每个或项相与, 构成一个或与式, 然后展开化简得到与或式, 对与或式中每个与项, 其代表的质蕴含加入 U 即构成一个最终结果, 因此与或式中的与项数就决定了 U 的最终结果数。

6) 将最终结果 U 中的所有质蕴含对应的规则按逻辑顺序排列即构成了一条合理的推理规则链。 U 的最终结果数就是最终推理链的数目。

显然, 算法中单属蕴含项表和质蕴含表可事先构造完成。因算法中的运算操作较繁琐复杂, 限于篇幅, 下面只给出一个简单的实例

例: 最简规则集为 $\{x \rightarrow y, y \rightarrow z, x \rightarrow v\}$, x 是初始条件, z, v 是最终结论。

$$P_1 = \{x\bar{y}, y\bar{z}, x\bar{v}\}, \quad P_2 = \{x\bar{z}, x\bar{v}\}.$$

给出初始条件 x , 在 P_2 中确定符合初始条件的质蕴含

$x\bar{z}, x\bar{v}$, 因此, 可推导两个结论 z, v 。仅以 $x\bar{z}$ 为例。

确定 P_1 中每个质蕴含包含的所有最小项如下:

$$p_1 = x\bar{y} \Leftrightarrow x\bar{y}z\bar{v} + x\bar{y}z\bar{v} + x\bar{y}z\bar{v} + x\bar{y}z\bar{v}$$

$$p_2 = y\bar{z} \Leftrightarrow xy\bar{z}\bar{v} + xy\bar{z}\bar{v} + xy\bar{z}\bar{v} + xy\bar{z}\bar{v}$$

$$p_3 = x\bar{v} \Leftrightarrow xyz\bar{v} + xy\bar{z}\bar{v} + x\bar{y}z\bar{v} + x\bar{y}z\bar{v}$$

构造单属蕴含项表, 质蕴含表分别见表 3, 表 4。

初始化: $U = \phi, V = \{x\bar{z}\}$

Table 3. Implications contained by only one prime implication
表 3. 单属蕴含项

质蕴含	单属蕴含项
$x\bar{y}$	$x\bar{y}v$
$y\bar{z}$	$\bar{x}y\bar{z}, y\bar{z}v$
$x\bar{v}$	$xyz\bar{v}$

Table 4. Prime implications
表 4. 质蕴含

最小项	质蕴含
$\bar{x}y\bar{z}v$	$x\bar{y}, x\bar{v}$
$xy\bar{z}v$	$y\bar{z}, x\bar{v}$
$\bar{x}y\bar{z}v$	$x\bar{y}, x\bar{v}$

扫描表 3, $x\bar{y}$ 的单属蕴含项 $x\bar{y}v$ 与 $x\bar{z}$ 有相同的最小项 $\bar{x}y\bar{z}v$,

将 $x\bar{y}$ 加入 U , 得 $U = \{x\bar{y}\}$, V 修改为 $V = \{xy\bar{z}\}$,

$y\bar{z}$ 的单属蕴含项 $y\bar{z}v$ 与 $x\bar{z}$ 有相同的最小项 $xy\bar{z}v$, 将 $y\bar{z}$ 加入 U , 得 $U = \{x\bar{y}, y\bar{z}\}$, V 修改为 $V = \phi$ 。

因 V 为空, U 已得到最后结果, 不必继续判断表 3 中的 $x\bar{v}$, 即使先考察 $x\bar{v}$, 会发现 $x\bar{v}$ 的单属蕴含项与 $x\bar{z}$ 没有共同的最小项。

将 U 中质蕴含对应的规则按逻辑先后顺序排列即有规则链 $x \rightarrow y, y \rightarrow z$ 。

4. 结论

本文针对简单形式的产生式规则集提出了一种非试探性的确定性地构造推理规则链的方法。该方法将规则集转化为逻辑函数, 利用规则集蕴含的规则与逻辑函数的蕴含项之间的等价对应, 从构成最简逻辑函数的质蕴含集中确定一个特定的最简子集, 其中的元素对应规则链中的规则, 这些对应规则依逻辑顺序排列即构成了一条合理的最简推理规则链。

本文仅是理论上的初步探讨, 所述方法只适用于规则为命题逻辑中条件命题这一简单形式的规则集, 难以推广到一般情形。限于篇幅, 文中许多细节未深入讨论, 如表格的构造, 查询的方法和效率, 逻辑表达式的快速展开和化简等。

参考文献 (References)

- [1] 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.
- [2] 宋光雄, 何永勇, 褚福磊. 基于双参数方法的故障诊断不确定性推理问题[J]. 清华大学学报, 2006, 46(8): 1397-1400.
- [3] 王永县, 朱涛, 李飞. 基于相似关系的广义近似推理方法[J]. 清华大学学报, 2002, 42(10): 1285-1288.
- [4] 陈晨, 周勇. 一种基于置信规则的模糊推理算法[J]. 电子科技, 2013, 26(4): 82-85.
- [5] Yeung, D.S., Wang, X.Z. and Tsang, E.C. (2004) Handling Interaction in Fuzzy Production Rule Reasoning. *IEEE Transactions on Systems, Man, & Cybernetics. Part B: Cybernetics*, **34**, 1979-1987.
- [6] 崔奇明. 基于 WEB 的非精确正向推理专家系统的研究与应用[J]. 电脑与信息技术, 2006, 14(3):46-49.
- [7] 孙臣良, 孙福玉, 梁宏友. 基于规则的非精确反向推理专家系统[J]. 辽宁工程技术大学学报, 2003, 22(s1): 145-147.
- [8] 陈光梦. 数字逻辑基础[M]. 上海: 复旦大学出版社, 2007.

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