

# The Algorithms about Validity of Argument Form in Logic Mathematician

Qiaoyun Liu

Qinghai Nationalities University, Xining Qinghai  
Email: qiaoyun2013@qq.com

Received: Apr. 14<sup>th</sup>, 2019; accepted: Apr. 25<sup>th</sup>, 2019; published: May 6<sup>th</sup>, 2019

---

## Abstract

It discusses the four methods to judge the validity of argument form in logic mathematician in this paper. In the end it gives the algorithms of judging validity of argument form.

## Keywords

Validity of Argument Form, Truth Table, Tautology, Induction Rules

---

# 数理逻辑论证形式有效性算法

刘巧云

青海民族大学, 青海 西宁  
Email: qiaoyun2013@qq.com

收稿日期: 2019年4月14日; 录用日期: 2019年4月25日; 发布日期: 2019年5月6日

---

## 摘要

本文讨论了数理逻辑中论证形式有效性的四种判断方法并给出了一种判断有效性的计算机算法。

## 关键词

论证形式有效性, 真值表, 重言式, 推理规则

---

Copyright © 2019 by author(s) 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

## 1. 引言

数理逻辑，是由推理组成的。我们假定只讨论或真或假的句子，把这种句子称为简单句或复合句。

当复合句符号化时，得到了纯语句形式，几个不同的语句可以有共同的语句形式。因此，我们研究语句形式而不是语句含义，从而讨论语句形式的有效性判断方法十分必要。字母  $p, q, r, \dots$  是语句变元，表示任意没有指明的简单语句。

## 2. 定义

### 1) 论证形式定义[1]

通常，一个论证形式是语句形式的有穷序列，序列最后一项称为结论，其余项称为前提。

### 2) 论证形式有效性定义[1]

论证形式  $\mathcal{A}_1, \mathcal{A}_2, \dots, \mathcal{A}_n$ ； $\therefore \mathcal{A}$  是无效的，如果对出现的语句变元的真值指派有可能使  $\mathcal{A}_1, \mathcal{A}_2, \dots, \mathcal{A}_n$  取值 1 而  $\mathcal{A}$  取值 0；否则论证形式是有效的。

## 3. 论证形式有效性判断方法

### 1) 用真值表判断

方法[1]：利用定义，对作为前提或结论出现的所有语句形式构造真值表，若所有前提为真时的语句变元赋值使得结论也为真，则说明此论证形式有效。否则，此论证形式无效。

例 1：如果今天周日，那么明天是周一。事实上今天是周日。从而，明天是周一。

解：设  $p$ ：今天是周日； $q$ ：明天是周一。此论证形式为： $p \rightarrow q, p; \therefore q$ 。写出真值表如下：

$p$	$\rightarrow$	$q$	$p$	$q$
1	1	1	1	1
1	0	0	1	0
0	1	1	0	1
0	1	0	0	0

前提都取值 1 的行只有第 1 行，结论也取值 1，从而此论证形式有效。

### 2) 用反证法判断

从上述例子可以看到，如果不同的语句变元数目很多，真值表将变得很庞大，从而是不切实际的。一般情况下，我们并不需要整个真值表就可得到有效或者无效的目的。

方法[1]：以有规则的方法寻找特殊类型的行，而不用构造整个真值表的带有试探性的易出错误的方法。具体的方法用例 2 说明。

例 2：对于春秋时期的某国，如果仓禀实或衣食足，则民知礼节或者知荣辱。如果民知礼节或者知荣辱，则或者国富，或者民强。如果民强，则百业兴。事实上该国并非国富，而且并非百业兴。从而，该国并非仓禀实且并非衣食足。

解：设 p: 国仓稟实; q: 国衣食足; r: 民知礼节; s: 民知荣辱; t: 国富; u: 民强; v: 国百业兴。  
从而写出论证形式为:  $(p \vee q) \rightarrow (r \vee s), (r \vee s) \rightarrow (t \vee u), u \rightarrow v, \sim t \wedge \sim v; \therefore \sim p \wedge \sim q$ 。

假设结论  $\sim p \wedge \sim q$  为假, 则 p 与 q 必有一个为真。此时,  $p \vee q$  必为 1, 前提  $(p \vee q) \rightarrow (r \vee s)$  为真时,  $r \vee s$  值也应该取 1。此时  $(r \vee s) \rightarrow (t \vee u)$  要为真时,  $t \vee u$  值应该取 1; 前提  $\sim t \wedge \sim v$  取真, t 与 v 都应该取假; t 为假时,  $t \vee u$  为真, 从而 u 为真。u 为真, v 为假, 从而  $u \rightarrow v$  值为 0。从而, 此论证形式有效。

### 3) 用重言式判断

方法[1]: 论证形式  $\mathcal{A}_1, \mathcal{A}_2, \dots, \mathcal{A}_n; \therefore \mathcal{A}$  是有效的当且仅当命题形式  $(\mathcal{A}_1 \wedge \mathcal{A}_2 \wedge \dots \wedge \mathcal{A}_n) \rightarrow \mathcal{A}$  是重言式。

例 3: 如果 a 为整数, 则它为有理数; 如果 a 为有理数, 则它为实数。所以, 如果 a 是整数, 则它为实数。

解: 设 p: a 为整数; q: a 为有理数; r: a 为实数。从而写出论证形式为:  $p \rightarrow q, q \rightarrow r; \therefore p \rightarrow r$ 。  
写出语句形式  $((p \rightarrow q) \wedge (q \rightarrow r)) \rightarrow (p \rightarrow r)$  的真值表。

(p	→	q)	∧	(q	→	r))	→	(p	→	r)
1	1	1	1	1	1	1	1	1	1	1
1	1	1	0	1	0	0	1	1	0	0
1	0	0	0	0	1	1	1	1	1	1
1	0	0	0	0	1	0	1	1	0	0
0	1	1	1	1	1	1	1	0	1	1
0	1	1	0	1	0	0	1	0	1	0
0	1	0	1	0	1	1	1	0	1	1
0	1	0	1	0	1	0	1	0	1	0

从真值表可以看出不论变元如何赋值, 最后的语句都取值 1。  $((p \rightarrow q) \wedge (q \rightarrow r)) \rightarrow (p \rightarrow r)$  是重言式, 所以  $p \rightarrow q, q \rightarrow r; \therefore p \rightarrow r$  有效。

### 4) 用推理规则判断

方法[2]: 从任意给定的前提出发, 应用相应的推理规则进行推演, 得到的结论在系统中被认为是有效的。推理规则有前提引入规则、结论引用规则、置换规则、加法式、简化式、假言推理、拒取式、析取三段论、前提三段论、合取式、构造性二难推理和破坏性二难推理。

例 4: 小王是工程师或者教师; 如果小王是工程师, 那么小李是医生; 如果小王是教师, 那么小李是医生; 小王是教师。所以, 小王是教师且小李是医生。

解: 设 p: 小王是工程师; q: 小王是教师; r: 小李是医生。此论证形式为  $p \vee q, p \rightarrow r, q \rightarrow r, q; \therefore q \wedge r$ 。

- i)  $p \vee q$             前提引入
- ii)  $p \rightarrow r$         前提引入
- iii)  $q \rightarrow r$         前提引入
- iv) r                (1)(2)(3)构造性两难
- v) q                 前提引入
- vi)  $q \wedge r$         合取式

## 4. 论证形式有效性算法

从上述例子可以看出, 利用真值表和重言式来验证论证形式有效性十分繁琐, 遇到语句变元较多情

况时，花费的时间较多。用反证法容易出错。计算机程序设计越来越普遍，计算机计算节省了很多时间。下面以例题的方式列出论证形式有效性程序：

例 5：如果生产下降或浪费严重，那么将造成物资匮乏。如果物资匮乏，那么或者物价暴涨，或者人民生活贫困。如果人民生活贫困，政府将失去民心。事实上，物价没有暴涨，而且政府赢得了民心。由此可见生产没有下降并且没有浪费严重。

解：设 p：生产下降；q：浪费严重；r：物资匮乏；s：物价暴涨；t：人民生活贫困；u：政府将失去民心。此论证形式为： $(p \vee q) \rightarrow r, r \rightarrow (s \vee t), t \rightarrow u, \sim s \wedge \sim u; \therefore \sim p \wedge \sim q$ 。利用 java 程序计算，最后得此论证形式有效。程序如下：

```
import java.lang.Math;
public class love {
public static int YH(int p, int q){
    if(p == 1 & q == 0){
        return 0;}
    else{return 1;}}//函数YH为蕴含函数定义;
public static int Y(int array[][],int n,int m){
    int cou = 1;int geshu = 0;
    int p = 0, q = 0,r = 0, s = 0, t = 0, u = 0;
    for(int i = 0; i < n; i ++){
        for(int j = 0;j <= m; j ++){
            switch(j){
                case 0:
                    p = array[i][j];
                    break;
                case 1:
                    q = array[i][j];
                    break;
                case 2:
                    r = array[i][j];
                    break;
                case 3:
                    s = array[i][j];
                    break;
                case 4:
                    t = array[i][j];
                    break;
                case 5:
                    u = array[i][j];//变元不够时可再进行添加;
            default:
                break;}}
    int y = YH(YH(p|q,r)&YH(r,s|t)&YH(t,u)&((~s)&(~u)),((~p)&(q)));//需要验证的论证形式改写
```

用第三种方法写出的语句形式;

```

    array[i][0] = y;
    if(y == 0) cou = 0;}
for(int x = 0; x < array.length / 2; x++){
    int temp = array[x][0];
    array[x][0] = array[array.length-1-x][0];
    array[array.length-1-x][0] = temp;}
for(int i = 0; i < array.length; i++){
    if(array[i][0]==0){
        System.out.print("\n此论证形式无效。");break;}//只要遇到赋值时结果为0, 此语句形式
不是重言式, 此论证形式无效;
    else{geshu ++;}}
    if(geshu==n){System.out.print("\n此论证形式有效。");};//所有赋值结果都为1时, 此语句为重言式,
论证形式有效;
    return 0;}
public static void main(String args[]){
    String str = "YH(YH(p,q,r)&YH(r,s,t)&YH(t,u)&((~s)&(~u)),((~p)&(q)))";//与int y保持一致;
    char[] cha1 = new char[str.length()];
    char[] cha2 = new char[str.length()];
    for(int i =0; i < str.length(); i++){
        cha1[i] = str.charAt(i);}
    int count = 0;
    for(int j = 0; j < str.length(); j++){
        if(((int)(cha1[j]) >= 97 & (int)(cha1[j]) <= 122)){
            for(int k = j+1; k < str.length(); k++){
                if(cha1[j] == cha1[k]){
                    cha2[j] = cha1[j];}}//计算小写字母个数;
            else{
                if(((int)(cha1[j]) > 122) | ((int)(cha1[j]) < 97)){
                    cha2[j] = cha1[j];}}//计算其它字符个数;
        }
    }
    for(int i =0; i < cha2.length; i++){
        if((int)(cha2[i]) != 0){
            count ++;} //计算重复的小写字母个数;
    }
    count = (int) (str.length()) - count;//计算输入的语句形式有几个变元;
    //System.out.print("\n此论证形式变元个数为: "+count+"\n");此处可输出;
    int[][] cha3 = new int[(int)(Math.pow(2,count))][count+1];
    for(int i = (int)(Math.pow(2,count))-1; i >= 0; i --){
        int s = i;
        int t = i;
        while(s != 0){

```

```

        for(int j = 0;j < count;j++){
            cha3[t][j] = s % 2;
            s = s / 2;}}//采用变元赋值时每次十进制多1的方法，将十进制改写成2进制形式给
变元赋值；

            //System.out.print(" "+cha3[t][j]);//输出每个变元的赋值；可输出
            //System.out.println();}}
        for(int j = 0;j < count;j++){
            cha3[0][j] = 0;}
            //System.out.print(" "+cha3[0][j]);//输出每个变元的赋值；可输出
            System.out.println();
        int y = Y(cha3,(int)(Math.pow(2,count)),count+1);} //调用 Y 函数计算最后结果。

```

## 基金项目

青海民族大学探究课程项目，项目编号：2018-TJKC-01。

## 参考文献

- [1] Hamilton, A.G. Logic for Mathematicians (Revised Edition). 北京: 清华大学出版社, 2003.
- [2] 王礼萍, 等. 离散数学简明教程[M]. 北京: 清华大学出版社, 2005.

### 知网检索的两种方式:

1. 打开知网页面 <http://kns.cnki.net/kns/brief/result.aspx?dbPrefix=WWJD>  
下拉列表框选择: [ISSN], 输入期刊 ISSN: 2160-7583, 即可查询
2. 打开知网首页 <http://cnki.net/>  
左侧“国际文献总库”进入, 输入文章标题, 即可查询

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

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