# 一个新的证明网定义及合理性*

黄林鹏 孙永强

(上海交通大学计算机系,上海 200030)

**摘要** 本文给出一个新的线性逻辑的证明网的定义并证明了所定义的证明网是线性逻辑的自然推理. 和 Girard 的原定义相比,使用本文给出的定义来判定一个证明结构是否为证明网的时间复杂度为 $O(n*n)$,并且在证明所定义的证明网是可矢列化时更加自然和简单.

**关键词** 线性逻辑,证明网,并行计算.

线性逻辑是法国 Girard 教授[1]发展起来的一种试图在逻辑层次上回答并行计算问题的新型逻辑系统.

证明网是线性逻辑中最新颖而重要的一个概念.

从直觉主义逻辑自然推理系统 NJ 的规则⇒I 和⇒E,Girard 诱导出规则+(par−link): $\frac{A\quad B}{A+B}$和规则×(times−link): $\frac{A\quad B}{A\times B}$, 上述两规则加上规则axiom−link: $\frac{}{A\quad A^{\perp}}$和 cut−link: $\frac{A\quad A^{\perp}}{}$构造了线性逻辑的证明结构.

由于规则+和×是形式上一样的局部规则,因此我们必须有一全局条件来保证或验证由上述规则构造的证明结构是逻辑正确的,即其存在和线性矢列演算之间的某一对应关系. Girard 在[1]中给出一个 longtrip 条件,其时间复杂度是指数阶的.

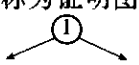
本文我们给出一个算法,并在此基础上重新定义了证明网的概念,证明了所定义的证明网是线性逻辑的自然推理,我们分析了算法的时间复杂度.

## 1 算法和定义

算法分为两步:抽象和染色

### 1.1 抽象

对证明结构进行抽象,结果为一有向图,称为证明图. 对证明结构的抽象可以图示如下:

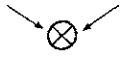
(a) axiom−link $\frac{}{A\quad A^{\perp}}$抽象为 


---

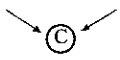
* 本文 1991−12−12 收到,1992−06−13 定稿

作者**黄林鹏**,30 岁,1992 年博士毕业于上海交通大学,主要研究领域为人工智能,并行计算,计算机逻辑. **孙永强**,63 岁,教授,博士导师,主要研究领域为新型语言,计算理论.

本文通讯联系人:黄林鹏,上海 200030,上海交通大学计算机系

(b) times—link $\frac{A \quad B}{A \times B}$ 抽象为 

(c) par—link $\frac{A \quad B}{A+B}$ 抽象为 

(d) cut—link $\frac{A \quad A^{\perp}}{CUT}$ 抽象为 

注:在证明图中,一个以原子为结论的结点,表示为○;而 cut 的处理由于和 times 类似,故这里和 Girard 一样,在算法和证明中不另列出.

### 1.2 标记

标记算法在由颜色构成的集合的簇上操作,基本的操作有:find 给定一元素,返回该元素所在集合名,union 实现两个不相交集合的并.(可用树来表示集合,树根即是集合名,树的每个结点代表集合中的一个元素).

算法如下:

```
(a)对 axiom—link 结点边标记:
   for every axiom—link node do
   mark each edge of it by some C,
   where {C} is a singleton never appered before;
(b)对其它结点进行标记:
  repeat
    case there is a times—node where two input edges have been marked by C1, C2:
        begin
          C1′:= find (C1);
          C2′:= find (C2);
          if C1′=C2′ then return ('FAILURE')
                     else begin
                          mark output edge (if any) by C1′;
                          union (C1′, C2′)
                          end
        end;
    case there is a par—node where two input edges has been marked by C1 、C2 and find(C1)=find(C2),
        say = C:
        begin
        mark output edge (if any )by C
        end
        until proof—graph can't be marked further ;
        if proof—graph has been suceessfully marked by elements just
          from one colur—set then return ('OK')
                                else return ('FAILURE')
```

在上述算法的基础上,我们给出一个新的证明网的定义如下:

**定义**. 若一个证明结构的相应的证明图使用上述算法能成功地被标记为同一类颜色,则该证明结构称为是证明网.

下面我们证明上述定义是合理的,即证明所定义的证明网是线性矢列演算的自然推理,它由下节两定理刻划.

## 2 合理性

本节两定理的陈述和 Girard "linear logic"一文中定理2.7、2.9一样,它们一起描述了证

明网和线性矢列演算之间的关系.

**定理1.** 如果 Π 是线性矢列演算中 $\vdash A1,\cdots,An$ 的一个证明那么相应 Π，我们可以构造一个证明网 Π−，Π− 的终结公式(即结论)为 A1(一次出现)，…，An(一次出现).

证明：证明网 Π− 可由线性矢列演算 Π 如下归纳构造：

case1：Π 是公理 $\dfrac{}{\vdash A, A^{\perp}}$

令 Π− 为 $\dfrac{}{A \quad A^{\perp}}$

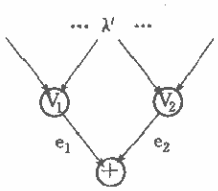
显然 Π− 是一个证明网；

case2：Π 由线性矢列演算 λ 经交换规则而得，令 Π− = λ−；

case3：Π 由线性矢列演算 λ 经由线性矢列演算规则 par 而得：$\dfrac{\overset{\lambda}{\vdash A, B, \Gamma}}{\vdash A+B, \Gamma}$ par

由归纳假设，存在证明网 λ−，在 λ− 中我们可以区分结论 A 和 B，则 Π− 可由 λ− 经下述规则构造：$\dfrac{\overset{\lambda-}{A \quad B}}{A+B}$

设 λ− 对应的证明图为 λ′，则 Π− 对应的证明图 Π′ 为：

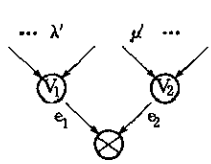



在染色过程中，把对 λ′ 中 Vi 的标记扩充为对其输出边的标记，易见存在某一时刻 e1 和 e2 将被标记为同一类颜色(由于 λ− 是一证明网，V1 和 V2 输入边的标记必属于同一集合)，由于 +(par) 结点的处理不影响 λ′ 中其它结点(∈Π′)的处理，因此，Π′ 中所有边必将被标记为同一集合元素，由定义 Π− 是一证明网；

case4：Π 由线性矢列演算 λ，μ 经规则 times 而得：$\dfrac{\overset{\lambda}{\vdash A, \Gamma} \quad \overset{\mu}{\vdash B, \Delta}}{\vdash A\times B, \Gamma,\Delta}$ times

由归纳假设存在证明网 λ− 和 μ−，并且 A，B 是可区分的，令 Π− 由 λ−、μ− 经下述规则构造而得：$\dfrac{\overset{\lambda-}{A} \quad \overset{\mu-}{B}}{A\times B}$

设 λ−，μ− 对应的证明图分别为 λ′、μ′，则 Π− 对应的证明图 Π′ 为：




在 Π′ 的标记过程中，设结点 V1、V2 满足条件，则在对 V1、V2 的处理中，我们考虑 e1、e2 的

标记，由于 $\lambda'$，$\mu'$不存在共同结点，因此，e1、e2将被标记为不同颜色，设为 C1、C2，在处理 times 结点时，C1、C2被合并成同一类颜色，由于 times 结点的处理不会影响 $\lambda'$ 和 $\mu'$ 中其它结点的标记且 $\lambda-$ 和 $\mu-$ 是证明网，因此最终 $\Pi'$ 将被标记为同一类颜色，即 $\Pi-$ 是证明网. 证毕.

**定理2.** 任给一个证明网 $\beta$，必存在一个线性矢列演算中的证明 $\Pi$，使得 $\Pi--=\beta$.

证明：对 $\beta$ 中的 link 数进行归纳：

(a) $\beta$ 中只有一个 link：

则 $\beta$ 必形为 $\dfrac{}{A\quad A^{\perp}}$

令 $\Pi$ 为 $\dfrac{}{\vdash A,\quad A^{\perp}}$

显然 $\Pi-=\beta$；

(b) $\beta$ 中 link 数$>1$：

则 $\beta$ 中必存在异于 axiom－link 的 times－link 或 par－link，

考虑 $\beta$ 中所有结论的结构：

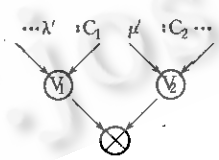
case1：存在一结论其结构为 A＋B，考虑略去该 par－link 的证明结构，它也是一证明网，由归纳假设存在线性矢列演算

令 $\overset{\lambda}{\vdash A, B, \Gamma}$

$$\Pi=\frac{\overset{\lambda}{\vdash A, B, \Gamma}}{\vdash A+B, \Gamma}\ \text{par}$$

则有 $\Pi-=\beta$；

case2：$\beta$ 中所有结论其结构皆形为 $A\times B$，即其证明图的终结点皆为 times 结点(包括◯结点). 考虑对该证明图的染色，在上述终结点中必存在一 times 结点它是最后被处理的(◯结点和 axiom－link 相关，最先被处理，par 结点由于不是终结点，在算法中不可能被最后处理)，设恰在处理该结点之前，网已被染成两不同类颜色 C1，C2如图所示.




记包含所有标记为 C1(C2)类的边(去除 e1、e2)及相关结点组成的证明图为 $\lambda'$ 和 $\mu'$ 由算法的性质，$\lambda'$ 和 $\mu'$ 没有公共结点，设它们对应的证明网分别为 $\lambda-$ 和 $\mu-$，由归纳假设，存在线性矢列演算 $\overset{\lambda}{\vdash A, \Gamma}$ 和 $\overset{\mu}{\vdash B, \Delta}$

令 $\Pi=\dfrac{\overset{\lambda}{\vdash A, \Gamma}\quad\overset{\mu}{\vdash B, \Delta}}{L\ A\times B, \Gamma, \Delta}\ \text{times}$

则 $\Pi--=\beta$，证毕.

注：在[1]中，Girard 称文章最困难的地方就是证明所定义的证明网的概念等价于矢列演算方法，即它是可矢列化的. 由于 Girard 采用 longtrip 条件来定义证明网，结果在证明定

理2.9.case2说明存在一个 times 结点使得证明网是可分裂时遇到困难，虽然 Girard 通过引入 empire 概念解决了该问题，但证明十分繁琐.

## 3 算法复杂度分析

设 n 为证明结构中公式数，则在标记算法(a)中时间花费最多为 O(n)，在标记算法(b)中最多有 O(n)次 union 运算，O(n * n)次 find 运算，但其中只有 O(n)次不是在 O(1)时间内完成的，由于一个有 O(n)次 find，O(n)次 union 操作的算法的时间复杂度为 O(nα(n,n))，其中 α 是一增长极慢的函数，α(n,n)<n，由此，算法时间复杂度为 O(n * n).

**致谢**　我们的工作得到法国 Girard 教授的帮助，在此表示感谢.

### 参考文献

1　Girard J Y. Linear logic. T.C.S.,1987,**50**:1—101.

2　黄林鹏，孙永强. 一个时间复杂度为 O(n * n)的证明网验证算法. "双 B"代数和计算机逻辑论文集，第二届全国计算机逻辑学术研讨会，苏州：上海交通大学出版社，1991:125—131.

3　Girard J Y. Towards a geometry of interaction. Contemporary Mathematics,1989,**92**:69—108.

4　黄林鹏，孙永强. 线性逻辑导论. 计算机科学，1991(1):15—19.

## A NEW DEFINITION OF PROOF—NETS

Huang Linpeng and Sun Yongqiang

(*Department of Computer Science, Shanghai Jiaotong University, Shanghai* 200030)

**Abstract**　This paper presents a new definition of proof—net and proves that what the authors defined is the "natural deduction of linear logic". The complexity of deciding whether a given proof structure is a proof—net by definition is O(n * n), by the way, the proof of sequentialization of proof—nets so defined is very natural and simple.

**Key words**　Linear logic, proof—nets, parallel computation.