arXiv:2602.04045v1 [cs.LO] 3 Feb 2026

# 贝叶斯网络与证明网：贝叶斯推断的证明论

R´emi Di Guardia、Thomas Ehrhard、J´erˆome Evrard 和 Claudia Faggian

IRIF，巴黎西岱大学 - 法国国家科学研究中心

**摘要。** 我们研究贝叶斯网络与线性逻辑中证明的图形表示之间的对应关系。本文有三个目标：建立贝叶斯推断的证明论阐释（秉承证明与程序之间的 Curry-Howard 对应思想），提供组合图形方法，并兼顾计算效率。

我们利用这样一个事实：图的分解比证明树或类型推导的分解更加灵活，即使这会使组合性更难处理。

**K关键词：** 证明网 · 贝叶斯网络 · 线性逻辑　　逻辑

## 1 引言

贝叶斯网络 [23] 是概率推理的重要工具，因为它们能够以紧凑的方式表达大型概率分布（因子化表示），并支持高效推断（因子化计算）。

本文通过线性逻辑，对贝叶斯网络和因子化推断提出一种证明论层面的阐释 [10,13], ，即其图语法——证明网——将切消去 （及其对偶的切割展开）嵌入为一种图重写过程。将贝叶斯网络与一种本质上具有资源意识的证明论所提供的丰富工具箱结合起来，其优势在于能够同时容纳组合式的图形推理与高效计算。

在范畴概率论的框架下，由 [1,17,18] 开创的大量研究已经揭示了贝叶斯网络与逻辑之间的深层联系，将一种组合式且结构化的方法引入概率推理（ [16] 中给出了深入论述）。其中的关键要素是字符串图的图形演算，它为形式化的图示推理提供了一种统一且富有表现力的语言，并将其解释到幺半群范畴中。

范畴论方法通常忽略了概率推理的时空消耗，而这恰恰是引入 [22]并发展贝叶斯网络的重要动机。近期，以线性逻辑为基础的研究 [5,6,8] 正将成本感知视角引入语义。具体而言， [8] 提出了一种语义框架，将贝叶斯网络的效率与类型

系统的组合性结合起来：通过赋予（线性逻辑的一种变体） $\lambda$-项一种基于因子的语义，而因子正是贝叶斯网络及其推理算法所依据的数学结构。

借助证明网形式系统，我们旨在同时发挥两种方法的最大优势：效率与图形推理。在此过程中，我们获得一种基础性理解，其中证明论的基石，即切消去（及其对偶，切割展开），发挥着突出作用。

**因子化表示与因子化计算。**联合分布是一个涉及许多变量的全局函数。处理复杂函数的常见方法，是将其因子分解为局部函数的乘积，每个局部函数仅依赖于变量的一个子集。在贝叶斯网络的设定中，因子分解同时涉及表示以及用于（精确）推断的算法。

在证明论中，将一个证明因子分解为更小分量的自然方式，是将其因子分解为若干子证明，然后再通过切割加以复合。我们正是遵循这一方式。在第8.1节中，我们因子分解一个证明网，使其成为更小证明网的复合，其解释具有更小的代价（第8.2节）。

**图形推理。** 在第 7 节中，我们通过给出一个极其简单的（图示）证明，证明了 d-分离的可靠性。d-分离是一种广为人知的图形判据，用于确立随机变量之间的条件独立性。

**图分解与图组合性** 无论是为了实现高效的因子分解，还是为了进行图形推理，我们都利用了这样一个事实：图的分解比证明树（或类型推导）的分解灵活得多。我们面临的挑战在于，（高效的）组合性并非显而易见（定理 5.2），而且将证明树与图分解关联起来也并非易事（第 8.3节）。

**开销。** 一般而言，采用范畴论方法时，实际计算 语义的开销会急剧增加，因为乘积 $\otimes$ 的行为类似于矩阵的张量积（示例见 [8] ）；计算 $n$个二元随机变量的语义，很容易产生规模远大于 $2^n$的中间计算，而后者是完整联合分布的规模。基于因子的语义（如贝叶斯网络以及 [8]中的语义）之所以高效，在于其乘积的定义（即因子乘积）并不像矩阵的张量积那样运作。

**相关工作。** 我们已经在上文提及了与本研究最相关的文献。这里简要评论几篇在技术上与本文联系更为紧密的论文。

带盒子的证明网的语法在 [5], 中引入。不过，尽管这种语法允许对贝叶斯网络进行编码，却无法刻画它们（证明网不一定对应于贝叶斯网络）。我们

受到那篇论文的启发，该论文主张在计算语义时考虑成本。 [5] 采用推理算法高效计算任意证明的指称。本文则采取不同的方向：我们聚焦于证明网，以及推理即解释；在这种方法中，支持计算的数据结构就是证明网本身。

我们采用了 [8] 中提出的因子语义；尽管我们继承了他们的大部分结果，但我们对“组件”和“组合性”的定义更强（正如我们在第 5.2 节中所解释的），从而得到定理 5.2。

## 2 背景

### 2.1 一个非正式示例

贝叶斯网络由两部分组成：定性组件由有向无环图给出，定量组件则由条件概率给出。这与线性逻辑（LL）的证明网十分相似。 [10,13]：证明网是对 LL 证明语法及其切消去的图形表示，并且可以自然地为其赋予定量解释（参见例如， [7]）。

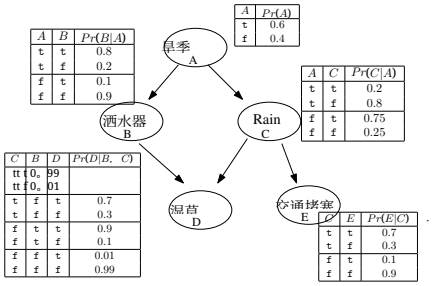



**图 1**。贝叶斯网络示例

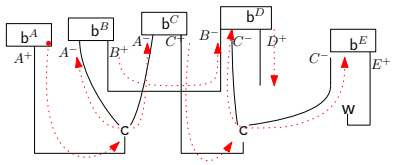



**图 2**。证明网 $\mathcal{R}_D$

**贝叶斯网络示例**。我们先从一个经典示例开始（出自 [4]）。我们希望对这样一个事实建模：早晨草坪是否潮湿，可能取决于是否下雨或洒水器是否开启。反过来，下雨与洒水器的调节都取决于季节。此外，交通堵塞与降雨相关。这五个变量之间的依赖关系（简写为 $A$、 $B$、 $C$、 $D$、 $E$）如图1所示，其中依赖关系的强弱由条件概率表量化。我们想知道：昨晚下雨了吗？假设现在处于旱季，我们的先验信念是，下雨的概率为 0.2。不过，如果我们观察到草坪是潮湿的，对此的确信程度就会提高。更新后的信念称为后验。图1中的模型使我们能够根据证据推断降雨的后验概率，形式化地说

$\Pr(C = \mathrm{t} \mid D = \mathrm{t})$，或者推断草坪潮湿的可能性，即也就是推断边缘概率$\Pr(D = \mathrm{t})$。条件概率和边缘概率是贝叶斯推断可以回答的典型查询，其核心是贝叶斯条件化。因此，要得到后验 Pr

$(C = \mathrm{t} \mid D = \mathrm{t})$，我们必须计算：

– 边缘概率 $\Pr(C = \mathrm{t}\ D = \mathrm{t})$，可以通过对联合概率中的其他变量求和消去来得到（边缘化）；– 再用证据的边缘概率 $\Pr(D = \mathrm{t})$ 进行归一化，该概率以类似方式计算。

**证明网示例。** 证明网 [10,13] 是线性逻辑序列式演算证明的图形表示。图 1 中的贝叶斯网络1可以编码为乘法 线性逻辑（MLL），表示为图 2中的证明网。

节点 $\mathrm{b}^A$、$\mathrm{b}^B$、$\mathrm{b}^C$、..... 是盒，用于存储语义信息——例如，与图 1中相同的条件概率分布。证明网的边将这些信息连接并传递。请注意，图 2 中的边标记为原子公式，可以是正的，也可以是负的。证明网中的信息流（其交互几何 [12,11,2]）遵循原子的极性：沿正原子（携带“输出信息”）向下流动，沿负原子（“输入信息”）向上流动。 MLL 允许复制负原子携带的信息——通过c节点（收缩）。 w节点（弱化）则会阻断这些信息。

证明网 $\mathcal{R}_D$ 见图2 只有一个结论 $D^+$。它表示边缘概率 $\Pr(D)$。下面非正式地看看如何从 $\mathcal{R}_D$中进行采样。

例 *2.1*（从证明网采样）。唯一的初始节点（相对于流而言）是 $\mathrm{b}^A$，它从 $\Pr(A)$ 中输出一个样本。假设该值为 t。这个样本通过 c 节点（收缩）传播到 $\mathrm{b}^C$ 和 $\mathrm{b}^B$。当 $\mathrm{b}^C$接收到 t 时，它会从分布 $\Pr(C|A = \mathrm{t})$ 中采样一个值（例如，以 0.2 的概率取 t）。假设 $\mathrm{b}^C$ 的输出为 t，而 $\mathrm{b}^B$ 的输出（通过类似过程获得）为 f。当盒 $\mathrm{b}^D$ 接收到这些值时，它会从 $\Pr(D|C = \mathrm{t}\ B = \mathrm{f})$ 中输出一个样本，该样本来自边缘概率 $\Pr(D)$。

### 2.2 贝叶斯推理的语言

让我们简要回顾一下所使用的贝叶斯建模语言。更多详情请参阅 [3] ，其中有简明的介绍；若要全面了解，则可参考标准教材 [23,4,21]。

贝叶斯方法为在不确定条件下对不完全信念进行推理提供了一种形式体系。由于我们无法确定某些感兴趣特征的状态，只能确定某个特征处于某种状态的可能性 。随机变量（r.v.s）表示所建模系统的特征。随机变量可以看作原子命题（例如 “潮湿”）的名称 ，其取值来自一组状态（例如

{t，f}）。系统建模为感兴趣变量所有可能取值上的联合概率分布 ——每种赋值都代表系统的一种可能状态。

随机变量。 我们采用标准约定：用大写字母（例如$X$， $Y$）表示随机变量，而小写字母（例如 x，y）表示这些变量的特定固定取值 ，即 x是随机变量 $X$的一个赋值。按照惯例，Pr($x$)表示Pr($X = x$)。为简单起见，这里假设随机变量均为二元变量，其取值集 Val($X$) 为 = {t，f} ——推广到任意离散随机变量是直接的。

有限集随机变量$\mathbf{X} = \{X_1, \ldots, X_n\}$定义了一个“复合”随机变量，其取值集 Val(**X**) 是 Val($X_1$) $\times \cdots \times$ Val($X_n$) 的笛卡尔积。笛卡尔积中的一个元组是**X**的一个赋值，记作**x**。

名称。 给定一个可数名称集 Names，我们为每个名称 $X \in$Names 关联一个有限 取值集，记为 Val($X$)（通常 Val($X$) = {t，f}）。从现在起，我们默认将名称 $X$ 与二元组 ($X$，Val($X$)) 等同，这实际上定义了一个随机变量。

对概率模型的查询。给定一个概率模型 Pr(**X**)，以及感兴趣的变量 $\mathbf{Y} \subset \mathbf{X}$，典型的查询是边缘概率 Pr(**Y**)，或 Pr(**Y**|**X** = **z**)，即给定证据 **z** 时**Z**下**Y**的概率。前者通过对不相关的随机变量进行求和消去得到；后者则通过**贝叶斯定理**求得，这需要计算两个**边缘概率**：Pr(**Y**，**Z**) 和 Pr(**Z**)，因为 Pr(**Y**|**Z**) = $\frac{\Pr(\mathbf{Y},\mathbf{Z})}{\Pr(\mathbf{Z})}$

### 2.3 贝叶斯网络：形式化定义

贝叶斯推理面临的挑战在于，联合概率分布通常过于庞大，难以显式表示。例如，64 个二元随机变量的联合概率分布对应 $2^{64}$ 个条目。

贝叶斯网络通过一种紧凑的方式表示联合概率分布，即采用因子化表示：为有向无环图中的每个节点关联一个条件概率表（条件概率表）。第 5.1 节将重新介绍贝叶斯网络的语义。

一个定义在随机变量集合 贝叶斯网络 $\mathcal{B}$ 上的贝叶斯网络是一对（$\mathcal{G}$，$\Phi$），其中：

– $\mathcal{G}$ 是定义在节点集合 **X**上的有向无环图（DAG）。– $\Phi$ 为每个变量 $X \in \mathbf{X}$ 分配一个条件概率表（*CPT*），用于表示 $X$ 在给定其父节点时的概率。

## 3 贝叶斯证明网

贝叶斯证明网以乘法线性逻辑的图语法为基础，并扩展了概率盒，用于编码条件概率表

（CPT）。我们修订乘法线性逻辑证明网的语法，并在第3.2节定义贝叶斯证明网及其图重写规则。

最后，我们将展示（第3.4节），图1 和2 中概述的贝叶斯证明网与贝叶斯网络之间的对应关系是可靠且完备的。

### 3.1 乘法线性逻辑（带概率盒）

我们假设给定一个可数的符号集，记作元变量 $X$、$Y$、$Z$。公式的语法采用线性逻辑乘法片段（MLL）的语法：

$$F, G ::= X^+ \mid X^- \mid 1 \mid \bot \mid F \otimes G \mid F \parr G$$

我们将 $X^+$ （分别将 $X^-$）称为**正的**（分别为**负的**）**原子公式**。**否定** $(\cdot)^\bot$ 通过以下方式归纳定义：$(X^+)^\bot \stackrel{\text{定义}}{=} X^-$，$(X^-)^\bot \stackrel{\text{定义}}{=} X^+$，$1^\bot \stackrel{\text{定义}}{=} \bot$，$\bot^\bot \stackrel{\text{定义}}{=} 1$，$(F \otimes G)^\bot \stackrel{\text{定义}}{=} F^\bot \parr G^\bot$ 以及$(F \parr G)^\bot \stackrel{\text{定义}}{=} F^\bot \otimes G^\bot$。

一个**序列式**是由公式组成的有限序列 $F_1, \ldots\ldots, F_n$。大写希腊字母$\Gamma$，$\Delta$，…… 表示序列式。

演算。 演算MLL$^\text{B}$ 是对乘法线性逻辑 MLL的扩展，由 [5]（与 [15]类似）引入。除MLL规则外，还有一条广义公理规则，称为盒 （稍后讨论）：

$$\frac{}{\vdash Y_1^-, \ldots, Y_k^-, X^+}\ \text{box} \qquad (k \geq 0)$$

盒规则结论中的所有公式都是原子的，其中恰有一个——**Main** 结论 $X^+$ ——是正的，且满足 $X \neq Y_i\ (\forall i \leq k)$。

在线性逻辑中，证明既有相继式演算语法，形式为 **证明树**（即相继式演算规则构成的树），也有一种称为**证明网**的图语法。本文关注后者，如图 3所示。与之对应的相继式演算是标准的——为求完整起见，第 A 节给出了其规则及其作为证明网的对应图示。图 3中的语法在标准证明网中加入了一种新节点，即盒，它对应于新的 盒-规则。我们也将盒-节点称为盒。

**定义 3.1**（线性逻辑$^\text{B}$ **证明网**）。一个线性逻辑$^\text{B}$ 证明网是一个带类型图 $\mathcal{N}$（要点*1*），并满足条件*2*（悬空边）和*3* （正确性）。

*1.* 带类型图[1]。 $\mathcal{N}$ 是一个部分[2] 图，其边以MLL 公式为标签。节点字母表见图*3*：每个节点都标有一条规则。与节点 n 关联的边需要符合语法中给出的类型条件，并分为 前提（属于 n，绘于其上方 n）或其 结论（绘于 n下方）。此外，我们要求每条边至多是一个节点的结论，且至多是一个节点的前提。

[1] 在文献中，带类型图通常称为证明结构或模。[2] 我们允许存在悬空边。

*2.* 悬空边。 $\mathcal{N}$ 中的每条边都是某个节点的结论。不是任何节点前提的边（即悬空边）称为 $\mathcal{N}$的结论。*3.* 正确性。 $\mathcal{N}$ 中的每个环都至少包含同一个 $\parr$节点或同一个c节点的两个前提。

我们用 $\mathcal{R}\colon \Delta$ 表示一个结论为 $\Delta$的证明网 $\mathcal{R}$ 。证明网$\mathcal{N}$的 **子网** 是 $\mathcal{N}$ 的一个子图，且其本身也是一个证明网。

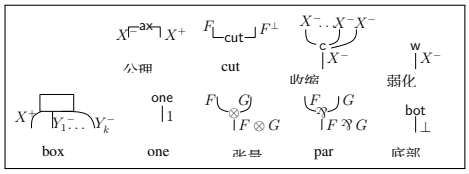



**图 3.** 节点语法。对于框节点， $k \geq 0$ 和 $X \neq Y_i$ （$\forall i \leq k$）

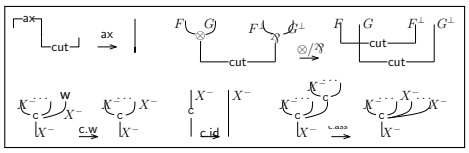



**图 4.** 归约规则

正确性条件（点 3）确保该图是相继式演算中的一棵证明树的像。

**定理 3.1（序列化）**。每个证明网 $\mathcal{N}$ 都是相继式演算中一棵证明树的像，也就是说，它可以通过图中的规则归纳生成。*11*。

**概率盒：采样与条件**。乍看之下，与通常的线性逻辑证明网相比，我们将为盒节点关联一个条件概率表，这些节点似乎不同寻常。这里我们简要说明如何利用线性逻辑的加性联结词构造这些盒，唯一额外添加的是一个表示抛硬币的样本节点。

假设给定形如 $\overset{p}{X^+}$ 的样本节点，其中 $0 \leq p \leq 1$。这样的节点表示二元随机变量 $X$上的概率分布：其取真值的概率为 $p$，取假值的概率为 $1-p$——示意图见图 5。我们将正原子公式 $X^+$ 视为布尔值的别名，它对应的公式是$1 \oplus 1$在线性逻辑中的形式。根据对偶性，这意味着 $X^-$ 是 $\bot \& \bot$的别名。将布尔值加法编码为 $1 \oplus 1$ 是标准做法；“如果-那么-否则”函数（接受一个布尔值并返回一个布尔值，因此其类型为 $\vdash \bot \& \bot$， $1 \oplus 1$）由一条 & 规则（一个加法框）编码，见图 5。

因此，任何 $CPT$ 都可以通过适当的样本节点以及（可能嵌套的）如果-那么-否则，以标准方式直接编码。图 盒节点内部工作原理的图形表示见图 5。

从现在起，我们不再显式展示每个概率盒的内部内容，而只将其与相应的条件概率表关联起来。

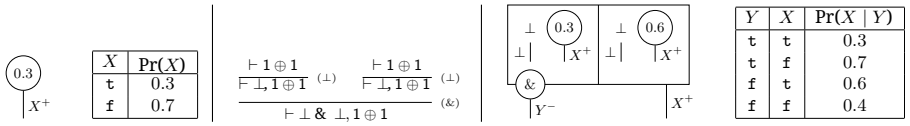



**图5.** 一个样本节点（左）、“若则否则”与&-规则（中），以及由样本节点和加性连接词构成的概率框（右）

注 *3.1*（弱化与收缩）。该语法允许对负原子公式应用结构规则（弱化与收缩）。回想一下，$X^-$ 表示 $\bot\&\ \bot$。在线性逻辑中，众所周知，由 $\bot$ 通过&（以及 ⅋）构造的公式，即具有负极性的公式，允许弱化与收缩。

### 3.2 贝叶斯证明网

**定义 3.2（贝叶斯证明网（bpn））**。我们称满足以下两个条件的带概率盒证明网为贝叶斯证明网$\mathcal{R}$： $\Delta$

*1.* 标记盒的正结论的原子两两不同；并且 *2.* 结论 $\Delta$ 中的所有原子都是正的。

我们也将贝叶斯证明网的子网称为贝叶斯证明网。

我们用 bpn 表示贝叶斯证明网。根据条件（1），我们用其（唯一的）正结论 $X^+$的名称来指称 bpn $\mathcal{R}$ 中的每个盒，记作 $\mathrm{b}^X$。我们称满足条件（2）的证明网为**正的**： $\Delta$ 其中 $\Delta$ 所有原子（子）公式都是正的。我们用 Boxes($\mathcal{R}$) 表示 $\mathcal{R}$的**所有盒构成的集合**。我们将 $\mathcal{R}$中标记盒的正结论的名称集合称为**主名称**（记作 Main($\mathcal{R}$)）。

### 3.3 归约、展开与范式

图。4 展示了切割重写步骤，以及收缩和弱化的结构规则。这些规则都是标准规则，并且保持证明网的正确性和结论不变。

**定义 3.3（归约、展开与范式）**。*MLL* 的范式化规则[B] 证明网见图4。这些重写规则中的每一条r 都定义了证明网上的一个二元关系 $\xrightarrow{\mathtt{r}}$，称为归约步骤，记作 $\mathcal{R} \xrightarrow{\mathtt{r}} \mathcal{R}'$（读作 $\mathcal{R}$ $r$归约为 $\mathcal{R}'$）。其逆步骤称为展开。因此，若 $\mathcal{R} \xrightarrow{\mathtt{r}} \mathcal{R}'$，则 $\mathcal{R}$ 是 $r$的展开 $\mathcal{R}'$。我们用 $\xrightarrow{\mathtt{R}}$ 表示依据图4中任一规则进行的归约。 $\mathcal{R}$若不存在 $\mathcal{R}'$ 使得 $\mathcal{R} \xrightarrow{\mathtt{R}} \mathcal{R}'$，则 $\mathcal{R}$处于范式（或简称为正规）。

**命题 3.1.** 归约 $\xrightarrow{\mathtt{R}}$具有终止性和合流性。

**范式与原子证明网**。 请注意，由于存在盒，处于范式的证明网 $\mathcal{N}$ 仍可能包含切。不过， $\mathcal{N}$ 具有一种特殊结构：它可以分解为一个证明网，其所有边都标记为原子，并位于其结论的公式树之上。

原子证明网。 如果一个证明网 $\mathcal{N}$ 的所有边上的公式标记都是原子，则称其为**原子的** 。若 $e$的标记为 $X$ ，即其标记为 $X^+$ 或$X^-$，我们就这样表示。

性质 *3.1*（范式）。设 $\mathcal{N}: F$ 是一个结论为$F$的范式证明网。则 $\mathcal{N}$可分解为两个子图 $\mathcal{N}_{\mathsf{at}}$和 $\mathcal{N}^F$，其中 $\mathcal{N}_{\mathsf{at}}$： $\Gamma$是一个原子子网，其结论 $\Gamma$ 是 $F$的原子子公式序列；而 $\mathcal{N}^F$是一个以前提 $\Gamma$为端点的带类型图，由 $F$的语法树构成。若 $\mathcal{N}$ 的结论为 $F_1, \ldots, F_k$，则也有类似性质。

### 3.4 贝叶斯网络和贝叶斯证明网

我们证明，每个贝叶斯网络都对应一个正的贝叶斯证明网，反之亦然。为了形式化这一结果，我们需要极化序的概念。

极化序。设 $\mathcal{R}$ 是一个原子证明网（这是极化证明网 [20]的一个简单情形）。我们记$Pol(\mathcal{R})$为一个与 $\mathcal{R}$具有相同节点和边的图，但其定向遵循标记的极性：正极性时向下，负极性时向上。以下简单性质对所有极化证明网都成立，且广为人知。

**引理 3.1（极化正确性** [19]）。假设 $\mathcal{R}$ 是一个带类型图（定义*3.1*，第*1*点），且其所有边都标记为原子。那么，当且仅当 $Pol$（$\mathcal{R}$）是一个有向无环图时， $\mathcal{R}$ 才是正确的。

极化序。极化定向在 $\mathcal{R}$的节点上诱导出一个偏序 $<_{\mathrm{pol}}$ 。特别地，若 $\mathrm{b}_1$ 和 $\mathrm{b}_2$ 是盒，则当且仅当存在一条从 $\mathrm{b}_1$ 到 $\mathrm{b}_2$、位于$Pol(\mathcal{R})$中的路径时，$\mathrm{b}_1 <_{\mathrm{pol}} \mathrm{b}_2$ 。我们将 $\mathsf{dag}(\mathcal{R})$定义为与 $(\mathsf{Boxes}(\mathcal{R}) <_{\mathrm{pol}})$ 相关联的有向无环图。

**从贝叶斯网络到贝叶斯证明网**。对每个贝叶斯网络 $\mathcal{B}$ ，关联一个证明网 $\mathcal{N}_{\mathcal{B}}$是很直接的，如图2所示。唯一需要仔细处理的是检查翻译所得的带类型图是否正确。这是引理3.1、 $\mathcal{N}_{\mathcal{B}}$ 是原子的这一事实，以及$Pol$（$\mathcal{N}_{\mathcal{B}}$）是有向无环图这一事实的推论，因为 $\mathcal{B}$ 是有向无环图。显然， $\mathcal{N}_{\mathcal{B}}$ 是贝叶斯网络。

**从贝叶斯证明网到贝叶斯网络**。值得注意的是，我们可以将一个贝叶斯网络 $\mathscr{B}(\mathcal{R})$与任意正的bpn $\mathcal{R}$的范式关联起来。设 $\mathcal{N}$ 为其范式；回顾性质3.1，设 $\mathcal{N}^{\mathsf{at}}$ 为 $\mathcal{N}$的最大原子子网。由于每个盒都恰好有一个主名称，因此（盒$(\mathcal{N})$ $<_{\mathrm{pol}}$）与以$\mathsf{Main}(\mathcal{R}) = \{X_1, \ldots, X_n\}$上的贝叶斯网络的有向无环图之间存在直接对应关系。

例 *3.1*。考虑图 $\mathcal{R}$ 中的证明网2 。显然，极化序恰好描述了图1中贝叶斯网络的有向无环图。

注意，与 bpn $\mathcal{R}$ 相关联的贝叶斯网络独立于 $\mathcal{R}$的结论。

查询。贝叶斯网络与证明网之间对应关系中的一个细节值得讨论。正 bpn $\mathcal{R}:\Gamma$ 同时编码了一个贝叶斯网络（概率模型）和对该模型的查询：查询由结论 $\Gamma$ 表示。正如将在第 5.2 节中形式化说明的那样， $\mathcal{R}:\Gamma$ 对应于一个定义在 $\mathrm{Nm}(\mathcal{R})$上的贝叶斯网络 $\mathcal{B}_{\mathcal{R}}$ 。虽然 $\mathcal{B}_{\mathcal{R}}$ 在 $\mathrm{Nm}(\mathcal{R})$上定义了一个概率分布 Pr，但 $\mathcal{R}:\Gamma$ 的语义是 Pr 在 $\mathrm{Nm}(\Gamma)\subseteq\mathrm{Nm}(\mathcal{R})$上的边缘分布。

## 4 图分解

树（例如证明、项和类型推导）会按照其归纳结构进行分解。证明网是证明树 $\pi$的像，因此总是可以进行反映 $\pi$ 的分解。然而，由于 bpn 是图，它在分解方面提供了更大的灵活性——我们在选择子图时有更大的自由度。

不过，这里仍有一些细微之处。对 bpn $\mathcal{N}$进行因子分解时，我们希望将其分解为若干由（任意数量的）切割节点连接的子网。对 $\mathcal{N}$而言，这可能无法立即实现，但只需先对 $\mathcal{N}$进行展开 即可轻松解决这一问题，如下文所述。

注 *4.1*（切分边）。注意，公理展开（公理归约的逆过程）会引入一个切割节点，从而可以将边分成 两部分。

**定义 4.1（分裂）**。如果 $\mathcal{N}=\mathrm{Cut}(\mathcal{R},\mathcal{S})$ 可以分解为两个由（任意数量的）切割节点连接的子网 $\mathcal{R}$ 和 $\mathcal{S}$，则记作 $\mathcal{N}=\mathrm{Cut}(\mathcal{R},\mathcal{S})$。

设 $\mathcal{N}$ 是一个证明网， $\mathcal{R}\subseteq\mathcal{N}$ 是一个原子子网。由 $\mathcal{R}$ 诱导的 $\mathcal{N}$ 的分裂，通过将每条既是 $\mathcal{R}$ 的结论、又是 $\mathcal{R}$之外某个节点的前提的边替换为其公理展开来完成。我们将所得的证明网称为 $\mathit{Split}_{\mathcal{R}}(\mathcal{N})$。它是 $\mathcal{N}$ 的一个展开，并可分解为 $\mathit{Split}_{\mathcal{R}}(\mathcal{N})=\mathit{Cut}(\mathcal{R},\mathcal{S})$，其中 $\mathcal{S}$是从 $\mathit{Split}_{\mathcal{R}}(\mathcal{N})$中移除 $\mathcal{R}$ 及新引入的切割节点后得到的子网。（图 *6* 展示了一个例子。）

尽管从图形视角来看，这样一种分解非常自然，但它在相继式演算的归纳语法中并无直接对应。具体而言，虽然 $\mathrm{Split}_{\mathcal{R}}(\mathcal{N})$、 $\mathcal{R}$ 和 $\mathcal{S}$ 都有相应的证明树，但没有理由认为与 $\mathrm{Split}_{\mathcal{R}}(\mathcal{N})$相关联的证明树可以由 $\mathcal{R}$ 和 $\mathcal{S}$的证明树得到。事实上，相继式演算中不存在允许通过多重切割来复合两个给定证明树的规则。因此，证明网中允许的分解多于证明树。

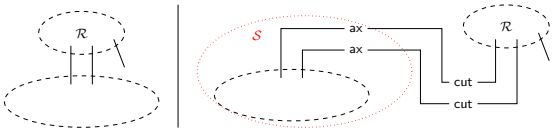



**图 6.** 定义 4.1 的图示， $\mathcal{N}$在左侧，$\mathsf{Split}_{\mathcal{R}}(\mathcal{N})$ 在右侧

## 5 语义与组合性

遵循 [8], ，我们采用与贝叶斯网络相同的语义。在回顾主要要素之后，我们扩展了组合性证明，其对象为 [8], ，方法是采用一种“组件”概念，该概念专属于图，因此并未出现在先前的工作中。

### 5.1 因子与贝叶斯网络的语义

推理算法依赖于一类称为因子的函数上的基本运算；这类函数推广了概率分布和条件分布的概念。因子也将是我们语义中的关键成分。

$\mathbf{x}$ **定义 5.1（因子）**。随机变量集合 $\mathbf{X}$ 上的因子 $\phi$ 是一个函数 $\phi : \mathsf{Val}(\mathbf{X}) \to \mathbb{R}_{\geq 0}$，它将每个元组 $\mathbf{x} \in \mathbf{X}$ 映射为一个非负实数。

字母 $\phi$, $\psi$ 取值于因子。当 $\mathbf{X}$ 可由上下文明确时，我们简写为 $\phi$ （省略上标 $\mathbf{X}$）；此时名称集合$(\phi)$ 表示$\mathbf{X}$。在关于贝叶斯网络的文献中， $\phi(\mathbf{x})$ 常写作 $\phi_{\mathbf{x}}$。我们在显式计算中采用这一便利记法。因子带有两种重要运算：求和（消去）与乘积。因子乘积的定义方式使得只有“相容”的赋值才会相乘。

给定子集$\mathbf{Y} \subseteq \mathbf{X}$，我们用$\mathbf{x}|_{\mathbf{Y}}$ 表示$\mathbf{x}$ 在$\mathbf{Y}$ 上的限制（因此，$\mathbf{x}|_{\mathbf{Y}} \in \mathsf{Val}(\mathbf{Y})$)。给定两个名称集$\mathbf{X}$ 和$\mathbf{Y}$，赋值$\mathbf{x} \in \mathsf{Val}(\mathbf{X})$与$\mathbf{y} \in \mathsf{Val}(\mathbf{Y})$兼容（简写为$\mathbf{x} \sim \mathbf{y}$ ），当$\mathbf{x}|_{\mathbf{X}\cap\mathbf{Y}} = \mathbf{y}|_{\mathbf{X}\cap\mathbf{Y}}$时；即它们在共同名称上取值一致。

– 从$\mathbf{x}$ $\phi$ 中对$\mathbf{Z} \subseteq \mathbf{X}$ 进行**求和消元**，得到一个在$\mathbf{Y}$ 上的因子 $\sum_{\mathbf{Z}} \phi$，其定义 $= \mathbf{X} - \mathbf{Z}$如下：

$$\left(\sum_{\mathbf{Z}} \phi\right)(\mathbf{y}) \stackrel{\text{def}}{=} \sum_{\mathbf{z} \in \mathsf{Val}(\mathbf{Z})} \phi(\mathbf{z}\,\mathbf{y})$$

– $\mathbf{x}$ $\phi_1$ 与$\mathbf{Y}$ $\phi_2$ 的**乘积** 是一个定义在$\mathbf{Z}$ 上的因子 $\phi_1 \odot \phi_2$ def $= \mathbf{X} \cup \mathbf{Y}$，定义如下：

$$(\phi_1 \odot \phi_2)(\mathbf{z}) \stackrel{\text{def}}{=} \phi_1(\mathbf{x})\phi_2(\mathbf{y}) \quad \text{where } \mathbf{x} = \mathbf{z}|_{\mathbf{X}} \text{ and } \mathbf{y} = \mathbf{z}|_{\mathbf{Y}}.$$

我们用 $\bigodot_n \phi_n$ 表示 $n$ 元乘积。我们用 $\mathbf{1}_{\mathbf{Y}} \stackrel{\text{def}}{=} \mathbf{Y}\mathbf{1}$ 表示定义在$\mathbf{XY}$ $\mathbf{X}$名称集$\mathbf{Y}$上的因子，它将 Val(**Y**) 中的每个元组都映射为 1。注意，$\phi \odot \mathbf{1} = \phi$ 当$\mathbf{Y} \subseteq \mathbf{X}$时。允许定义在空变量集上的因子，并称其为平凡因子。特别地，我们用 $\mathbf{1}_{\emptyset} \stackrel{\text{def}}{=} \mathbf{1}^{\emptyset}$ 表示将空元组映射为 1 的平凡因子。乘积和求和都是可交换的，乘积满足结合律，而且——至关重要的是——它们在适当条件下满足分配律:

$$\text{If } \mathbf{Z} \cap \mathsf{Nm}(\phi_1) = \emptyset \text{ then } \sum_{\mathbf{Z}} (\phi_1 \odot \phi_2) = \phi_1 \odot \left( \sum_{\mathbf{Z}} \phi_2 \right) \tag{1}$$

这种分配律是精确推理算法所依赖的关键性质。CPT 是因子，因此可以进行求和和乘积运算。请注意，这类运算的结果不一定是 CPT，但当然是因子。

注*5.1*（因子运算的开销）。从因子 $\phi$ 中对任意数量的变量进行求和消去，需要 $\mathbf{O}(2^w)$ 的时间和空间，其中 $w$ 是其定义所涉及的变量数量。将 $k$ 个因子相乘，需要 $\mathbf{O}(k \cdot 2^w)$ 的时间和空间，其中 $w$ 是结果因子中的变量数量。

**贝叶斯网络的语义**。贝叶斯网络中分配给每个节点的条件概率表是一个因子 $\phi^X$，其涉及变量 $\{X\} \cup \mathsf{Pa}(X)$，其中 $\mathsf{Pa}(X)$ 表示 $X$ 在 $\mathcal{G}$ 中的父节点集合。贝叶斯网络上的独立性假设给出:

**定理5.1**（[22]）。一个贝叶斯网络 $\mathcal{B}$ 定义在随机变量集合 $\mathbf{X}$ 上，它在 $\mathbf{X}$ 上定义了唯一的概率分布（其语义）：$\llparenthesis \mathcal{B} \rrparenthesis \stackrel{def}{=} \bigodot_{X \in \mathbf{X}} \phi^X$。

注意，$\llparenthesis \mathcal{B} \rrparenthesis$ 在子集$\mathbf{Y} \subseteq \mathbf{X}$ 上的边缘分布 为$\sum_{\mathbf{X}-\mathbf{Y}} \llparenthesis \mathcal{B} \rrparenthesis$。

### 5.2 贝叶斯证明网的语义

bpn $\mathcal{R} : \Delta$ 的语义与贝叶斯网络的语义类似。

盒子的语义。我们已经看到，对于每个（名称为）原子，我们都关联一个布尔随机变量；对于每个盒 $\mathsf{b}^X$：$X^+$，$Y_1^-$，$\ldots Y_k^-$，我们关联 $\Pr(X|Y_1, \ldots, Y_k)$，它是一个因子，以下简记为 $\phi^X$。因此，$\mathsf{b}^X$ 的语义显而易见:

$$\llparenthesis \mathsf{b}^X \rrparenthesis \stackrel{\text{def}}{=} \phi^X$$

*bpn* 的语义*bpn*。正证明网 $\mathcal{R} : \Delta$ 定义了 $\mathsf{Nm}(\Delta)$ 上的边缘分布。与各个盒相关联的 CPT 的乘积，就是底层模型的联合概率分布；最后，对结论中未出现的所有名称进行求和消元，即可得到所需的边缘分布。

**定义 5.2（bpn 的语义）**。设 $\mathcal{R}$： $\Delta$ 是一个*bpn*。其语义为

$$(\!|\mathcal{R}|\!) \stackrel{def}{=} \sum_{\mathbf{X}-\mathsf{Nm}(\Delta)} \left( \bigodot_{\mathsf{b}\in\mathsf{Boxes}(\pi)} (\!|\mathsf{b}|\!) \right)$$

对于 $\mathbf{X} = \bigcup_{\mathsf{b}\in\text{盒}(\pi)}$名称集合(b)。

注记 *5.2.* 若 $\mathcal{R}$ 是一个证明网，且 $\mathsf{Boxes}(\pi) = \emptyset$，则 $(\!|\pi|\!) = \mathbf{1}_{\emptyset}$。

### 5.3 不变性与组合性

如此定义语义有两个明显的优点：

– 它与贝叶斯网络语义之间的对应关系一目了然，从而能够高效地表示概率模型；– 通过归约得到的语义具有不变性，这一点也显而易见，因为盒和结论都是不变的。

然而，关键问题在于组合性。在指称语义中，通常会归纳地定义解释；这样一来，组合性便是内在的，而难点在于证明语义在归约和展开下保持不变。采用全局定义后，不变性便自然成立，但组合性仍需证明。

这样的语义是否与按组件给出的模块化定义相容？答案是正的，而且比以往工作中的意义更强 [8],在那些工作中，组合性遵循类型推导的归纳式、树状定义。

证明树的组合性。 根据定理 3.1，每个证明网都是某棵序列演算证明树的像，如图 11 所概括。借鉴 [8]，，给定一个证明网 $\pi$，它由 $\pi_1$、……、 $\pi_n$ 的复合得到，我们得到 $(\!|\pi|\!)$，它来自$(\!|\pi_1|\!)$、……、 $(\!|\pi_n|\!)$，如下所示

$$(\!|\pi|\!) = \sum_{\mathbf{Z}} \left( \bigodot_i (\!|\pi_i|\!) \right) \quad \text{where } \mathbf{Z} = \bigcup_i \mathsf{Nm}((\!|\pi_i|\!)) - \mathsf{Nm}(\Delta) \tag{2}$$

复合通过首先 做乘积 $\bigodot_i (\!|\pi_i|\!)$来获得——这会在名称 $\bigcup_i$名称集合$((\!|\pi_i|\!))$ 上产生一个因子——并随后 进行边缘化，即对不出现在结论 $\Delta$中的名称做求和消去。这样一种复合概念可以看作复合 $=$ 并行复合 $+$ 隐藏。

请注意——正如 [8],方程2 所强调的，这并非显而易见，因为求和消元与乘积通常并不满足分配律，只有在适当条件下才成立。

图组合性。本文要求一种比 [8], 中更强的组合性概念，因为我们希望采用一种更一般的“组件”概念，而这种概念只有在图论语境中才自然。如上所见，由于证明网 $\mathcal{N}$ 是证明树的像，因此当然可以按照表 Fig. 中的规则对其进行归纳分解。11。然而，证明网首先是一个图，我们可能希望根据任意子图来操作它。我们证明，无论如何将证明网 $\mathcal{R}$分解为子网，都可以组合地定义其语义。特别是，如定义4.1所示，我们可以将证明网 $\mathcal{R}$ 分解为两个由任意多个切连接的分量。这种分解在类型推导的归纳结构（[8]）或相继式演算中都没有对应形式（在相继式演算中，两个证明树之间至多只能通过一个单一的切相连）

在第 8.3节中，我们将证明，总能以一种保持语义不变的方式为这种分解赋型，而且所选的分解可以归纳定义。

### 5.4 图的组合性

图组合性的证明依赖于这样一个事实：原子 bpn 满足一个关键性质，使其成为连接树；连接树是消息传递所依据的数据结构，而消息传递是贝叶斯网络精确推理中最常用的算法。具体而言，在每个原子 bpn 中，标记为同一个原子 $X$ 的边构成一棵树，因此，任意两条标记为同一个原子 $X$ 的边，都由一条路径相连，且该路径上的所有边都标记为 $X$。

**引理 5.1（连接树式性质）**。设 $\mathcal{R} : \Lambda$ 是一个正的 *bpn*。

– $\mathcal{R}$在标记为同一个原子 $X$ 的边上的限制构成一棵有向树，其根是盒$\mathfrak{b}^X$的主结论。– 每个 $X \in$ 名称集合$(\mathcal{R})$ 都是由一个盒$\mathfrak{b}^X$引入的。

证明见第B节。因此，对于每个bpn，都有以下结论。

**推论 5.1（内部名称）**。设 $\mathcal{N}$ 是一个*bpn*，满足 $\mathcal{N} = \mathrm{Cut}(\mathcal{N}_1, \mathcal{N}_2)$，其中 $\mathcal{N}_i : \Gamma_i$。则

$$Z \in \mathsf{Nm}(\mathcal{N}_j) \ \textit{and}\ Z \notin \mathsf{Nm}(\Gamma_j) \quad \Rightarrow \quad Z \notin \mathsf{Nm}(\mathcal{N}_i) \ \textit{for}\ i \neq j.$$

利用这一事实，我们能够证明，语义确实可以通过复合来计算，从而验证公式（2），对于图论意义下的组件也是如此。

**定理 5.2（图的组合性）**。设 $\mathcal{R} = \mathrm{Cut}(\mathcal{R}_1, \mathcal{R}_2)$: $\Delta$ 是一个*bpn*。则$(\!|\mathcal{R}|\!) = \sum \mathbf{z}((\!|\mathcal{R}_1|\!) \odot (\!|\mathcal{R}_2|\!))$，其中$\mathbf{Z} = \big($名称集合$((\!|\mathcal{R}_1|\!)) \cup$ 名称集合$((\!|\mathcal{R}_2|\!))\big)$ –名称集合$(\Delta)$。

证明见第 B节。

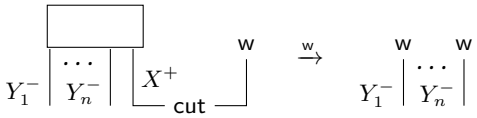



**图7.** box/w 归约

## 6 弱化：剪枝图

回想一下，给定一个贝叶斯网络 $\mathcal{B}$，它定义了关于$\mathbf{X}$的概率分布 Pr，我们可以通过对其他变量求和消去，得到 Pr 关于感兴趣的变量$\mathbf{W} \subseteq \mathbf{X}$ 的边缘分布。如果我们只关心子集$\mathbf{W}$，则可以通过剪枝掉对应于不会用到的随机变量的节点，在不丢失信息的情况下使用一个更小的 $\mathcal{B}$：也就是说，将$\mathcal{B}$ 限制为$\mathbf{W}$的祖先节点。

在逻辑方面，我们对此已有充分理解（并且曾广泛使用）：边缘化在逻辑上是通过弱化实现的。例如，在图2中，要对 $E$ 进行边缘化，只需切割与一个 w-节点相连的边。剪枝对应于实际执行标准的 box/w 归约（见图7）这一新的重写系统与先前的归约以及语义都具有良好的相容性。

**引理6.1.** 归约$\mathsf{w} \longrightarrow$ 具有终止性和合流性。更重要的是，$\overset{\mathsf{R}}{\longrightarrow} \cup \mathsf{w}\!\!\longrightarrow$具有终止性和合流性。

**引理6.2**。 给定一个正 *bpn* $\mathcal{R}$，如果 $\mathcal{R} \mathsf{w}\!\!\longrightarrow \mathcal{S}$ 那么 $(\!|\mathcal{R}|\!) = (\!|\mathcal{S}|\!)$。

引理 6.1 是直接的，而引理 6.2 的证明见第 C节。

## 7 条件独立性上的图形推理

我们将在此展示如何使用 bpn 进行图形推理。与贝叶斯网络一样，一个简单直观的图形判据可以用来确定随机变量集合之间的条件独立性。我们的贡献是给出一个关于正确性的直接 且即时的图形证明。

贝叶斯网络允许通过一种称为 *d*-分离的图形判据来推导条件独立性，该判据最早由 Pearl 提出 [23]。在此，我们将 Darwich 在 [4]（定理4.1）中给出的、以剪枝和非连通性表述的 d-分离重新构造为一种测试，其判定所需的时间和空间均与有向无环图的规模呈线性关系。在弦图设置中也使用了类似的图形判据 [9]。

**条件独立性**。 条件独立性是概率论中的一项基本性质。它在贝叶斯建模中极为有用：它简化了推理，即关于随机变量组的推理，并且是若干推理算法的核心。

直观上，若关于一随机变量的信息不能提供关于另一随机变量的信息，则这两个随机变量相互独立。独立性可以条件化于来自第三个变量的信息。形式上，设 Pr 为随机变量 $\mathbf{W}$ 以及 $\mathbf{X},\mathbf{Y},\mathbf{Z} \subseteq \mathbf{W}$ 这三个互不相交的随机变量集合上的概率分布。

– $\mathbf{X}$ 和 $\mathbf{Y}$ 相互**独立**，记作 $\mathbf{X} \perp \mathbf{Y}$，当且仅当 $\Pr(\mathbf{X},\mathbf{Y}) = \Pr(\mathbf{X}) \odot \Pr(\mathbf{Y})$。– $\mathbf{X}$ 和 $\mathbf{Y}$ 在给定 $\mathbf{Z}$ 时**条件独立**，记作 $\mathbf{X} \perp \mathbf{Y} \mid \mathbf{Z}$，当且仅当 $\Pr(\mathbf{X},\mathbf{Y} \mid \mathbf{Z}) = \Pr(\mathbf{X} \mid \mathbf{Z}) \odot \Pr(\mathbf{Y} \mid \mathbf{Z})$。

**bpn 上的 d-分离bpn**。取 $\mathcal{M}$ ： $\Gamma$ 任意一个正的bpn ，它关于 $\xrightarrow{\mathrm{R}}$ 和$\mathsf{w} \rightarrow$均处于范式。回想一下，它有一个与其语义相同的原子子网 $\mathcal{M}^{\mathsf{at}}$ （性质3.1）。以下图形判据可用于确定概率分布 $(\!|\mathcal{M}|\!) = (\!|\mathcal{M}^{\mathsf{at}}|\!)$中随机变量$\mathsf{Nm}(\Gamma)$之间的条件独立性。

**定义 7.1（不连通）**。对于 $\mathcal{M}$: $\Gamma$ 一个正 *bpn*，处于 $\mathrm{R},\ \mathsf{w}\!\!\rightarrow$范式，令 $\{\mathbf{X}$、$\mathbf{Y}$、$\mathbf{Z}\}$ 为 $\mathsf{Nm}(\Gamma)$的一个划分。若在从 $\mathcal{M}^{\mathsf{at}}$ 中移除所有标记为 $\mathbf{Z}$ 的边后得到的图中，$\mathbf{X}$ 与 $\mathbf{Y}$ 之间不存在路径（对于任意 $X \in \mathbf{X}$ 和 $Y \in \mathbf{Y}$），则称 $\boldsymbol{X}$ 与 $\mathbf{Y}$ 被 $\mathbf{Z}$ 不连通。

不连通意味着$\mathbf{X}$ 和$\mathbf{Y}$ 在给定$\mathbf{Z}$的条件下相互独立。

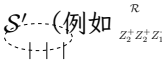
**定理 7.1（判据的可靠性）**。在与定义 7.1相同的假设下，若$\mathbf{X}$ 和$\mathbf{Y}$ 被$\mathbf{Z}$ 分隔，则$\mathbf{X} \perp \mathbf{Y} \mid \mathbf{Z}$ 在该分布中成立$(\!|\mathcal{M}|\!)$。

我们通过图形分解并结合语义的组合性，给出一个直接而简明的证明。

**证明**。将 Pr 称为与 $(\!|\mathcal{M}|\!)$ 相关联的概率分布。我们要证明 $\Pr(\mathbf{X},\mathbf{Y} \mid \mathbf{Z}) = \Pr(\mathbf{X} \mid \mathbf{Z}) \odot \Pr(\mathbf{Y} \mid \mathbf{Z})$。

准备所需要素。令 $\mathcal{R}$ 为子图，其节点是在从 $\mathcal{M}^{\mathsf{at}}$ 中移除所有标记为$\mathbf{Z}$的边后所得图中，与盒 $\mathsf{b}^X$ （对应于 $X \in \mathbf{X}$）相连的那些节点。容易验证， $\mathcal{R}$ 是一个子网。我们将 $\mathcal{M}^{\mathsf{at}}$ 展开为 $\mathrm{Split}_{\mathcal{R}}(\mathcal{M}^{\mathsf{at}}) = \mathrm{Cut}(\mathcal{R},\mathcal{S})$，它与 $\mathcal{M}^{\mathsf{at}}$ 具有相同的语义，因此也与 $\mathcal{M}$ 相同。

请注意，名称集合$(\mathcal{R}) \cap \mathbf{Y} = \emptyset$: 否则，引理5.1 将给出一条标记为 $Y \in \mathbf{Y}$ 的路径，连接 $\mathcal{R}$ 和$\mathsf{b}^Y$ 中的一条边 $\mathcal{S}$；矛盾，因为 $\mathcal{R}$ 和 $\mathcal{S}$ 之间的边都标记为$\mathbf{Z}$。同样，名称集合$(\mathcal{S}) \cap \mathbf{X} = \emptyset$。

图形证明。 在本证明中，我们考虑四个bpn，见图8 的示意：首先是切割$(\mathcal{R},\mathcal{S})$，然后是切割$(\mathcal{R}',\mathcal{S})$，即用$\mathbf{X}$对切割$(\mathcal{R},\mathcal{S})$所作的弱化，接着是切割$(\mathcal{R},\mathcal{S}')$，即用$\mathbf{Y}$对切割$(\mathcal{R},\mathcal{S})$所作的弱化，最后是切割$(\mathcal{R}',\mathcal{S}')$，即同时用$\mathbf{X}$ 和$\mathbf{Y}$所作的弱化。下文中，我们用图8中的相应图示来表示 $\mathcal{R}$、 $\mathcal{R}'$、 $\mathcal{S}$ 和 $\mathcal{S}'$（例如 

即 $\mathcal{R}'$）。

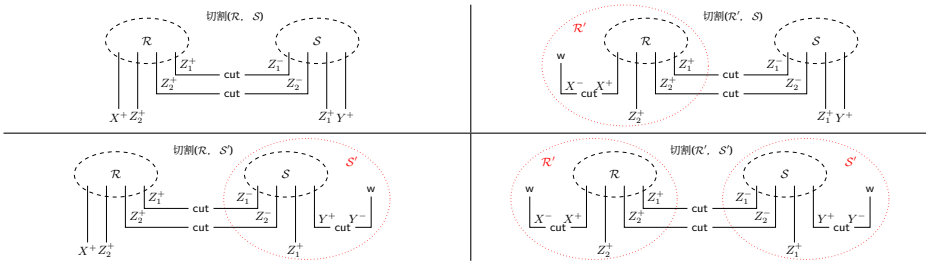



**图 8.** bpn 在定理 7.1 的证明中所考虑的

根据语义的组合性（定理5.2），对这四个bpn:

$$\Pr(\mathbf{X},\mathbf{Y},\mathbf{Z}) = \big(\!\!\big|\ \mathcal{R}\ \big|\!\!\big) \odot \big(\!\!\big|\ \mathcal{S}\ \big|\!\!\big)$$

$$\Pr(\mathbf{X},\mathbf{Z}) = \sum_{\mathbf{Y}} \Pr(\mathbf{X},\mathbf{Y},\mathbf{Z}) = \big(\!\!\big|\ \mathcal{R}\ \big|\!\!\big) \odot \big(\!\!\big|\ \mathcal{S}\ \big|\!\!\big)$$

$$\Pr(\mathbf{Y},\mathbf{Z}) = \sum_{\mathbf{X}} \Pr(\mathbf{X},\mathbf{Y},\mathbf{Z}) = \big(\!\!\big|\ \mathcal{R}\ \big|\!\!\big) \odot \big(\!\!\big|\ \mathcal{S}\ \big|\!\!\big)$$

$$\Pr(\mathbf{Z}) = \sum_{\mathbf{X},\mathbf{Y}} \Pr(\mathbf{X},\mathbf{Y},\mathbf{Z}) = \big(\!\!\big|\ \mathcal{R}\ \big|\!\!\big) \odot \big(\!\!\big|\ \mathcal{S}\ \big|\!\!\big)$$

（因为求和消去对应于弱化，这可由复合性得出）。

固定取值$\mathbf{x}$、$\mathbf{y}$ 和$\mathbf{z}$。称$\mathbf{z}_r$ 为出现在 $\mathcal{R}$结论中的变量$\mathbf{z}$ （的取值），称$\mathbf{z}_s$ 为出现在 $\mathcal{S}$ 结论中的变量$\mathbf{z}$（的取值）。那么：

$$\begin{aligned}
\Pr(\mathbf{x}\mid\mathbf{z})\Pr(\mathbf{y}\mid\mathbf{z}) &= \frac{\Pr(\mathbf{x},\mathbf{z})}{\Pr(\mathbf{z})}\frac{\Pr(\mathbf{y},\mathbf{z})}{\Pr(\mathbf{z})} \\
&= \frac{\big(\!\!\big|\ \mathcal{R}\ \big|\!\!\big)_{\mathbf{x},\mathbf{z}_r} \big(\!\!\big|\ \mathcal{S}\ \big|\!\!\big)_{\mathbf{z}_s} \big(\!\!\big|\ \mathcal{R}\ \big|\!\!\big)_{\mathbf{z}_r} \big(\!\!\big|\ \mathcal{S}\ \big|\!\!\big)_{\mathbf{y},\mathbf{z}_s}}{\big(\!\!\big|\ \mathcal{R}\ \big|\!\!\big)_{\mathbf{z}_r} \big(\!\!\big|\ \mathcal{S}\ \big|\!\!\big)_{\mathbf{z}_s} \big(\!\!\big|\ \mathcal{R}\ \big|\!\!\big)_{\mathbf{z}_r} \big(\!\!\big|\ \mathcal{S}\ \big|\!\!\big)_{\mathbf{z}_s}} \\
&= \frac{\big(\!\!\big|\ \mathcal{R}\ \big|\!\!\big)_{\mathbf{x},\mathbf{z}_r} \big(\!\!\big|\ \mathcal{S}\ \big|\!\!\big)_{\mathbf{y},\mathbf{z}_s}}{\big(\!\!\big|\ \mathcal{R}\ \big|\!\!\big)_{\mathbf{z}_r} \big(\!\!\big|\ \mathcal{S}\ \big|\!\!\big)_{\mathbf{z}_s}} \\
&= \frac{\Pr(\mathbf{x},\mathbf{y},\mathbf{z})}{\Pr(\mathbf{z})} = \Pr(\mathbf{x},\mathbf{y}\mid\mathbf{z})
\end{aligned}$$

因此，$\mathbf{Pr(X,\ Y\ \mid Z)} = \mathbf{Pr(X\ \mid Z)} \odot \mathbf{Pr(Y\ \mid Z)}$，如所愿。

## 8 高效计算语义

我们来考察计算一个bpn $\mathcal{R}$ 语义的代价，该 bpn 的定义见第5.2节。记 $m$ 为 $\mathcal{R}$ 中盒的数量，并记 $n$ 为其基数

名称集合(盒($\mathcal{R}$))。回顾注5.1，根据定义 $(\!|\mathcal{R}|\!)$ 中的公式计算5.2 需要计算并存储 $\mathbf{O}(m \cdot 2^n)$ 个值。不过，如果我们能够将一个*bpn*因式分解为若干较小网络的复合，而这些网络的解释计算代价更低，那么我们还能进一步降低代价。

在证明论中，将证明分解为较小分量的自然方式，是将其分解为通过切割组合在一起的子证明。

例 *8.1*（路线图）。取bpn $\mathcal{R}_D$ （图9）。要通过定义5.2来解释它，我们需要计算 $2^5$ 的值，然后进行边缘化。不过，仅通过公理展开，我们就能将 $\mathcal{R}_D$ 改写为 $\mathcal{R}'_D$ （图10），由于复合性，我们能够以较低的成本对其进行解释，本节将对此加以说明。请注意 $(\!|\mathcal{R}'_D|\!) = (\!|\mathcal{R}_D|\!)$。

### 8.1 切网

在线性逻辑文献中，将证明网分解为由切割节点连接在一起的子网，称为割网 [14]。

**定义 8.1（割网）**。我们用 $\mathcal{R} = \mathrm{Cut}(\mathcal{R}_1, \ldots, \mathcal{R}_n)$ 表示证明网 $\mathcal{R}$，其中给定了一个划分，将其分为子网 $\mathcal{R}_1, \ldots, \mathcal{R}_n$ 以及切割节点，这些节点将子网分隔开（每个此类切割的边都是不同子网的结论）。

切割$(\mathcal{R}_1, \ldots, \mathcal{R}_n)$ 是一个割网，如果具有节点 $\{\mathcal{R}_1, ..., \mathcal{R}_n$ 的骨架图 $\}$ 并且一条边$(\mathcal{R}_i, \mathcal{R}_j)$ 恰好在 $\mathcal{R}_i$ 与 $\mathcal{R}_j$ 之间至少存在一个切割时才存在，是一棵树。

例 *8.2*（割网）。该 bpn 在图 10 中可被划分为一个具有 3 个分量（及 4 个切）的割网。该 bpn $\mathcal{R}_D$ 在图 9 中是一个具有单一组件的（平凡）割网。

割网是将证明网组织成树状结构的一种方式。请注意，子网的选择是完全任意的：每个子网本身也可以是一个割网。当我们需要显式的树结构时，我们使用如下记法来标明哪个子网被选为根，并将该割网称为有根的。根的选择是任意的：每个子网都可以扮演这一角色。

记法 *8.1*（有根切网）。我们用 $\mathcal{N} = \mathcal{R}(\mathcal{M}_1, \ldots, \mathcal{M}_h)$ 表示将 $\mathcal{N}$ 分解为一个切网，其中子网 $\mathcal{R}$（根）与每个$\mathcal{M}_i$相连，而这些 $\mathcal{M}_i$ 本身也是切网（带有根）。

例 *8.3*（有根切网）。考虑割网 $\mathcal{R}'_D = \mathrm{Cut}(\mathcal{M}_1, \mathcal{M}_2, \mathcal{M}_3)$，见图 10。通过选择 $\mathcal{M}_1$ （相应地 $\mathcal{M}_2$, $\mathcal{M}_3$）作为根，我们可以将此割网写为

$$\mathcal{R}'_D = \mathcal{M}_1(\mathcal{M}_2(\mathcal{M}_3)) \quad \text{or} \quad \mathcal{R}'_D = \mathcal{M}_2(\mathcal{M}_1, \mathcal{M}_3) \quad \text{or} \quad \mathcal{R}'_D = \mathcal{M}_3(\mathcal{M}_2(\mathcal{M}_1))$$

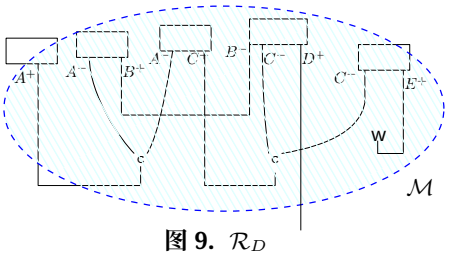



**图 9.** $\mathcal{R}_D$

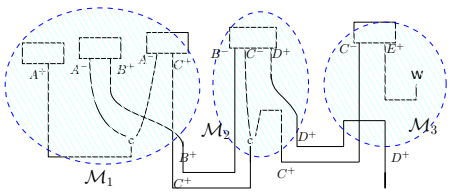



**图 10.** $\mathcal{R}'_D$。注意 $(\!|\mathcal{R}_D|\!) = (\!|\mathcal{R}'_D|\!)$

### 8.2 通过解释进行高效推理

凭借组合性，我们可以按照其树形归纳地解释有根切网。

**定理 8.1（归纳解释）**。设 $\mathcal{R}(\mathcal{N}_1, \ldots, \mathcal{N}_n)$: $\Delta$ 是一个有根切网。注意 $\mathbf{X} = \bigcup_{\mathsf{b} \in \text{盒}(\pi)}$名称集合(b)，我们有：

$$(\!|\mathcal{R}(\mathcal{N}_1, \ldots, \mathcal{N}_n) : \Delta|\!) = \sum_{\mathbf{X} - \mathsf{Nm}(\Delta)} \left( (\!|\mathcal{R}|\!) \odot \bigodot_{i:1}^{n} (\!|\mathcal{N}_i|\!) \right)$$

例 *8.4*（因子化解释）。让我们计算图 $(\!|\mathcal{R}'_D|\!)$中的10 以 $\mathcal{M}_3$ 为根的切网 $\mathcal{M}_3(\mathcal{M}_2(\mathcal{M}_1))$。

| 割网 | 计算 | 所得因子 名称 | |
|---|---|---|---|
| $\mathcal{M}_1$ | $\sum_A \phi^A \odot \phi^B \odot \phi^C$ | $\phi^1(B, C)$ | $ABC$ |
| $\mathcal{M}_2(\mathcal{M}_1)$ | $\sum_B \phi^D \odot \phi^1(B, C)$ | $\phi^2(C, D)$ | $BCD$ |
| $\mathcal{M}_3(\mathcal{M}_2(\mathcal{M}_1))$ | $\sum_{C,E} \phi^E \odot \phi^2(C, D)$ | $\phi^3(D)$ | $CDE$ |

以这种方式计算 $\mathcal{R}'_D$的语义时，所计算的最大中间因子的大小为 $2^3$。这与 $2^5$ 在例8.1中的大小形成对比。

**成本分析**。确定解释成本的关键参数是每个分量中的名称数量，我们称之为宽度（类比贝叶斯网络中的类似概念）。给定一个割网 $\mathcal{N} = \mathsf{Cut}(\mathcal{R}_1、\ldots\ldots、\mathcal{R}_n)$，我们将 $\mathsf{Width}(\mathcal{N})$ 定义为其任一子网的名称集合 $|\mathsf{Nm}(\mathcal{R}_i)|$ 的最大基数减一。

例*8.5*。图9中的割网宽度为4，图10 中的割网宽度为2。

**定理8.2（解释切网的代价）**。设 $\mathcal{N}$ 为一个正切网，$n = |$名称集合$(\mathcal{N})|$表示出现在 $\mathcal{N}$中的名称数量。归纳计算 $(\!|\mathcal{N}|\!)$ 的时间和空间代价为 $\mathbf{O}\left(n \cdot 2^{\mathrm{Width}(\mathcal{N})}\right)$。

**对bpn进行因子分解，以及因子分解算法**。通过反复应用分裂（定义4.1），我们可以从任意证明网构造一个割网。在此基础上，我们可以在bpn 中仿照贝叶斯网络的精确推理算法，构造一个割网，使其根据定理8.1 进行解释后，能够实现所需的计算。例如，图9 中的证明网可以通过先执行由 $\mathcal{M}_1$ 诱导的分裂，再执行由 $\mathcal{M}_2$ 诱导的分裂，展开为图10中的证明网，其代价更低——这对应于变量消元算法 [4]（第 6 章），其消元顺序为（$A$，$B$，$C$）。附录D 概述了如何在bpn中仿照变量消元。

### 8.3 割网与类型化

本文广泛使用分裂（定义 4.1）将一个 bpn $\mathcal{R}$分解为割网 Cut($\mathcal{R}_1$，……，$\mathcal{R}_n$)，它由若干子网 $\mathcal{R}_i$ 通过（可能很多个）切节点连接而成（回顾定义 8.1）。这种图形分解先验上在相继式演算的证明树中没有对应形式。事实上，虽然每个 $\mathcal{R}_i$ 都是一棵证明树 $\pi_i$的像，但相继式演算中没有允许一次执行多重切割的规则，而在证明网中这样做却相当自然。换言之，虽然 $\mathcal{R}$ 肯定是一棵证明树 $\pi$的像，但证明树的归纳结构与这种分解毫无相似之处：可能不存在这样的 $\pi$，其像为 $\mathcal{R}$，并且包含所有 $\pi_i$ 作为子证明。

我们将在此证明，$\mathcal{R}$ 分解为 Cut($\mathcal{R}_1$, ……，$\mathcal{R}_n$) 后，可以通过对 $\mathcal{R}$进行一些重写，将其翻译回相继式演算：$\mathcal{R}$是证明树 $\pi$ 的像，其中包含作为子树的 $\pi_i$，它们的像即为子网 $\mathcal{R}_i$。这为图示推理提供了一个证明论层面的对应形式。实现这一翻译的关键技术结果如下：通过 $\otimes/\parr$-展开（即图 4 中 $\otimes/\parr$-规则的逆过程），我们将割网 $\mathcal{R}$ 变换为一个等价的割网，其中任意两个子网之间至多有一个切割节点。这是通过类型化过程实现的：该过程逐步将公式 $A_1$、……、$A_n$ 上的 $n$ 个切节点转化为一个作用于大公式上的单一切割，该大公式由 $A_1$、……、$A_n$构造而成。举例来说，考虑图 10 中的 $\mathcal{R}'_D$：$\mathcal{M}_1$ 和 $\mathcal{M}_2$之间有两个切，分别作用于公式 $B^+$ 和 $C^+$；$\mathcal{M}_2$ 和 $\mathcal{M}_3$之间也有两个切，分别作用于公式 $C^+$和 $D^+$。执行两次 $\otimes/\parr$-展开后，我们可以得到一个正确的割网，其中 $\mathcal{M}_1$ 和 $\mathcal{M}_2$之间有一个切，作用于公式 $B^+ \otimes C^+$；$\mathcal{M}_2$ 和 $\mathcal{M}_3$之间也有一个切，作用于 $C^+ \parr D^+$。

给定一个割网，其中任意两个子网之间至多有一个切，将其转译为证明树是显而易见的，因为割网的树结构是明确的：只需取与各子网对应的证明树，再用与每个切割节点对应的切规则将它们组合起来。回到我们的例子，下面是一个证明树的开头，它对应于 $\mathcal{R}'_D$ 的展开，其中 $\pi_i$ 是一个对应于 $\mathcal{M}_i$ 的证明树（展开过程中还加入了额外的 $\otimes$ 和 $\parr$）：

$$\dfrac{\dfrac{\dfrac{\pi_1}{\vdash B^+ \otimes C^+} \quad \dfrac{\pi_2}{\vdash B^- \parr C^-, C^+ \parr D^+}}{\vdash C^+ \parr D^+}\ \mathsf{cut}(B^+ \otimes C^+) \quad \dfrac{\pi_3}{\vdash C^- \otimes D^-, D^+}}{\vdash D^+}\ \mathsf{cut}(C^+ \parr D^+)$$

**定理 8.3**。 给定一个割网 $\mathcal{R} = \text{切割}(\mathcal{R}_1, \ldots\ldots, \mathcal{R}_n)$，存在一系列$\otimes/\parr$-展开（图中 $\otimes/\parr$-规则的逆过程，见图 4），对各对节点之间的切 $(\mathcal{R}_i, \mathcal{R}_j)$ 进行操作，使结果成为一个割网 $\mathcal{R}' = \text{切割}(\mathcal{R}'_1, \ldots\ldots, \mathcal{R}'_n)$，其中 每对节点 之间 $(\mathcal{R}'_i, \mathcal{R}'_j)$至多有一个切。

证明（见第 E 节）是构造性的，并给出一种线性算法，将割网转换为任意一对子网之间至多有一个切的割网。对于每一对（$\mathcal{R}'_i$, $\mathcal{R}'_j$），所得唯一切的类型（即 标记该切的公式）由 $R_i$ 与 $R_j$之间起始各切的类型通过 $\parr$ 和 $\otimes$ 获得。

## 参考文献


1. Coecke, B., Spekkens, R.W.：经典与量子贝叶斯推断的图示。《Synth. 186(3)》，651–696（2012）。https://doi.org/10.1007/S11229-011-9917-5, https://doi.org/10 5 2 1007 11229 011 9917 . /s - - - . Danos, V., Regnier, L.：乘法结构。《数理逻辑档案》28，181–203（1989）3. Darwiche, A.：贝叶斯网络。载于：van Harmelen, F., Lifschitz, V., Porter, B.W.（编），《知识表示手册》，《人工智能基础》，第3卷，第467–509页。爱思唯尔（2008）。https://doi.org/10.1016/ S1574-6526(07)03011-8 4. Darwiche, A.：《贝叶斯网络建模与推理》。剑桥大学出版社（2009）5. Ehrhard, T., Faggian, C., Pagani, M.：定量乘法线性逻辑的和积算法。载于：Gaboardi, M., van Raamsdonk, F.（编），第8届形式化计算与演绎结构国际会议，FSCD 2023，2023年7月3日至6日，意大利罗马。LIPIcs，第260卷，第8:1–8:18页。Schloss Dagstuhl - Leibniz-Zentrum f¨ur Informatik（2023）。https://doi.org/10.4230/LIPICS.FSCD. 2023 8 6 8, 10 4230 2023 . https://doi.org/ . /LIPIcs.FSCD. . . Ehrhard, T., Faggian, C., Pagani, M.：在线性 λ 演算中将变量消元作为重写。载于：Vafeiadis, V.（编），《编程语言与系统——第34届欧洲编程研讨会》，ESOP 2025，作为软件理论与实践国际联合会议 ETAPS 2025 的一部分举行，加拿大安大略省汉密尔顿，2025年5月3日至8日，论文集，第I部分。《计算机科学讲义》，第15694卷，第292–321页。施普林格（2025）。https://doi.org/10.1007/978-3-031-91118-7_12, https://doi.org/10.1007/978-3-031-91118-7_127. Ehrhard, T., Tasson, C.：概率性推值调用。《计算机科学逻辑方法》15(1)（2019）。https://doi.org/10.23638/LMCS-15(1:3)2019, https://doi.org/10 2019 8 23638 15 1 3 . /LMCS- ( : ) . Faggian, C., Pautasso, D., Vanoni, G.：精确的高阶贝叶斯网络。《ACM 编程语言汇刊》8(POPL)，2514–2546（2024）。https://doi.org/10. 1145 3632926 9 3632926, 10 1145 / https://doi.org/ . / . Fritz, T., Klingler, A.：范畴概率中的 d-分离判据。《机器学习研究杂志》24，46:1–46:49（2023），https://jmlr.org/papers/v24/ 22-0916.html 10. Girard, J.Y.：线性逻辑。《理论计算机科学》50，1–102（1987）

11. Girard, J.Y.：交互几何 I：系统的一种解释 $F$。载于：Ferro, Bonotto, Valentini, Zanardo（编），逻辑讨论会 '88。North-Holland（1988）12. Girard, J.Y.：迈向交互几何。载于：《计算机科学与逻辑中的范畴》。第69–108页。美国数学学会，普罗维登斯（1989），纯数学专题讨论会第 n°9213. Girard, J.Y.：线性逻辑：语法与语义。载于：Girard, J.Y., Lafont, Y., Regnier, L.（编），《线性逻辑进展》。伦敦数学学会讲义丛书，第222卷，第1–42页（1995）14. Girard, J.Y.：《孤独之地：从逻辑规则到规则逻辑》。《数学结构与计算机科学》**11**(3)，301–506（2001年6月）15. Guerrini, S., Masini, A.：解析 MELL 证明网。《理论计算机科学》**254**(1-2)，317–335（2001）16. Jacobs, B.：结构化概率推理。书稿准备中（2025），
http://www.cs.ru.nl/B.Jacobs/PAPERS/ProbabilisticReasoning.pdf 17. Jacobs, B., Zanasi, F.：贝叶斯学习的谓词/状态变换器语义。载于：Birkedal, L.（编），第三十二届程序语义数学基础会议，MFPS 2016，美国宾夕法尼亚州匹兹堡卡内基梅隆大学，2016年5月23日至26日。《理论计算机科学电子札记》，第325卷，第185–200页。爱思唯尔（2016）。https://doi.org/10.1016/j.entcs.2016038 1809038，
101016201609..https://doi.org/./j.entcs...。Jacobs, B., Zanasi, F.：贝叶斯推理的逻辑要义。载于：《概率编程基础》，第295–332页。剑桥大学出版社（2020）19. Laurent, O.： ´逻辑中的极化研究`博士论文，马赛第二大学´（2002年3月）20. Laurent, O.：极化证明网与 λ-$\mu$演算。《理论计算机科学》**290**3 211161188
2003101016030439750100297 2003(), –(). https://doi.org/./S-()-。Neapolitan, R.：《贝叶斯网络学习》。Prentice Hal（ ）22. Pearl, J.：信念网络中的融合、传播与结构化。《人工智能》**29**(3)，241–288（1986）。
https://doi.org/10.1016/0004-3702(86)90072-X 23. Pearl, J.：智能系统中的概率推理——合理推断网络。摩根·考夫曼（1988）

# 附录

## 一种带箱子的乘法线性逻辑

图11 给出了生成相继式演算证明树的规则，适用于MLL$^{\mathsf{B}}$。它还为每个证明树 $\pi$ 给出其像 $\mathcal{N}$ 作为证明网。从右向左阅读该图，我们得到证明网 $\mathcal{N}$的序列化。

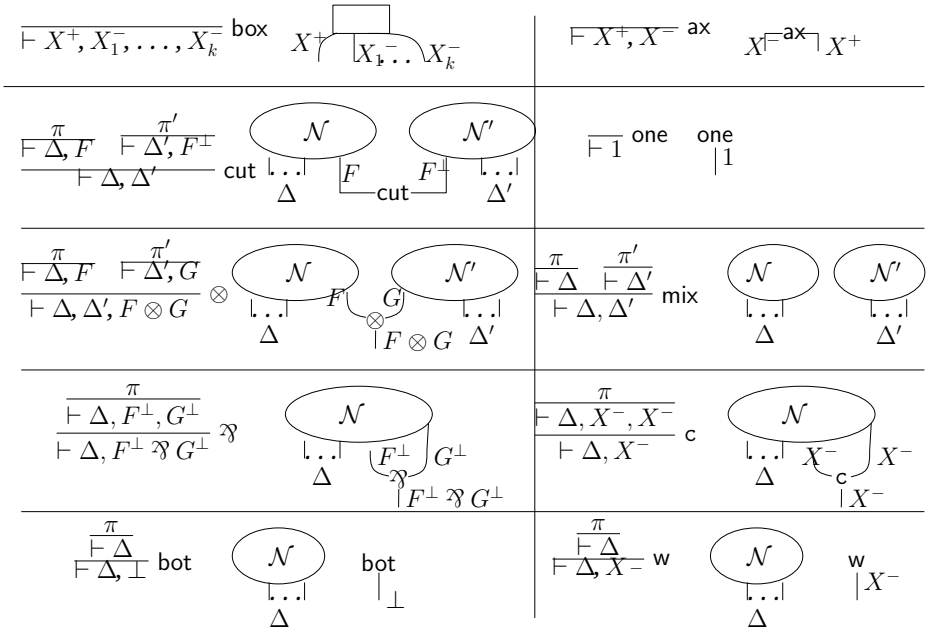



**图 11.** 相继式演算、证明网与序列化

## B 第5.2节的证明

**引理 B.1（引理 5.1，命名路径）**。设 $\pi : \Lambda$ 为一个证明网，并设 $X^+$ 为盒 $\mathsf{b}^X$ 的主名称。则在 $\pi$ 中，每条名称为 $X$ 的边，要么连接到 $\mathsf{b}^X$ 的主结论，要么连接到一个负结论 $X^-$，其所经由的有向路径中所有边的名称均为 $X$。存在唯一这样的路径。

证明。 我们先从两个简单的观察开始。

1. 通过检查证明网语法图 3 并考虑边的极化定向，我们发现，在 $\pi$ 每条边至多有一个

具有相同底层名称的父节点。请注意，唯一没有任何同名父节点（分别为同原子的父节点）的边，要么是盒的主边，要么是 $\pi$ 的负结论。

2. 根据 (1)，回想一下，每个节点至多有一个父节点的有向无环图是一棵树，我们得到以下关键观察：如果我们将 $\pi$ 划分为极大连通子图，且其中的顶点是名称相同的边 $X$，那么每个这样的子图都是一棵有向树，其根对应于盒的主类型，或结论上下文中的一条边 $\Lambda$。

现在设 $X$ 是一个盒的主名称。由于 $\pi$ 是正的，不存在以 $X^-$标记的负结论。由于我们要求 $\pi$ 的主名称两两不同，因此可以得出，在 $\pi$ 中，名称相同的边恰好构成一个 连通分支 $X$。

一个直接的推论是，如果名称 $X$ 被求和消去，那么它必定是一个主名称。

??

**定理 B.1（定理 5.2，图的组合性）**。设 $\mathcal{R} = \text{切割}(\mathcal{R}_1, \mathcal{R}_2)$：$\Delta$ 是一个*bpn*。那么

$$(\!|\mathcal{R}|\!) = \sum_{\mathbf{Z}} \left((\!|\mathcal{R}_1|\!) \odot (\!|\mathcal{R}_2|\!)\right) \qquad \textit{where } \mathbf{Z} = \left(\mathsf{Nm}((\!|\mathcal{R}_1|\!)) \cup \mathsf{Nm}((\!|\mathcal{N}_1|\!))\right) - \mathsf{Nm}(\Delta)$$

证明。令 $\Phi_{\mathcal{M}}$ $\stackrel{\text{def}}{=}$ $\bigodot_{\mathfrak{b} \in \text{盒}(\mathcal{M})} (\!|\mathfrak{b}|\!)$，对每个 $\mathcal{M}$均如此。我们假设每个$\mathcal{R}_i$都有结论 $\Delta_i$。根据定义 5.2，我们可以写成 $(\!|\mathcal{R}_i|\!) = \sum \mathbf{W}_i \, \Phi_{\mathcal{R}_i}$，其中 $\mathbf{W}_i = \mathsf{Nm}(\Phi_{\mathcal{R}_i}) - \mathsf{Nm}(\Delta_i)$ $(i \in \{1, 2\})$。关键是，根据推论 5.1，$\mathbf{W}_1 \cap \mathsf{Nm}(\mathcal{R}_2) = \emptyset$以及 $\mathbf{W}_2 \cap \mathsf{Nm}(\mathcal{R}_1) = \emptyset$。因此，根据公式 (1)，求和与乘积分配开来，我们得到：

$$(\!|\mathcal{R}_1|\!) \odot (\!|\mathcal{R}_2|\!) = \sum_{\mathbf{W}_1} \Phi_{\mathcal{R}_1} \odot \sum_{\mathbf{W}_2} \Phi_{\mathcal{R}_2} = \sum_{\mathbf{W}_1} \sum_{\mathbf{W}_2} (\Phi_{\mathcal{R}_1} \odot \Phi_{\mathcal{R}_2}) = \sum_{\mathbf{W}_1} \sum_{\mathbf{W}_2} \Phi_{\mathcal{R}} \qquad (3)$$

其中我们使用了以下事实：$\text{盒}(\mathcal{R}) = \text{盒}(\mathcal{R}_1) \cup \text{盒}(\mathcal{R}_2)$。现在令 $\mathbf{Y}_i \stackrel{\text{def}}{=} \mathsf{Nm}((\!|\mathcal{R}_i|\!))$；由于 $\mathsf{Nm}(\Phi_{\mathcal{R}_i}) = \mathbf{Y}_i \uplus \mathbf{W}_i$，并且根据推论 5.1，对于$i \neq j$，$\mathbf{W}_i \cap \mathsf{Nm}(\Phi_{\mathcal{R}_j}) = \emptyset$，我们有

$$\mathsf{Nm}(\Phi_{\mathcal{R}}) = (\mathbf{Y}_1 \cup \mathbf{Y}_2) \uplus (\mathbf{W}_1 \uplus \mathbf{W}_2). \qquad (4)$$

令 $\mathbf{Z} \stackrel{\text{定义为}}{=} (\mathbf{Y}_1 \cup \mathbf{Y}_2) - \text{名称集合}(\Delta)$; 由公式 (4) 和推论 5.1, $\text{名称集合}(\Phi_{\mathcal{R}}) - \text{名称集合}(\Delta) = \mathbf{Z} \uplus \mathbf{W}_1 \uplus \mathbf{W}_2$。因此

$$\sum_{\mathbf{Z}} \left((\!|\mathcal{R}_1|\!) \odot (\!|\mathcal{R}_2|\!)\right) = \sum_{\mathbf{Z}} \sum_{\mathbf{W}_1} \sum_{\mathbf{W}_2} \Phi_{\mathcal{R}} = \sum_{\mathsf{Nm}(\Phi_{\mathcal{R}}) - \mathsf{Nm}(\Delta)} \Phi_{\mathcal{R}} \stackrel{\text{def}}{=} (\!|\mathcal{R}|\!)$$

其中我们对$\mathbf{Z}$ 在等式（3）两边进行求和消元。

## 第 6 节的证明

**引理 C.1（引理 6.2）**。给定一个正 *bpn* $\mathcal{R}$，如果 $\mathcal{R} \xrightarrow{w} \mathcal{S}$ 则 $(\!|\mathcal{R}|\!) = (\!|\mathcal{S}|\!)$。

证明。根据定义， $\mathcal{R}$ 有一个子网 $\mathcal{N}$，其形状如图7左侧所示；而 $\mathcal{S}$ 是同一个图，只是将 $\mathcal{N}$ 替换为图7右侧所示的形状。现在使用定义 4.1：对于某个证明网 $\mathcal{D}$，我们有 $\mathsf{Split}_{\mathcal{N}}(\mathcal{R}) = \mathsf{Cut}(\mathcal{D}, \mathcal{N})$，以及 $\mathsf{Split}_{\mathcal{M}}(\mathcal{S}) = \mathsf{Cut}(\mathcal{D}, \mathcal{M})$，因为这两个图仅在 $\mathcal{N}$ 和 $\mathcal{M}$上不同。根据组合性（定理 5.2），我们得到：

$$(\!|\mathcal{R}|\!) = (\!|\mathsf{Split}_{\mathcal{N}}(\mathcal{R})|\!) = (\!|\mathcal{D}|\!) \odot (\!|\mathcal{N}|\!)$$
$$(\!|\mathcal{S}|\!) = (\!|\mathsf{Split}_{\mathcal{M}}(\mathcal{S})|\!) = (\!|\mathcal{D}|\!) \odot (\!|\mathcal{M}|\!)$$

根据定义 5.，我们计算2 $(\!|\mathcal{M}|\!) = \mathbf{1}_{\emptyset}$ 和 $(\!|\mathcal{N}|\!) = \sum_X \phi$，其中使用了与 $\mathcal{N}$ 的盒相关联的条件概率表。但， $\phi$ 是 $X$上的条件概率表， $\sum_X \phi = \mathbf{1}_{\{Y_1;\ \ldots;\ Y_n\}}$。因此：

$$(\!|\mathcal{R}|\!) = (\!|\mathcal{D}|\!) \odot \mathbf{1}_{\{Y_1;\ldots;Y_n\}} = (\!|\mathcal{D}|\!) = (\!|\mathcal{D}|\!) \odot \mathbf{1}_{\emptyset} = (\!|\mathcal{S}|\!)$$

## D bpn中的变量消元

我们在此概述bpn中著名变量消元算法 [4]（第 6 章）的一种类比，该算法用于高效计算贝叶斯网络的语义。其思想如下。假设给定一个正原子bpn $\mathcal{C}$： $\Gamma$，以及名称集合($\mathcal{C}$)–名称集合($\Gamma$)中所有随机变量的消元顺序（$X_1, X_2, \ldots, X_n$）。我们希望以特定方式计算 $\mathcal{C}$ 的语义，即$(\!|\mathcal{C}|\!) = \sum_{X_1, X_2, \ldots, X_n} \left( \bigodot_{\mathfrak{b} \in \text{盒}(\mathcal{D})} (\!|\mathfrak{b}|\!) \right)$。在此计算中，我们希望首先对 $X_1$ 求和消元，即将针对$X_1$ 的求和消元恰好分配到包含它的因子上。这意味着我们首先计算 $\sum_{X_1} \left( \bigodot_{\mathfrak{b} \in \text{盒}(\mathcal{C})\ s.t.\ X_1 \in \text{名称集合}(\mathfrak{b})} (\!|\mathfrak{b}|\!) \right)$

，

这会生成一个新的因子。请注意，这比对 $X_1$ 所有盒进行求和消元的计算成本更低。接着，我们重复这一过程，再对 $X_2$ 进行求和消元：再次将其求和消元操作仅分配给包含它的因子（因此也可能作用于对 $X_1$求和消元时生成的新因子），如此反复，直到对最后一个变量 $X_n$求和消元。作为示例，请看图9 和10： $\mathcal{R}'_D$ 是 $\mathcal{R}_D$ 按顺序（$A$、 $B$、 $C$）进行因子分解的结果。

为此，我们将 $\mathcal{C}$ 展开为一个割网 $\mathcal{C}'$，使得根据定理 $\mathcal{C}'$ 计算8.1 的解释恰好能实现这一点。这样的 $\mathcal{C}'$ 只需通过对精心选择的子网反复进行分裂即可得到（定义4.1）。

**引理D.1**。取 $\mathcal{C}$： $\Gamma$ 一个正的原子*bpn*，以及（$X_1$，……， $X_n$），它是名称集合（$\mathcal{C}$） – 名称集合（$\Gamma$）中所有随机变量的一个消元顺序。存在 $\mathcal{C}'$ 的一个展开 $\mathcal{C}$，使得使用定理计算*8.1*恰好按照给定顺序进行变量消元。

证明。 通过依次对 $X_1$、……、 $X_n$ 应用以下断言即可得到该结果。该断言对应于恰好分离出包含给定 r.v. $X$ 的割网部分，同时跟踪此前的所有计算。

**断言** 取一个 r.v. $X$ 和一个正且原子的割网 $\mathcal{D} = \mathcal{R}(\mathcal{M}_1、\ldots\ldots、\mathcal{M}_k、\mathcal{N}_1、\ldots\ldots、\mathcal{N}_l)$，使得 $X$ 出现在所有 $\mathcal{M}_i$ 的结论中，而不出现在任何 $\mathcal{N}_j$ 的结论中。则存在一个割网 $\mathcal{D}' = \mathcal{R}_1(\mathcal{M}'、\mathcal{N}_1、\ldots\ldots、\mathcal{N}_l)$，使得 $\mathcal{D}' \xrightarrow{R}{}^{*} \mathcal{D}$、 $\mathcal{M}' = \mathcal{R}_2(\mathcal{M}_1、\ldots\ldots、\mathcal{M}_k)$、 $X \notin \mathrm{Nm}(\mathcal{R}_1)$，并且对于 $\mathcal{R}_2$ 中的所有框节点 b， $X \in \mathrm{Nm}(\mathrm{b})$。

直观地说，子网 $\mathcal{M}_1、\ldots、\mathcal{M}_k、\mathcal{N}_1、\ldots、\mathcal{N}_l$ 对应于已经完成的计算，其中 $(\!|\mathcal{M}_i|\!)$ 是至少包含在 $X$上的那些因子。与此同时，根 $\mathcal{R}$包含我们尚未使用的因子。下一步计算是在所有因子 $(\!|M_i|\!)$ 的乘积中对 $X$ 求和消去——并且还要对 $\mathcal{R}$中那些使用了 $X$ 的因子进行相应计算。

因此，可以从完整的 bpn $\mathcal{C}$ 出发，将其视为只有一个子网的割网（即所有因子尚未使用，一切都在根中），然后依次对 $X_1、X_2$等应用上述结果，直到 $X_n$，从而得到与变量消元相关计算所对应的割网。

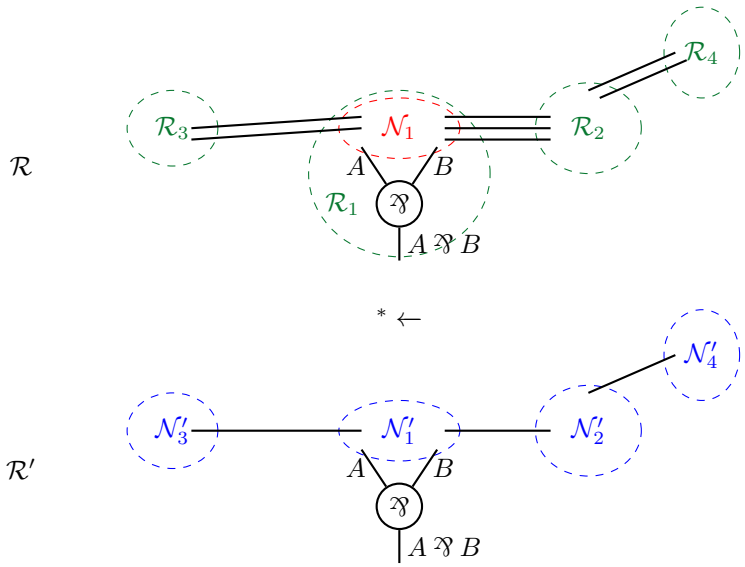
现在我们来证明断言。令 $\mathcal{R}_2$ 为 $\mathcal{R}$ 的最小子网，它包含所有标记为 $X$的边：它由 $X$上的所有切割、ax 和 c，以及所有输入或输出为 $X$的盒组成。定义 $\mathcal{M}' \stackrel{\text{def}}{=} \mathcal{R}_2(\mathcal{M}_1, \ldots \mathcal{M}_k)$（其中包含的切割节点位于 $\mathcal{D}$之间）。令 $\mathcal{D}'$为割网 $\mathcal{D}' \stackrel{\text{def}}{=} \mathrm{Split}_{\mathcal{M}'}(\mathcal{D}) = \mathcal{R}_1(\mathcal{M}', \mathcal{N}_1、\ldots、\mathcal{N}_l)$，其根 $\mathcal{R}_1$ 由 $\mathcal{R}$ 中不属于 $\mathcal{R}_2$ 的节点构成（以及分裂过程中添加的 ax 节点）。根据构造，我们具备所有所需的性质。

## 第 E 节证明：8.3

**定理 E.1（定理 8.3）**。给定一个割网 $\mathcal{R} = \mathrm{Cut}(\mathcal{R}_1, \ldots, \mathcal{R}_n)$，存在一系列 $\otimes/\parr$-展开（即图 $\otimes/\parr 4$ 中 -规则的逆过程），作用于各对 $(\mathcal{R}_i, \mathcal{R}_j)$ 之间的切，使得结果是一个割网 $\mathcal{R}' = \mathrm{Cut}(\mathcal{R}'_1, \ldots, \mathcal{R}'_n)$，且 每对 之间至多有一个切 $(\mathcal{R}'_i, \mathcal{R}'_j)$。

证明。我们对 $\pi$ 在 $\mathcal{R}$上的序列化进行归纳。称一个割网为**适当的**，若其各子网之间至多有一个切割节点。我们用图示说明每种情形，情形取决于 $\pi$ 的最后一条规则是什么。在这些图中，椭圆表示诸如 $\mathcal{R}_1, \ldots, \mathcal{R}_n$ 这样的子图，而它们之间的粗边表示切割节点。除规则交换外，可以假设$\pi$ 中， $mix$-规则之上只有 ax、1、盒或 $mix$-规则，因此 $\pi$的最后一条规则——若它尚非适当——必为 $\parr$-、 $\bot$-、c-、w-、$\otimes$-或切割规则。

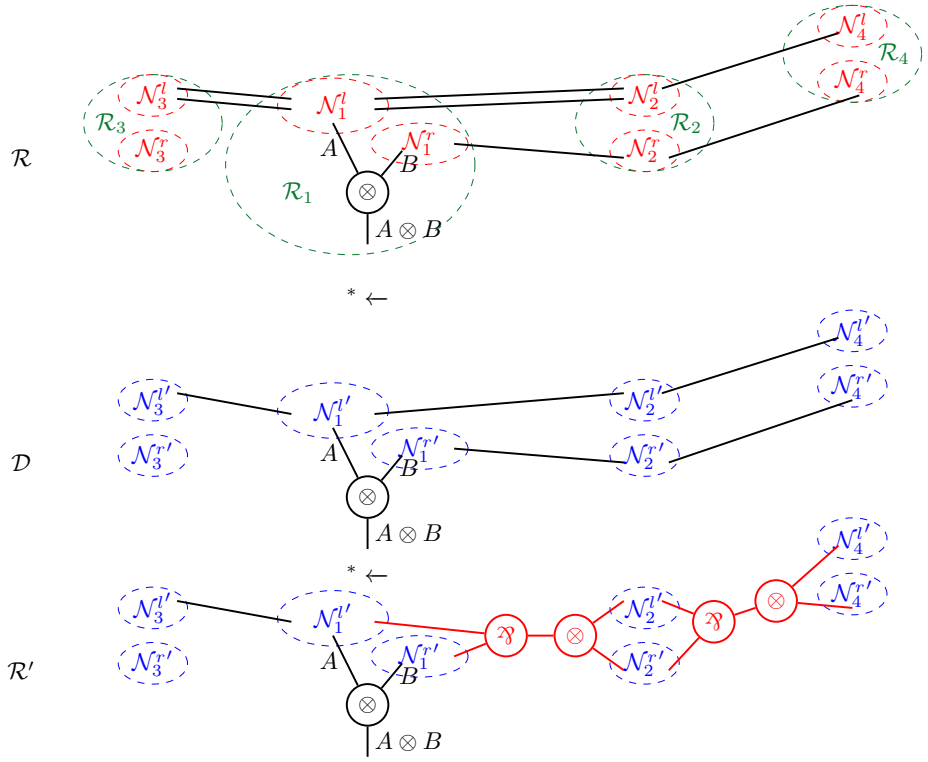
• 假设 $\pi$ 是一个证明 $\rho$，其后接一个 $\parr$-规则；在 $\mathcal{R}$ 中，这对应于一个节点$v$。不失一般性， $v$ 属于 $\mathcal{R}_1$。将 $\mathcal{N}_1$ 称为在 $\mathcal{R}_1$中移除 $v$。根据归纳假设，对于序列化$\rho$的切网 $\mathcal{N} = \mathrm{Cut}(\mathcal{N}_1, \mathcal{R}_2, \ldots, \mathcal{R}_n)$，存在一个正规切网 $\mathcal{N}' = \mathrm{Cut}(\mathcal{N}'_1, \mathcal{N}'_2, \ldots, \mathcal{N}'_n)$，使得 $\mathcal{N}' \to^* \mathcal{N}$。令 $\mathcal{R}' \stackrel{\text{def}}{=} \mathrm{Cut}(\mathcal{N}'_1 \cup \{v\}, \mathcal{N}'_2, \ldots, \mathcal{N}'_n)$，其中 $\mathcal{N}'_1 \cup \{v\}$是通过将$v$ 添加到 $\mathcal{N}'_1$ 而得到的图，并且其边与$v$ 在 $\mathcal{R}_1$中的边相同。于是， $\mathcal{R}'$ 是一个正规切网，并且$\mathcal{R}' \to^* \mathcal{R}$使用与 $\mathcal{N}' \to^* \mathcal{N}$中相同的消去步骤。




- $\pi$ 的最后一条规则是 $\perp$-规则、c-规则或 w-规则的情形，与前一种情形类似。

• 现在假设 $\pi$ 的最后一条规则是 $\otimes$-规则，它位于证明 $\rho^l$ 和 $\rho^r$之间；我们将其在 $\mathcal{R}$ 中对应的节点记为 $v$。不失一般性，假设 $v$ 属于 $R_1$。根据 $\rho^l$和 $\rho^r$对$\mathcal{R}$ 进行划分，可得 $\mathcal{R} = \mathsf{Cut}(\mathcal{N}_1^l \cup \mathcal{N}_1^r \cup \{v\}, \mathcal{N}_2^l \cup \mathcal{N}_2^r, \ldots, \mathcal{N}_n^l \cup \mathcal{N}_n^r)$。根据序列化 $\rho^l$的切网 $\mathcal{N}^l \mathsf{def}= \mathsf{Cut}(\mathcal{N}_1^l, \mathcal{N}_2^l, \ldots, \mathcal{N}_n^l)$上的归纳假设，存在一个正规切网 $\mathcal{N}^{l\prime} = \mathsf{Cut}(\mathcal{N}_1^{l\prime}, \mathcal{N}_2^{l\prime}, \ldots, \mathcal{N}_n^{l\prime})$，使得$\mathcal{N}^{l\prime} \rightarrow^* \mathcal{N}^l$；同理，对 $\rho^r$也可得到 $\mathcal{N}^{r\prime} = \mathsf{Cut}(\mathcal{N}_1^{r\prime}, \mathcal{N}_2^{r\prime}, \ldots, \mathcal{N}_n^{r\prime}) \rightarrow^* \mathcal{N}^r \mathsf{def}= \mathsf{Cut}(\mathcal{N}_1^r, \mathcal{N}_2^r, \ldots, \mathcal{N}_n^r)$。

所需的割网 $\mathcal{R}'$由 $\mathcal{N}^{l\prime}$和 $\mathcal{N}^{r\prime}$按如下方式得到。考虑$\mathcal{D} \overset{\text{定义}}{=} \mathsf{Cut}(\mathcal{N}_1^{l\prime} \cup \mathcal{N}_1^{r\prime} \cup \{v\}, \mathcal{N}_2^{l\prime} \cup \mathcal{N}_2^{r\prime}, \ldots\ldots, \mathcal{N}_n^{l\prime} \cup \mathcal{N}_n^{r\prime})$。虽然 $\mathcal{D}$是一个满足 $\mathcal{D} \rightarrow^* \mathcal{R}$的割网（例如先执行 $\mathcal{N}^{l\prime} \rightarrow^* \mathcal{N}^l$中的步骤，再执行$\mathcal{N}^{r\prime} \rightarrow^* \mathcal{N}^r$中的步骤），但它未必是正确的。事实上，某些$\mathcal{N}_i^{l\prime} \cup \mathcal{N}_i^{r\prime}$和 $\mathcal{N}_j^{l\prime} \cup \mathcal{N}_j^{r\prime}$之间可能有两条切割，其中一条来自 $\mathcal{N}^{l\prime}$，另一条来自 $\mathcal{N}^{r\prime}$。由于 $\mathcal{D}$ 的骨架图是一棵树，不妨利用对称性假设，组件 $i$ 比 $j$更靠近第一个组件——即包含 $v$ 的组件。对于每一对这样的切割，我们应用一次 $\parr - \otimes$ 切割展开，在组件 $i$ 中放置一个 $\parr$-节点，并在组件 $j$中放置一个$\otimes$-节点。所得分量是正确的，因为在 $\mathcal{D}$中，从 $\mathcal{N}_j^{l\prime}$到 $\mathcal{N}_j^{r\prime}$的唯一路径会经过 $v$，也就会经过这两条合并的切割。对所有可能的 $i$和 $j$都这样处理后，就得到了所需的 $\mathcal{R}'$，其中 $\mathcal{R}' \rightarrow^* \mathcal{D} \rightarrow^* \mathcal{R}$。




• 当 $\pi$ 的最后一条规则是切割规则，且其对应的节点并非某些 $\mathcal{R}_i$ 与 $\mathcal{R}_j$ 之间的切割时，情况与 $\otimes$ 情形类似。

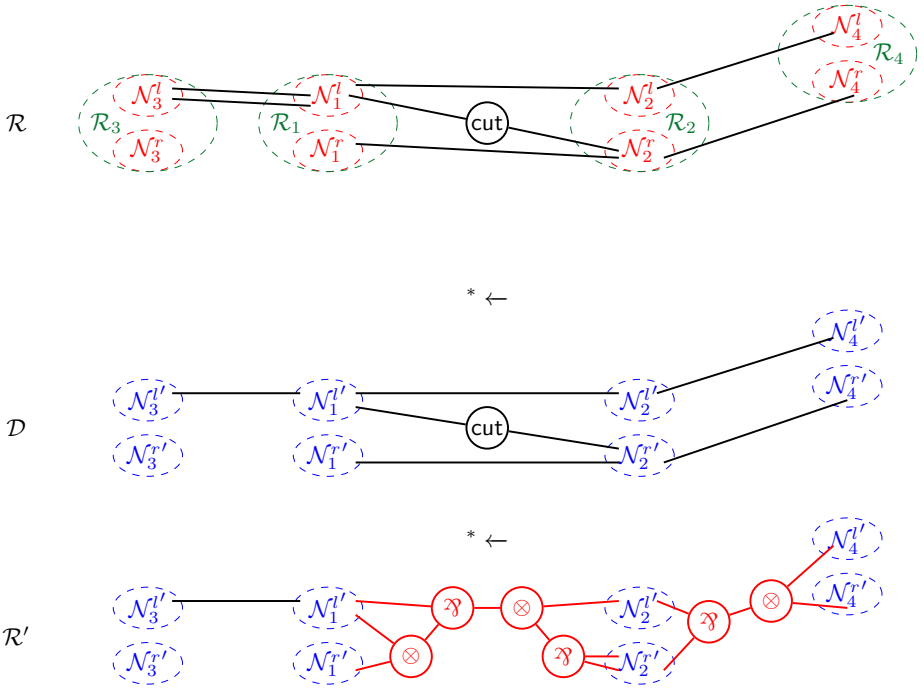
• 最后一种情况是： $\pi$ 的最后一个规则是证明 $\rho^l$与 $\rho^r$之间的切割规则，其中与该切割对应的节点 $v$ ，根据对称性，位于$\mathcal{R}_1$ 与 $\mathcal{R}_2$之间。我们的推理与 $\otimes$ 情况类似，但最后还需增加一步。根据 $\rho^l$和 $\rho^r$对 $\mathcal{R}$进行划分，我们有 $\mathcal{R} = \mathsf{Cut}(\mathcal{N}_1^l \cup \mathcal{N}_1^r, \mathcal{N}_2^l \cup \mathcal{N}_2^r, \ldots, \mathcal{N}_n^l \cup \mathcal{N}_n^r)$。特别地， $v$是一次切割，其一个前提位于 $\mathcal{N}_1^l$中，另一个位于 $\mathcal{N}_2^r$中。根据序列化 $\rho^l$的切网 $\mathcal{N}^l \mathsf{def}= \mathsf{Cut}(\mathcal{N}_1^l, \mathcal{N}_2^l, \ldots, \mathcal{N}_n^l)$上的归纳假设，存在一个正规切网 $\mathcal{N}^{l\prime} = \mathsf{Cut}(\mathcal{N}_1^{l\prime}, \mathcal{N}_2^{l\prime}, \ldots, \mathcal{N}_n^{l\prime})$，使得 $\mathcal{N}^{l\prime} \to^* \mathcal{N}^l$；类似地，可以得到 $\mathcal{N}^{r\prime} = \mathsf{Cut}(\mathcal{N}_1^{r\prime}, \mathcal{N}_2^{r\prime}, \ldots, \mathcal{N}_n^{r\prime}) \to^* \mathcal{N}^r \mathsf{def}= \mathsf{Cut}(\mathcal{N}_1^r, \mathcal{N}_2^r, \ldots, \mathcal{N}_n^r)$。

所需的割网 $\mathcal{R}'$由 $\mathcal{N}^{l\prime}$和 $\mathcal{N}^{r\prime}$按如下方式得到。考虑$\mathcal{D}_{\text{定义}}= \mathsf{Cut}(\mathcal{N}_1^{l\prime} \cup \mathcal{N}_1^{r\prime}, \mathcal{N}_2^{l\prime} \cup \mathcal{N}_2^{r\prime}, \ldots\ldots, \mathcal{N}_n^{l\prime} \cup \mathcal{N}_n^{r\prime})$。这个割网确实可化简为 $\mathcal{R}$，但它未必是适当的：不仅某些 $\mathcal{N}_i^{l\prime} \cup \mathcal{N}_i^{r\prime}$和 $\mathcal{N}_j^{l\prime} \cup \mathcal{N}_j^{r\prime}$之间可能有两个切割，其中一个来自 $\mathcal{N}^{l\prime}$，另一个来自$\mathcal{N}^{r\prime}$；切割节点 $v$还可能在 $\mathcal{N}_1^{l\prime} \cup \mathcal{N}_1^{r\prime}$和 $\mathcal{N}_2^{l\prime} \cup \mathcal{N}_2^{r\prime}$之间引入第二个或第三个切割。第一种情况可能发生在一对 $(i, j) \neq (1, 2)$ 中，处理方法与上文的 $\otimes$情况完全相同。因此，我们将重点讨论这样一种情况：$\mathcal{N}^{l\prime}$和 $\mathcal{N}^{r\prime}$中前两个分量之间已经存在切割。我们有两种

子情况：要么 $\mathcal{N}^{l\prime}$和 $\mathcal{N}^{r\prime}$中只有一个（假设是 $\mathcal{N}^{l\prime}$）有这样的切割；要么两者都有。

如果只有 $\mathcal{N}^{l\prime}$在 $\mathcal{N}_1^{l\prime}$和 $\mathcal{N}_2^{l\prime}$之间有一个切割节点，那么从 $\mathcal{N}_2^{l\prime}$到 $\mathcal{N}_2^{r\prime}$的唯一路径会经过 $v$。因此，我们可以应用一个 $\parr-\otimes$切割展开，在 $\mathcal{N}_1^{l\prime}$中放置一个 $\parr$节点，并在 $\mathcal{N}_2^{l\prime}\cup\mathcal{N}_2^{r\prime}$中放置一个 $\otimes$节点，从而得到所需的 $\mathcal{R}'$。

现在，假设在 $\mathcal{N}_1^{l\prime}$与 $\mathcal{N}_2^{l\prime}$之间存在切节点 $u_l$，并且在$\mathcal{N}_1^{r\prime}$与 $\mathcal{N}_2^{r\prime}$之间存在一个 $u_r$。在这种情况下，我们执行以下两个 $\parr-\otimes$切割展开步骤。首先，我们展开切割 $v$与 $u_r$，方法是在 $\mathcal{N}_1^{l\prime}\cup\mathcal{N}_1^{r\prime}$中放入一个 $\otimes$-节点，并在 $\mathcal{N}_2^{r\prime}$中放入一个$\parr$-节点，同时加入一个新的切割节点 $v'$。由于从 $\mathcal{N}_1^{l\prime}$到 $\mathcal{N}_1^{r\prime}$的路径必须经过 $v$，该结果是正确的。然后，我们展开 $v'$与 $u_l$，方法是在$\mathcal{N}_1^{l\prime}\cup\mathcal{N}_1^{r\prime}\cup\{\otimes\}$中放入一个 $\parr$-节点，并在 $\mathcal{N}_2^{l\prime}\cup$中放入一个 $\otimes$- 节点（$\mathcal{N}_2^{r\prime}\cup\{\parr\}$）。同样，由于从 $\mathcal{N}_2^{l\prime}$ 到 $\mathcal{N}_2^{r\prime}\cup\{\parr\}$ 的路径必须经过 $v'$，该结果是正确的。这就给出了一个正规切网$\mathcal{R}'$，使得 $\mathcal{R}'\to^*\mathcal{D}$，并完成了该证明。[3]




[3] $u_l$ 和 $u_r$ 之间表面上的不对称并不重要：也可以先用 $u_l$ 展开 $v$，再用 $u_r$展开，从而得到另一个可归约为 $\mathcal{R}$的正规切网。

## F bpn 伪影判据

我们所说的证明网始终是指带有概率盒且命名良好的证明网，即标记这些盒的正结论的原子两两不同。

待办：正文中也要修改

我们回顾引理 5.1

**引理 F.1（连接树式性质）**。令 $\mathcal{R}$：$\Lambda$ 为一个正的原子证明网。将 $\mathcal{R}$ 限制在标记为同一原子 $X$ 的边上，得到一棵有向树，其中

*1.* 根是盒 $\mathsf{b}^X$ 的主结论；*2.* 叶要么是 $\mathcal{R}$ 的正结论，要么是一个盒的输入。

### F.1 有向路径的性质

下面汇总一些有向路径的技术性质。根据极化正确性（引理 3.1），有以下结论：直接给出证明很容易，而且这样做可能更好。

**命题 F.1（有向环）**。设 $\mathcal{R}$是一个原子网。 $\mathcal{R}$存在一个开关环，当且仅当 $\mathcal{R}$ 存在一个有向环（沿着边的定向）。

**引理 F.2（有向路径）**。设 $\mathcal{R}$ 是一个原子网（不一定正确）。

– 一条以$\mathsf{b}^X$ 为源的 $X$-路径不能进入环。– 给定一条 $X$-路径，只有其源节点和目标节点可以是盒。– 任意两条连续的边 $\xrightarrow{e_1}\xrightarrow{e_2}$ 名称不同，当且仅当公共顶点是一个盒。在这种情况下， $e_2$ 是该盒的主结论。

由此可得：

**推论 F.1（盒环）**。设 $\mathcal{R}$ 是一个带有有向环$r$的原子网。如果 $r$ 经过至少一个盒，那么 $r$ 必然由$k > 1$ 条有向路径拼接而成，其中每个 $r^{X_i}$ 都是一条以$\mathsf{b}^{X_i}$为源、以$\mathsf{b}^{X_{i+1}}$ 为目标的 $X_i$-路径（其中 $X_{k+1} = X_1$）。

### F.2 原子 bpn 的伪影判据

**定义 F.1（贝叶斯证明网（bpn））**。一个证明网 $\mathcal{R}$：$\Delta$ 是贝叶斯的，当且仅当

*1.* 结论中的所有原子 $\Delta$ 都是正的；或 *2.* $\mathcal{R}$ 是满足（*1*）的证明网的子网。

注 *F.1.* 注意，如果 $\mathcal{R}$的负结论中没有任何一个名称同时也是某个盒的主名称，那么 $\mathcal{R}$ 就是一个bpn。事实上，我们总能得到一个正证明网，使得 $\mathcal{R}$ 是其子网。如果 $\mathcal{R}$ 的所有结论都是原子的，我们执行如下操作：

– 将所有由同一 $X^-$ 标记的负结论 $e_1\colon X^-, \ldots, e_k\colon X^-$ 收集在一起，作为单结论 $e\colon X^-$的收缩的前提。– 添加一个单结论为 $f\colon X^+$的盒$\mathsf{b}^X$，以及 $f\colon X^+$ 与$e\colon X^-$之间的一个切割。

如果 $\mathcal{R}$ 有任何非原子结论 $F$，我们先将每个这样的 $F$ 与 $F^\perp$ 的展开进行切割，然后照上述方法继续。

是否有必要定义结论的原子化，尚可斟酌。

**定义 F.2（伪影）**。给定一个结论为 $\Delta$ 的原子证明网 $\mathcal{R}$，其$\widehat{\ }$伪影闭包 $\mathcal{R}$ 定义为：对每个满足 $X^- \in \Delta$ 且 $\mathsf{b}^X$ 属于 $\mathcal{R}$ 的名称 $X$，执行以下操作：

– 将所有标记为 $X^-$ 的负结论 $e_1\colon X^-, \ldots, e_k\colon X^-$ 收缩在一起。令 $e\colon X^-$ 为收缩所得的结论。– 如果至少存在一个正结论 $f\colon X^+ \in \Delta$，通过一条 $X$-路径与 $\mathsf{b}^X$ 相连，则在 $e\colon X^-$ 和 $f\colon X^+$ 之间添加一个切割。否则，$\smile\mathcal{R}$ 未定义。

如果 $\mathcal{R}$ 有定义且是一个证明网，则 $\smile\mathcal{R}$ 满足伪影检验。

**命题 F.2**。一个原子证明网 $\mathcal{R}$ 满足伪影检验，当且仅当 $\mathcal{R}$ 是一个 *bpn*。

$\widehat{\ }$证明。仅当。根据假设，$\mathcal{R}$ 是一个证明网。对于它的所有负结论，我们按照注释 F.1 中的方法进行，从而得到一个包含 $\mathcal{R}$ 的正证明网。

若。假设存在一个正证明网 $\mathcal{N}$，其中 $\mathcal{R}$ 是一个子网。我们证明，若 $\widehat{\mathcal{R}}$ 有一个切换环，则 $\mathcal{N}$也有一个切换环，这与它是证明网这一事实矛盾。

根据命题 F.1，$\widehat{\mathcal{R}}$ 有向环 $r$，而该环（由于 $\mathcal{R}$ 是一个证明网）必然经过某个伪影切。令 $f\colon X^+$ 为此类切割的正边。根据构造，存在一条从 $X$-路径，从 $\mathsf{b}^X$ 到 $f\colon X^+$。因此，根据推论 F.1，该有向环经过 $k > 1$ 个盒。假设访问顺序为$\mathcal{C} = \mathsf{b}^{X_1} \to \ldots \to \mathsf{b}^{X_k} \to \mathsf{b}^{X_1}$。注意，环中的每个 $\mathsf{b}$ 都有一条负边，其名称与先前访问的盒相同。

在 $\mathcal{N}$中，我们有相同的盒。注意，对于 $\mathcal{C}$ 中每一对相邻的盒$\mathsf{b}^X \to \mathsf{b}^Y$，根据5.1中的联合树性质（引理 $\mathcal{N}$，在 $X$中存在一条 $e$-路径，将负结论 $X^-\colon \mathsf{b}^Y$ 与$\mathsf{b}^X$连接起来。所有这类路径的拼接在 $\mathcal{N}$中形成一个有向环。