arXiv:2602.06272v1 [math.LO] 6 Feb 2026

PBNF-TRANSFORM 作为命题演算的一种表述，I

PELLE BROOKE BORGEKE

摘要。本文是一系列文章中的一篇，展示了如何使用代数多项式计算命题陈述：将代数多项式作为联结词的符号，并称这些符号为算子。这些多项式源自对偶原理与析取布尔范式（DBNF）之间的变换；若在单位正方形中进行几何化，并采用简单的模 2 代数方法，便会得到这些多项式。我们将此称为 PBNF-变换。PBNF 是多项式布尔范式（Polynomial Boolean Normal Form）的缩写，因为这里涉及的这些族都以 DBNF 为基础。在本系列的第一篇论文中，我们展示了如何将命题双射映射到不同的多项式族 $g = g(p_j,\ q_k,\ a)$，这些族属于 $\mathcal{H}(g)$，我们称之为 PBNF之屋。我们还可以用 PBNF 替代逻辑联结词，因为多项式实际上是这些联结词的几何化；两种系统是同构的。这种命题演算（PC）表述的优点是，相关方法几乎变得不言自明。无需公理，也无需真值表，只需列出多项式（它们本身便足以说明问题）；唯一的推理规则是代入规则。

1. 引言：基础知识

我们将在此使用多项式布尔范式（简称 PBNF），对经典命题演算（具有 2 个真值，即 1 或 0）进行数学表述。借助标准的数学技巧，我们有效地处理数理逻辑中的各种问题，例如证明命题和定理，甚至发现新的命题和定理。我们的观点以变换理论为基础，因此使用对偶空间进行 PBNF-变换，从 DBNF（析取布尔标准形）的表述及算子空间中的对偶进行变换。由此得到布尔多项式，即模2代数，它们只使用一条推理规则，也就是代入规则。逻辑中的典型公理并非必需，因为该方法采用单位正方形的几何学。事实上，如果我们转而提及在此证明的一个定理，就会发现代换可视为该定理所给出的线性过程；这样一来，系统没有逻辑推理规则。实际上，PBNF-变换也可以直接用作多项式算子，而非逻辑

日期：2026年2月5日。2010 数学主题分类。主要分类。关键词与短语。算子、范式、对偶化、PBNF-变换、多项式族。

联结词，因为 PBNF 是对联结词的一种同构几何化。其思路是将 Church [3], 等人所使用的对偶原理联系起来；这些原理是不同算子空间之间的变换, 在此称为 $\mathcal{OP}$ （索引为 $_*$ 表示标准空间，索引为 $_{**}$ 表示补空间）

$$\mathcal{OP}_* \ni (\wedge, \vee, \Rightarrow, f) \mapsto (\vee, \wedge \not\Leftarrow, t) \in \mathcal{OP}_{**}.$$

如果我们以析取布尔范式来看这种变换（在其他 BNF 中，我们用x 和 y 表示 p和 q ），例如取上面的合取（$x \wedge y$），便得到 $\mapsto$ （$x \wedge y$） $\vee$ （$x \wedge y'$） $\vee$ （$x' \wedge y$）。下一步是将 DBNF 变换为多项式族 g（p_i，q_j，a_0）$\in \mathcal{H}$（g），我们称之为 PBNF之屋。然后我们可以写成

$$p \wedge q \mapsto pq \in g(p, q, 1) \mapsto pq + p(q+1) + (p+1)q \tag{1.1}$$

$$= (p+1)(q+1) + 1 \in g(p', q', 1'), \tag{1.2}$$

现在它已处于 PBNF 形式。这种对偶标准形-多项式变换[1]——我们称之为 PBNF-变换——给出了模 2 的布尔代数，将命题演算转化为代数计算。映射 f 将 $\mathcal{OP}$中的论元（这里包括算子类、空类（p，q、操作数或字母）、单一算子和二元算子、单一命题及复合命题，我们用 $S_{c(k,l)}$，表示）映射为线性和二次多项式形式，使我们能够立即用 $\mathbb{Z}_2$中的代数方法，以简单的方式计算命题。我们可以根据命题需要，在之屋中改用另一种多项式族。例如，可以选择某个多项式族的补、重新命名真值，或为算子使用不同的输入，并将其写作向量$op_{\boldsymbol{x}(x_1,x_2,x_3,x_4)}$ ，或直接写作 $op_{\boldsymbol{x}}$。输入（p_i，q_j）由选择器 $\mathcal{S}$决定，映射到 $\mathcal{H}$（g）中的（二元）多项式族。以下协议（规则和条件）概述了具体做法。（（$\cdot$）：全称量词，$[\cdot]$：存在量词，$|$：异或，。：$=$（$\cdot$），内层括号。）

$$(g)[\mathcal{H}][f][a_j][p_j, q_k] \ \big(g \in \mathcal{H}, f : \mathcal{OP} \to (\mathbb{Z}_2 | \mathcal{H}(\mathbb{Z}_2)), f(p.op_{\boldsymbol{x}}\ q) \tag{1.3}$$

$$= g(p_j, q_k, a_0) = a_1 pq + a_2 p + a_3 q + a_4 = (1|0, \left|g(p_i, q_j, a_0)(\mathbb{Z}_2)\right| \hookrightarrow \mathcal{OP}), \tag{1.4}$$

$$a_{0 \le n \le 4} = 0 | 1, i = 1, 2, 3, 4 \wedge j = 1, 2. \tag{1.5}$$

我们注意到，f 的值域在 $\mathbb{Z}_2$ 和 $g \in \mathcal{H}$中均为实数值。若不得到重言式（**1**）或矛盾式（**0**），则可对 $\mathbb{Z}_2$ 计算多项式（$a_1 pq + a_2 p + a_3 q + a_4$）（**1**，**0**）。它也可用于逆作用（此处常称为纤维 或拉回） g^{-1}（$a_1 pq + a_2 p + a_3 q + a_4$） $\hookrightarrow \mathcal{OP}$。例如，我们可能会想知道以下命题是否相等

$$(p \wedge q) \vee r = p \wedge (q \vee r).$$

[1]Stoll [8] 和 Hal- mos [4], 曾在一些例子中朝这个方向做过尝试；在 Bergmann [2], 中，我们还发现了另一种使用最小值和最大值计算的算术方法。

等式两边的变换都属于族 g（p，q，1）：

$$\mapsto (p^S+1)(q^S+1)+1=p^Sq^S.$$

代换后的多项式

$$(pq+1)(r+1)+1=p(q+1.r+1+1). \tag{1.6}$$

这些命题并不相等，因为它们在同一族中对应不同的多项式。解出 (1.6) 并化简后，我们得到 $r=pr$；对于.....的旋转，真命题

$$(p\wedge q)\vee(p\wedge r)=p\wedge(q\vee .p\wedge r).$$

现在，我们可以通过令 $r=q$，代回字母 p 和 q，得到 [2]

$$(p\wedge q)\vee(p\wedge q)=p\wedge(q\vee .p\wedge q)$$

这也为真。PBNF 是一种典范形，这意味着逻辑等价的两个公式会转换为相同的 PBNF，从而能轻松判断两个公式是否相同，便于自动定理证明。例如，DBNF 就不是这样：$p\vee\neg p$ 是 DBNF，但并未化简为 1，尽管它们在逻辑上等价。在 PBNF 中，计算多项式时，我们得到 [3]，其中加法是模 2 加法，且 $p^2=p$,

$$(p\vee\neg p)\mapsto(p+1)p+1=p^2+p+1=p+p+1=1\in g(p,q,1).$$

或者我们可以看到，$(p+1)p=0$ 是对偶命题 $(p\wedge\neg p)$。也可以将算子 $\mathbf{x}(x_1,x_2,x_3,x_4)$（我们将其视为向量）写成布尔矩阵形式，$\left[\begin{smallmatrix}x_1x_2\\x_3x_4\end{smallmatrix}\right]$，以获得一种新的视角；如果这样做，我们会发现，只需取 p 或 q 作为最小原始联结词，即可构成一个完备系统（在向量情形下，只有 $\uparrow$ 和 $\downarrow$ 被视为完备的单联结词）。在我们的变换中，还可以从向量情形转向矩阵系统，其中 $N(op_{\boldsymbol{x}})$ 从一个算子中取出数字串，

$$N(\downarrow)=0001\mapsto\begin{bmatrix}00\\01\end{bmatrix}\mapsto\begin{bmatrix}p+1&0\\0&q+1\end{bmatrix}\in g(p,q,1)=(p+1)(q+1).$$

注意，$\det\left[\begin{smallmatrix}00\\01\end{smallmatrix}\right]=0$，这意味着该矩阵是布尔矩阵，这一点适用于所有非平凡情形（$N(\Leftrightarrow)=\left[\begin{smallmatrix}10\\01\end{smallmatrix}\right]$ 是布尔矩阵，但对于这个单位矩阵，行列式为 1）。另请注意，$\left[\begin{smallmatrix}(p+1)&0\\0&q+1\end{smallmatrix}\right]^2=\left[\begin{smallmatrix}(p+1)&0\\0&q+1\end{smallmatrix}\right]$，因此这种表述仍然是布尔的。下面简要概述本文内容：在第2节中，我们首先考察算子，它们是 PBNF 变换的参数，并通过多项式族给出更多示例。定理的证明见第3节，我们还将在那里更深入地探讨

[2]有了 $p\vee q\mapsto(p^S+1)(q^S+1)+1$, $p\wedge q\mapsto pq^S$ 以及用于代换的 S，我们得到 LS:$(pq+1)(pq+1)+1=pq$，RS: $p(q+1.pq+1+1)=p(pq+q+pq+1)+1)=pq$.[3]多项式 $(p+1)p+1$ 看起来不像析取，它是根据德摩根定律得到的 $(\neg p\wedge p)'=p\vee\neg p$，并用于得到 1=T。如果我们取 0=T，则会得到 $p(p+1)=0$，因为在空间 $g(p,q,0)$ 中，我们有 $\wedge\mapsto\vee$.

相关内容。更高级的应用也由此展开，例如 Church [3] 中对对偶定理的一般化。

2. 算子作为 PBNF-变换的参数

现在，我们将更详细地介绍我们对命题演算的表述。本章节可视为比第一章更全面的导论。在下一章中，我们将深入探讨这一主题并证明相关定理。我们首先讨论作为起点的不同类型的算子。这些算子是 PBNF-变换的参数，它会生成多项式，例如 $g_{(p_i,q_j,1)}(p \Leftrightarrow q) = p + q + 1 \in g_{(p_1,q_1,1)}$。我们从选择器 $\mathcal{S}$ 开始，在这里将其依次编号为 $p_i, q_j, i =$1、2、3、4，$j =$ 1、2。请注意，q 可以有两个面（$1010 - q_1, 0101 - q_2$），而 p 有四个面（$1100 - p_1, 0110 - p_2, 0011 - p_3, 1001 - p_4$）。（我们采用这个顺序，尽管（$p_1$）$' = p_3$，而（$p_2$）$' = p_4$。）这里我们有一个公式，其中可以使用 p 和 q，而不能使用.....（回想一下，添加数字串时不能有 0 或 1，1100+0011=1 或 1100+1100=0，但 1100 1010=0110 可行）。我们得到

$$\begin{cases} p_i + p_j = q_{i+j} \\ p_i + q_j = p_{i+j} \end{cases} \tag{2.1}$$

这表明我们不能使用两个相同的，或 $p|q$ 和 $p|q$的拉回。选择器必须给出由元素 1 和 0 组成的两种齐次组合和两种异质组合。我们有 p_1(1，1，0，0) 和 q_1(1，0，1，0)，我们称它们为 1st 和 $2^{nd} Selector$（通常简称为 p 和 q），它们具有不同的真值（此处 $1 =$ 为真，$0 =$为假；请注意，我们也会通过重新命名 $0 =$ 真和 $1 =$假来使用相反的约定），例如在这些映射中

$$\underset{\Leftrightarrow}{\overset{\Rightarrow}{}}\begin{pmatrix} p_1 & 1 & 1 & 0 & 0 \\ q_1 & 1 & 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} \Rightarrow & 1 & 0 & 1 & 1 \\ \Leftrightarrow & 1 & 0 & 0 & 1 \end{pmatrix} \quad \underset{\Leftrightarrow}{\overset{\Rightarrow}{}}\begin{pmatrix} p_3 & 0 & 0 & 1 & 1 \\ q_2 & 0 & 1 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \Leftarrow & 1 & 1 & 0 & 1 \\ \Leftrightarrow & 1 & 0 & 0 & 1 \end{pmatrix}. \tag{2.2}$$

在下一次变换中，我们发现 $\Leftrightarrow$ 变成了 p 和 p'，由此想到这些选择器也是二元算子（稍后我们将对此进行探讨）。

$$\underset{\Leftrightarrow}{\overset{\Rightarrow}{}}\begin{pmatrix} p_4 & 1 & 0 & 0 & 1 \\ q_1 & 1 & 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} \Leftarrow & 1 & 1 & 1 & 0 \\ p & 1 & 1 & 0 & 0 \end{pmatrix} \quad \underset{\Leftrightarrow}{\overset{\Rightarrow}{}}\begin{pmatrix} p_4 & 1 & 0 & 0 & 1 \\ q_2 & 0 & 1 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \uparrow & 0 & 1 & 1 & 1 \\ p_3 & 0 & 0 & 1 & 1 \end{pmatrix} \tag{2.3}$$

这项技术帮助我们为特定命题选择最方便的多项式。我们将真值向量算子写作$op_{\boldsymbol{x}(p_n,q_n)(x_1,x_2,x_3,x_4)}$，在 $\Rightarrow$ 的情形下，我们得到$op_{\boldsymbol{x}1,0,1,1}$，其中 x(1, 0, 1,1)是选择器不同输入的值域，并且我们省略那些通常可从上下文中明确看出的选择器。算子值域中的一个值，或一个输出，我们常称之为该算子的投票值或选取值 。逻辑代数的几何化发生在单位正方形的维恩图中（关于这一点以及格的更多内容，请参阅 [10]）。两个选择器表示两个集合在4个不同区域（1、2、3和4）中的单位分割，我们取 p_1(1100)和 q_2(1010)（我们称 $x_1x_2x_3x_4, x = 0|1$ 为以向量形式给出的算子的数字串），在 $p \cap q$处具有非空交集）。这是区域编号1。编号2是 $p \setminus q$，编号3是 $q \setminus p$，而补

$(p' \cap q')$是编号4。如果我们比较 p(1100)和 q(1010)，就会看到每个数字代表一个区域， p 位于 q的左侧，因此可以清楚哪些区域属于 p、属于 q，或同时属于两者。每一个二元算子代表特定的区域；例如，析取 $\vee$ 的数字串为(1110)，因此它占据除4以外的所有区域。因此，当我们改变输入时，可以切换到不同的算子空间。总共，一个二元运算符空间 $\mathcal{OP}$ 包含16个不同的二元算子，其输出为四位数字串$(1|0,1|0,1|0,1|0)$。在 $p = q$的情形下，单位正方形中只有2个区域： p 和 p'。于是我们令 p 为区域1， p' 为区域2。这种几何用于一元算子，我们共有4$(1|0,1|0)$个，其中有些在大多数文献中既未被命名也未被考虑，除了否定之外——我们在此常称之为拉回作用；然而，它们可以在我们所研究的陈述演算的不同情形中使用。一元算子的协议为([·]:=存在， |:=异或)

$$[f][\mathcal{H}][a_j][\mathrm{T}|\mathrm{F}][p_j|q_k]\big(f:\mathcal{OP}\to(\mathbb{Z}_2|\mathcal{H}), f(op_{\boldsymbol{x}(x_1,x_2)} := f(p_j|q_k,a_0)(op_{\boldsymbol{x}}) \tag{2.4}$$

$$= a_1p|q + a_0 = (1|0, \big|f(a_1p_j|q_k,a_0)),\ a_{n=0,1} = 0|1, \tag{2.5}$$

$$(g(p_j|q_k,a_0))' = g(p'_j|q'_k,a'_0), g(p_j|q_k,\underline{a_0}) \mapsto (1|0 = (\mathrm{T}|\mathrm{F})|(\mathrm{F}|\mathrm{T})), \tag{2.6}$$

$$p_j|q_k \in \mathcal{S} = (1|0)\big). \tag{2.7}$$

这里可以使用 p或 q ，因为多项式族关于 p或 q是线性的：$(0, p|q,$ $p|q+1, 1)$。对于这些线性多项式族，我们使用 f而不是 g。这里的命题演算或语句演算指的是，我们使用$\mathcal{H}(f,g)$，而不是 $op_{\boldsymbol{x}(x_1,x_2,x_3,x_4)}(p_n,q_n)$ $\in \mathcal{OP}$. 每个算子都双射映射到 $\mathcal{H}(f,g)$，即由 p 和q构成的线性和二次多项式之之屋，例如对于双条件联结词[4], ，这里我们展示 DBNF 中间的一步，

$$op_{\boldsymbol{x}(1,0,0,1)}(p_1,q_1) \mapsto (x\wedge y)\vee(x'\wedge y') \mapsto pq + (p+1)(q+1) \tag{2.8}$$

$$= p + q + 1 \in g(p,q,1) \subset \mathcal{H}(f,g). \tag{2.9}$$

但在另一个多项式族中，我们得到

$$op_{\boldsymbol{x}(1,0,0,1)}(p_1,q_1) \mapsto p+q \in g(p',q',0) \subset \mathcal{H}(f,g). \tag{2.10}$$

因为这是第一个情况的补。计算一个命题意味着证明或证伪它；此时它被称为重言式（Taut=1/0），对输入的每种组合都为真，或者被称为矛盾式（Taut=0/1）′。我们想知道是否

[4]并非总是如此： $\vee \mapsto +$。要做到这一点，我们需要不相交集合，$x \cap y = \emptyset$. 否则，$x \vee y \mapsto p + q + pq = (p+1)(q+1) + 1$.

在 $(p\dot{\wedge}(.p \Rightarrow q) \dot{\Rightarrow} q$ 中有一个重言式，因此我们将其映射到 [5]

$$(p\dot{\wedge}.p \Rightarrow q)\dot{\Rightarrow}q \mapsto p^S(q+1)+1 \in g(p,q,1) \tag{2.11}$$

$$= p(p.q+1+1)(q+1)+1p = (p.q+1+p)(q+1)+1 \tag{2.12}$$

$$= pq(q+1)+1 = 1. \tag{2.13}$$

主算子到映射的映射（此处用点标记为（$\Rightarrow$））是一个条件联结词，而 S 表示我们代入 p。那里有一个合取，它是主算子： $p \wedge q \mapsto pq^S$。最后，我们将条件联结词$q^S = (p(q+1)+1)$代入，并借助布尔算术得到（1.6），其中 $x + x = 0$ 且 $x \cdot x = x.$。使用 PBNF，我们还可以像之前看到的那样，在计算得出 1/0 以外的值时，额外获得一些结果，例如联结词消去（我们使用 $\to \mid \Rightarrow$）。

$$(.A \wedge B \to C) \Leftrightarrow (A \to .B \to C).$$

假设我们想消去左侧的合取，以证明该命题。$(A(p) \wedge B(r)) \to C(q)) \mapsto p^S($ $q+1)+1 = pr(q+1) + 1$. 右侧 $A(p) \to (B(r) \to C(q)) \mapsto p(q^S+1)+1 = p($ $r.q+1+1+1) + 1 = pr(q+1) + 1$. 当然，我们可以证明整个命题为真。接下来，我们将 ad LS+RS+1，因为 $p \Leftrightarrow q \mapsto p+q+1 \in g(p,\ q,\ 1)$。因此我们得到

$$((A \wedge B) \to C) \Leftrightarrow (A \to (B \to C)) \mapsto p^S + q^S + 1$$

$$= (pr(q+1)+1) + (pr(q+1)+1) + 1 = 1$$

这意味着我们得到了一个重言式。我们还可以改写

$$pr(q+1)+1 = p(r(q+1))+1 = p(r(q+1)+1)+1)+1$$

以消去 $pr \hookrightarrow p \wedge r$。 [6]

定义 2.1。 联结词或算子属于我们称为 $\mathcal{OP}$ 的算子类别。我们有 4 个与一个命题字母相连的一元算子，其中最常用的是否定（$\neg p$），编号为 $1-4$）。它们属于 $\mathcal{OP}_1(op_{\boldsymbol{x}(x_1,x_2)})$。有 16 个二元算子（编号为 5–20 ）用于连接命题字母（文字、操作数），例如 $p \wedge q$，它是最常见的算子之一（$\vee$，$\wedge$，$\Rightarrow$，$\Leftrightarrow$）$\in \mathcal{OP}$。它们属于 $\mathcal{OP}_2(op_{\boldsymbol{x}(x_1,x_2,x_3,x_4)})$。我们将这些二元联结词分为原始算子（ $\mathcal{OP}_P$ ），它们是入门所需的算子，可作不同的选择

[5]这里 $p \Rightarrow q = \neg p \vee q = (p \wedge \neg q)' \mapsto p(q+1) + 1 \in g(p,\ q,\ 0)$，如果在 $g(p',\ q',\ 0)$ 中有 $(p+1)q$，[6]那么或许使用 $g(p',\ q',\ 0)$ 会更简单，$p \to q \mapsto (p+1)q$，但我们还有 $\wedge \mapsto (p+1)(q+1) + 1 \in g(p',\ q',\ 0)$。我们通常将 g^{-1} 称为纤维，并使用 $\hookrightarrow$.

（例如， $\neg$、$\wedge$或 $\neg$、$\vee$）。 [7]接下来是范式算子 $\mathcal{OP}_{NF}$（$\vee$、$\wedge$、$\neg$）以及我们称为导出算子的那一组$\mathcal{OP}_D$（$\Rightarrow$、$\not\Rightarrow$、$\Leftarrow$、$\not\Leftarrow$、$\uparrow$、$\downarrow$）。这些算子都具有不均匀的异质性，因此可以通过添加数字串得到一个均匀算子。下一组是选择器，它们都是包含两个 1 和两个 0 的算子（同样具有异质性），因此可以充当输入 $\mathcal{OP}_{S_1}$（p、q、p'、q'）和 $\mathcal{OP}_{S_2}$（$\Leftrightarrow$、$|$），也可以只充当 $\mathcal{OP}_S = (p_1, p_2, p_3, p_4; q_1, q_2)$。选择器 $\mathcal{OP}_{S_2}$ 通常只被视为二元算子。对于 $\mathcal{OP}_{S_1}$，情况正好相反；这些算子通常只作为输入，但我们在此说明，它们也可以视为具有特殊性质的二元算子。第四组由平凡算子 [8] $\mathcal{OP}_{T_1}$ 和 $\mathcal{OP}_{T_0}$（1 和 0）组成，它们是仅有的齐性二元算子。现在我们可以建立一个线性序，因为第一个算子可以构造出下一个算子，依此类推。

$$\mathcal{OP}_P - \mathcal{OP}_{NF}(\wedge, \vee\neg) - \mathcal{OP}_S(p, q, p', q', \Leftrightarrow, |) \tag{2.14}$$

$$-\mathcal{OP}_D(\Rightarrow, \not\Rightarrow, \Leftarrow, \not\Leftarrow, \downarrow, \uparrow) - \mathcal{OP}_{T_1} \subset \mathcal{OP}. \tag{2.15}$$

这不是严格序；例如，可以使用 $p \Rightarrow q = \neg p \vee q$ 和 $p \not\Rightarrow q = p \wedge \neg q$. 构造范式算子。请注意，这种序不属于范式序；范式序根据 $op_{\boldsymbol{x}}$ 中 1 或 0 的数量进行排序。

$$\longrightarrow \text{Increasing order of ones}$$

$$op_{\mathbf{0}}(\iota_0) \subset op_{\mathbf{1}}(\wedge, \not\Rightarrow, \not\Leftarrow, \downarrow) \subset op_{\mathbf{2}}(p, q, p', q', \Leftrightarrow, |)$$

$$op_{\mathbf{3}}(\vee \Rightarrow, \Leftarrow, \uparrow) \subset op_{\mathbf{4}}(\iota_1)$$

$$\longleftarrow \text{Increasing order of zeroes}$$

[7]人们通常认为，这些原始联结词应尽可能少，以使命题演算（PC）更加清晰；但也有人指出，随着命题变得更加复杂，这种简化会带来问题。其中一种单原始联结词表述 使用谢弗竖线，通常记作竖线 $|$，但这里我们使用 $\uparrow$。稍后我们将给出相关示例，说明如果允许使用带加法和乘法的矩阵代数，选择器$(p|q)$便可作为单原始联结词。PBNF 中的“原始多项式”可视为 $(p,\ q,\ 1)$：其中有两个选择器，以及用于二元算子的 1；而对于一元算子，则为 $(p,\ 1)$ 或 $(q,\ 1)$。事实上，单原始联结词的概念在多项式表述中并不重要，因为这里只需作出这一种选择。相反，我们将 $(p,\ q,\ 1)$ 视为完整多项式的生成元， $pq + p + q + 1 = (p + 1)(q + 1) \hookrightarrow p \uparrow q$(非析取， $\neg\vee$)。我们还有另一个单原始联结词，即 $pq + 1 \hookrightarrow p \downarrow q$(非合取，$\neg\wedge$)，它展示了多项式作为命题演算的代数几何表述的示例。

[8]在大多数命题演算的表述中，平凡算子仿佛理所当然地存在，但它们并没有那么平凡：如果使用 $1 \mid 0$ 作为真值，就需要它们；即使使用 T/F（或 t/f），这些字母也只是看起来理所当然地存在。在 Church´的表述中 P_1，原始项是 $[\Rightarrow]$（括号和条件联结词），而原始常量是 f。他先定义（$\rightarrow :=$ 表示）， $1 \rightarrow f \Rightarrow f$，然后定义（$\neg p \rightarrow p \Rightarrow 0$，这样就足够了，而且相当巧妙。平凡算子可以置于字母之间，形成一个逻辑算子格，并根据输入的真值对输入进行求值。 T_0 不相信任何输入，而 T_1 相信每一个输入。这组中的其他算子有 $\wedge$ 和 $\vee$，还有选择器， p_1， q_1。更多内容将在后续论文中介绍。

算子空间 $\mathcal{OP}$	$Number$	$Names$
(1) 一元算子	$1-4$	$=, -, \neg, +$
(2) 二元算子	$5-14$	$\wedge, \vee, \Rightarrow \Leftarrow, \not\Rightarrow, \not\Leftarrow, \Leftrightarrow, \nmid, \downarrow, \uparrow$
(3) 平凡算子	$15-16$	$Zero(\iota_1)$ $One(\iota_0)$
(4) 双常量算子	$17-20$	p, q, p' q'
(5) 常量（空类算子）	$21-24$	$\in$, $=$, $(\cdot)$: $=$对所有, $[\cdot]$: $=$存在
(5′) 操作数（空类算子）	$25-$	$p, q, r, s, p_1, q_1, r_1, s_1 \ldots$

稍后介绍不同情形下的格时，我们会对此加以说明；如前所述，可以将 $0 \mapsto 1$ 和 $1 \mapsto 0$ 互换，以分别表示真值和假值。算子用于构造公式或陈述形式，并由此形成算子空间。它们可以与算子空间中的对偶相连 $\mathcal{OP}_x$，也可以直接与 $\mathcal{H}(f, g)$ 相连；后者是与算子空间 $\mathcal{OP}_x$相关联的多项式族。量词 $\forall x$或这里的（x）以及 $\exists$ 或这里的 $[x]$ 通常不称为算子，而是称为 $\mathcal{OP}$ 中的常量，与原始项“$=$”和 $\in$并列。这里我们也将它们称为空类算子。算子置于命题之间以构成其他命题，而“$\in$”和“$=$”则置于常量之间以构成命题。例如，我们用 $\mathcal{OP}, p, q, r, s, p_1, q_1, r_1, s_1 \ldots$中的常量构造命题 $p = q$ 或 $p \in q$。我们还将 p, p' 和 q, q' 放入 $\mathcal{H}(f, g)$ 中；它们兼具常量、命题字母（必要时也可以使用 p 和 q）和算子的双重性质（选择器 p, q, p', q'）。有了这组算子—— $\mathcal{OP}$中的第五组——总计可得 24 个算子：4 个一元算子、10 个普通二元算子、2 个平凡算子、4 个双常量（既是选择器/算子，也是命题字母），以及 4 个常量。我们也将命题字母，即操作数，称为空类算子，归为第 5’ 组；它们从 25 开始编号，依次递增。

注释 2.2。 可以注意到，对于条件联结词 $p \Rightarrow q$，该算子属于 Tarski [9], 提出的一类共轭句，还包括逆命题 $p \Leftarrow q$，逆命题 $\neg p \Rightarrow \neg q$ 和逆否命题 $\neg p \Leftarrow \neg q$。稍后证明 $\mathcal{H}^*$ 时将展示这一点，届时这些共轭句会被转换为 p 和 q 中的多项式。我们现在已经可以看出结果，最右侧的多项式展示了它们之间的联系。

共轭句	$Name$	$Polynom\ g(p,\ q,\ 1)$	$Polynom\ g(p',\ q',\ 0)$
(1)$p \Rightarrow q$	$Conditional$	$p(q+1)+1$	$(p+1)q$
(2)$\neg p \Rightarrow \neg q$	$Inverse$	$(p+1)q+1$	$p(q+1)$
(3)$p \Leftarrow q$	$Converse$	$(p+1)q+1$	$p(q+1)$
(4)$\neg p \Leftarrow \neg q$	$Contrapositive$	$p(q+1)+1$	$(p+1)q$。

我们观察到 $1 = 4$ 和 $2 = 3$（将在下一章节中证明），这使我们有机会选择用来证明该等价关系

$$p \Leftrightarrow q = (p \Rightarrow q \wedge p \Leftarrow q), \text{Conditional-Converse},$$

以下内容

$$p \Leftrightarrow q = (p \Rightarrow q \wedge \neg p \Rightarrow \neg q), \text{Conditional-Inverse},$$

$$p \Leftrightarrow q = (\neg p \Leftarrow \neg q \wedge p \Leftarrow q), \text{Contrapositive-Converse},$$
$$p \Leftrightarrow q = (\neg p \Leftarrow \neg q \wedge \neg p \Rightarrow \neg q), \text{Contrapositive-Inverse}$$

.

3. PBNF 之屋中的多项式族

到目前为止的内容将给出一些证明和定义技术的示例，展示语句演算如何运作。稍后，我们将深入探讨这些内容，并证明所用的定理。在这里，我们研究并使用具有线性多项式 $f(p,\ a)$ 的四个最自然的族。

(3.1) $1.f(p,1), \text{Little normal family : normal input and reading.}$

(3.2) $2.f(p,1) \mapsto f(p,1'), \text{Little complement family : reading 1(T) to 0(T).}$

(3.3) $3.f(p,1) \mapsto f(p',1), \text{Little pullback family : input } p \text{ to } p' = p+1.$

(3.4) $4.f(p',1) \mapsto f(p',0), \text{Little pullback complement family :}$

(3.5) 更改输入 p 并读取。

我们将回过头来解释这一族中的规则和条件；这些规则和条件与二次多项式 $\in \mathcal{H}(g)$ 并不重合（请记住，这里采用了不同的几何学，即仅有 $x = p$ 和 $\neg x = p+1$）。我们称之为

(3.6) $1.\mathcal{H}^* g(p,q,1),\ \text{Normal Family},$

(3.7) $2.\mathcal{H}' : g(p,q,1' = 0),\ \text{Complement Family},$

(3.8) $3.\mathcal{H}'' : g(p',q',1),\ \text{Pullback Family},$

(3.9) $4.\mathcal{H}^{**} : g(p',q',1'),\ \text{Pullback Complement Family}$

注3.1。注意，$'$ 表示向二元多项式形式添加 1=1111，对命题取否定或进行拉回。根据真值顺序，我们将多项式族分为两组：(1, 0) 的标准族，或 (0, 1) 的补族。当然也存在混合族，例如 $g(p,q',0)$ 或 $g(p',q,1)$，有时使用它们会很方便。我们不打算研究所有多项式族，因为只看前四个就能看出规律；其他的只是调整$op_{\boldsymbol{x}(x_1,x_2,x_3,x_4)}$的位置，或改变真与假的取值。要实际计算族的数量，我们从第一个选择器 p 开始，它有四种排列：(1, 1, 0, 0)、(0, 1, 1, 0)、(0, 0, 1, 1) 和 (1, 0, 0, 1)；第二个选择器 q 有两种：(1, 0, 1, 0)和(0, 1, 0, 1)。如果允许将六种可能性（$p, q, p', q', p\downharpoonright, p^{\Leftrightarrow}$）都作为选择器，就会得到 $6 \cdot 4 = 24$ 种对偶空间的可能性（请注意，补(x+x′ = 1)和算子(x+x = 0)，或两个相同的项，不能一起使用）。这些族可以从格的底部（标准空间）或顶部（补空间）读取；这意味着，如果用 (1, 0) 或 (0, 1) 表示真和假，就会得到 48 个对偶空间；或者说，相对于基空间有 47 个对偶空间，它们以 1st 和 $2^{nd}Selector$作为输入。

定义 3.2。 多项式族由四个粒子或简单形式构成，它们由变量 p 和 q 组成，对应于 DBNF($x(p)$ $y(q)$)$pq = x \wedge y$， $p(q+1) = x \wedge y'$， $(p+1)q = x' \wedge y$和 $(p+1)(q+1) = x' \wedge y'q$。这些 DBNF 通常称为短语，而在 CBNF 中称为子句；这种变换是通过德摩根否定实现的，例如 $\neg(x' \vee y') = x \wedge y$。在多项式族中，我们改为加 1，即 $(p+1.q+1+1)' = (p+1+1)(q+1+1) + 1 + 1 = pq$ （模2）。

以下是上文所述最常见的多项式族表格。下一章节中，我们将证明这一模式。

OP	$\mathcal{H}^*$: $g(p,\ q,\ 1)$	$\mathcal{H}'$: $(g\ (p, q, 1')\ )$	$\mathcal{H}''(p',\ q',\ 1))$	$\mathcal{H}^{**}$: $(g(p',\ q',\ 1')$
p	$p(1,\ 1,\ 0,\ 0)$	p'	p'	p
p'	$p'(0, 0, 1, 1)$	p	p	p'
q	$q(1,\ 0,\ 1,\ 0)$	q'	q'	q
q'	$q'(0, 1, 0, 1)$	q	q	q'
$\neg p$	$p+1$	p	p	$p+1$
$p \vee q$	$(p+1)(q+1)\ +1$	$(p+1)(q+1)$	$pq+1$	pq
$p \wedge q$	pq	$pq+1$	$(p+1)(q+1)$	$(p+1)(q+1)\ +1$
$p \Rightarrow q$	$p(q+1)\ +1$	$p(q+1)$	$(p+1)q+1$	$(p+1)q$
$p \not\Rightarrow q$	$p(q+1)$	$p(q+1)\ +1$	$(p+1)q$	$(p+1)q+1$
$\neg p \Rightarrow \neg q$	$(p+1)q+1$	$(p+1)q$	$p(q+1)\ +1$	$p(q+1)$
$p \Leftarrow q$	$(p+1)q+1$	$(p+1)q$	$p(q+1)\ +1$	$p(q+1)$
$p \not\Leftarrow q$	$(p+1)q$	$(p+1)q+1$	$p(q+1)$	$p(q+1)+1$
$\neg p \Leftarrow \neg q$	$p(q+1)\ +1$	$p(q+1)$	$(p+1)q+1$	$(p+1)q$
$p \downarrow q$	$(p+1)(q+1)$	$(p+1)(q+1)\ +1$	pq	$pq+1$
$p \uparrow q$	$pq+1$	pq	$(p+1)(q+1)\ +1$	$(p+1)(q+1)$
$p \Leftrightarrow q$	$(p+1)\ +(q+1)\ +1$	$(p+1)+(q+1)$	$(p+1)+(q+1)\ +1$	$(p+1)+(q+1)$
$p\|q$	$(p+1)\ +\ (q+1)$	$(p+1)+(q+1)\ +1$	$(p+1)+(q+1)$	$(p+1)+(q+1)\ +1$
ι_1	$(p+1)pq+1$	$(p+1)pq$	$p(p+1)(q+1)$	$p(p+1)(q+1)\ +1$
ι_0	$(p+1)pq$	$(p+1)pq+1$	$p(p+1)(q+1)\ +1$	$p(p+1)(q+1)$

不过，先给出一些定义和例子，以便澄清相关内容。

定义3.3。 命题是指可以判定为真（T 或 1/0）或假（F 或 0/1）的陈述句，这称为该句子的真值。对于每个命题，我们可以将它们命名为 p， q， r， s，p_1， q_1， r_1， s_1，.....我们将单一命题称为文字、文字或空类 算子。文字可以与联结词或算子结合，形成复合命题。

定义3.4. 我们归入陈述形式的命题， $S_{c(k,l)}$,依据其所含不同字母的数量以及联结词的数量而分属不同的类。在类 $S_{c(k\geq 0,\ l\geq 1)}$, 中， k 用于计数联结词的数量， l 用于计数不同文字的数量。借助 c，我们表明语句演算是经典的，即我们有两个真值：1和0。在某些非经典系统中，我们可以有多个真值，并且命题依赖于距离(x)或时间(t)，此时我们使用 $S_{n(k,l)}(x, t)$，而 $n \in \mathbb{N}$ 则为真值的数量。我们还使用其他大写字母，例如 $T_{c(k,l)}$ 或 $R_{c(k,l)}$ ，以表示不同的形式。

定义3.5。 若每一种使 S 为真的（1，0）赋值也使T 为真，则命题形式 $S_{c(k,l)}$ 逻辑蕴含或蕴涵另一个命题形式 $T_{c(k,l)}$ 。我们用 $S_{c(k,l)} \vdash T_{c(k,l)}$ 表示这一事实。若 $S_{c(k,l)}$ 在每组真值输入下

都只取值1/0（真），则 $S_{c(k,l)}$ 是矛盾式；若它在每组真值输入下都只取值1/0（假），则它是矛盾式。

定理 3.6。

$$(S_{c(k,l)} \vdash T_{c(k,l)}) \Leftrightarrow (S_{c(k,l)} \Rightarrow T_{c(k,l)}) \text{ is a tautologi.}$$

证明。 我们可以使用引言中的以下内容

$$(3.10) \qquad (S_{c(k,l)} \vdash T_{c(k,l)}) \begin{pmatrix} p & 1 & 1 & 0 & 0 \\ q & 1 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} S_{c(k,l)} & 1 & 1 & 0 & 0 \\ T_{c(k,l)} & 1 & 1 & 1 & 0 \end{pmatrix} \Leftrightarrow \begin{pmatrix} S & 11 & 10 & 01 & 00 \\ \Downarrow T & 1 & 1 & 1 & 1 \end{pmatrix}$$

这可以表述如下： $p \vdash q$ 当且仅当 p 为真时，q也必为真。这意味着 $\neg(p = 1 \wedge q = 0)$，因此 $p \Rightarrow q$ 永远不为假，所以 $p \Rightarrow q$ 是一个重言式。□

例 3.7。 我们要证明 $(p \vdash p) \Leftrightarrow (p \Rightarrow p)$。我们有 $\neg(p = 1 \wedge q = 0) = \neg(1 \wedge 0) = 0 \vee 1 \mapsto (0 + 1)(1 + 1) + 1 = 1$ 用于左侧。

$$1 \Leftrightarrow (p \Rightarrow p) \mapsto g(p, q, 1) : p + q^S + 1 = 1 + p(p + 1) + 1 + 1 = 1.$$

例 3.8。 而这里

$$(p \vdash .p \vee q) \Leftrightarrow (p \Rightarrow .p \vee q).$$

对于 $(p \vdash .p \vee q)$，我们使用 $g(p', q', 0)$，而 $\neg(p = 0 \wedge .p \vee q = 1)$ 给出 $((p+1)(pq+1) + 1)(0,1) = ((0 + 1)(01 + 1) + 1) = 0$. 用于等价关系

$$g(p', q', 0) : p^S + q^S = 0 + (p + 1)pq = 0.$$

这证明了该命题。我们注意到，即使没有任何固定的多项式 $g(\vdash)$，我们仍然能够构造出一个。

例 3.9。 接下来，我们对 $(p \wedge q) \vdash q \Leftrightarrow (.p \wedge q \Rightarrow q)$ 使用一种特殊形式 $g(p, q', 1)$。左侧：$(p \wedge q) \vdash q$表示 $\neg(p \wedge q) = 1 \wedge q = 0)$ 不成立。我们得到 $(p+1)^S q^S = ((p+1)q+1)q$以及 $p+q = p+(p+1)q(q+1)+1 = p+(p+1) = 1$.这里，我们专门针对这个问题使用一个 a g-族，说明多项式形式非常灵活。

例 3.10。 下一个例子有许多条件联结词，我们在这里尝试 $g(p, q', 0)$

$$(p \Rightarrow q \Rightarrow p) \vdash p \Leftrightarrow (p \Rightarrow q \Rightarrow p) \Rightarrow p.$$

这被映射为 $g(p, q', 0)$： $q^S p = (p + 1)(q + 1)pqq = 0$；当我们尝试求出形如 $p(p + 1)$ 或$q(q + 1)$ 的矛盾式时，这是一个简单的计算，它们的值为零。

引理 3.11。 现在我们使用 $g(p, q, 1)$来证明分配律 $\in S_{c(5,3)}$

$$p \wedge (q \vee r) = (p \wedge q) \vee (p \wedge r)$$

证明。 我们从映射为 pq^S 的左侧开始

$p \wedge (q \vee r) \mapsto p.(q+1)(r+1)+1 = (pq+p)(r+1)+p = pqr+pq+pr = p(qr+q+r)$

右侧被映射到 $(p^S + 1)(q^S + 1) + 1$

$$(pq + 1)(pr + 1) + 1 = pqr + pq + pr = p(qr + q + r)$$

□

例3.12。 众所周知，条件联结词并不能得出逆命题；通常这是通过给出一个反例来证明的，这足以推翻一个定理，但也许并不令人满意。这里的 $\cdot$ 表示主算子

$$(p \Rightarrow q) \dot{\Rightarrow} (q \Rightarrow p) \tag{3.11}$$

我们将其映射为

$$g(p, q, 1) : p^S(q^S + 1) + 1 = p^S q^S + 1 + p^S.$$

第一次乘法和补 1 得到

$$p^S q^S + 1 = (p.q + 1 + 1)(.p + 1\ q + 1) + 1 = p + q$$

这就是异或。我们将其相加并拉回

$$p^S : p + q + p(q + 1) + 1 = (p + 1)q + 1 \hookrightarrow q \Rightarrow\ p$$

并得到

$$(p \Rightarrow q) \dot{\Rightarrow} (q \Rightarrow p) \vdash (q \Rightarrow\ p). \tag{3.12}$$

我们看到，事实上，逆命题对于条件联结词是幂等的，反之亦然。我们用模式来展示这一点，并回顾一下，逆命题和条件联结词在这些输入上的投票值相同

$$(\Leftarrow)^{\Rightarrow} = \Leftarrow\ \left(\begin{smallmatrix} \Rightarrow & 1 & 0 & 1 & 1 \\ \Leftarrow & 1 & 1 & 0 & 1 \end{smallmatrix}\right) \mapsto \left(\begin{smallmatrix} \Leftarrow & 1 & 1 & 0 & 1 \end{smallmatrix}\right)$$

$$(\Rightarrow)^{\Leftarrow} = \Rightarrow\ \left(\begin{smallmatrix} \Leftarrow & 1 & 1 & 0 & 1 \\ \Rightarrow & 1 & 0 & 1 & 1 \end{smallmatrix}\right) \mapsto \left(\begin{smallmatrix} \Rightarrow & 1 & 0 & 1 & 1 \end{smallmatrix}\right)$$

定理3.13。 我们可以说，如果两个命题产生的多项式能够化简为相同的标准形，那么这两个命题等价。此外，*PBNF* 是一种典范形。

证明。 我们注意到，DBNF 的多项式代数表达式如下所示，而右侧则是 PBNF。

$$x \wedge y\ \mapsto pq$$

$$x \wedge \neg y \mapsto p(q + 1)$$

$$\neg x \wedge y \mapsto (p + 1)q$$

$$\neg x \wedge \neg y \mapsto (p + 1)(q + 1).$$

每个形式 $S_{c(k,l)}$ 都通过一个双射 g 映射到这些粒子（从变换列表中可以清楚看出），因此 g^{-1} 存在，并且我们有$g^{-1}((g(S_{c(k,l)})) = S_{c(k,l)}$. 我们不可能有 $g(S_{c(k,l)}) = g(S^*_{c(k,l)})$，否则 DBNF 会将两个形式映射到同一个多项式上，并且 $g^{-1}(g(S_{c(k,l)})) = g^{-1}((g(S^*_{c(k,l)}))$ 给出 $S_{c(k,l)} = S^*_{c(k,l)}$，这就构成了矛盾式。显然，多项式代数中的每次约简都必须以极小多项式结束，因为 $\mathbb{Z}_2$ 中的加法和乘法规则意味着，例如，我们不可能有 $g(p \vee \neg p) \neq g(1)$。因此，PBNF 是一种典范形。□

命题 3.14。 我们声称 $p \wedge q \Rightarrow r \Leftrightarrow p \Rightarrow (q \Rightarrow r)$，但它们的逆命题并不等价 $r \Rightarrow p \wedge q \not\Leftrightarrow (q \Rightarrow r) \Rightarrow p$

证明。

$$p \wedge q \Rightarrow r \mapsto_{g(p,q,1)} (p^S(q^S+1)+1 = [p = pq; q = r] = pq(r+1)+1$$

$$p \Rightarrow (q \Rightarrow r) \mapsto p(q^S+1)+1 = [q = (q(r+1)+1] = p(q(r+1)+1+1)+1$$

$$= pq(r+1)+1.$$

我们现在使用

$$p \Rightarrow q \mapsto p^S + q^S + 1 = pq(r+1)+1+pq(r+1)+1+1 = 1.$$

对于逆命题，我们得到

$$r \Rightarrow p \wedge q \mapsto_{g(p,q,1)} p^S(q^S+1)+1 = r(pq+1)+1$$

以及右侧

$$(q \Rightarrow r) \Rightarrow p \mapsto (p+1)q^S+1 = (p+1)(q(r+1)+1)+1 = (p+1)q(r+1)+(p+1)+1$$

$$= (pq+q)(r+1)+p = pqr+pq+qr+q+p = pq(r+1)+p+q(r+1)$$

$$= (r+1)(pq+q)+p = (r+1)q(p+1)+p.$$

□

注记 3.15。 这里，我们使用了三个命题（p、 q、 r），并且可以将上面的组合命题写成 PBNF，但我们无法确定将结果拉回算子空间的拉回（$\hookrightarrow$），因此失去了双射。仅使用p和 q，就有 16 个多项式都表示一个二元算子。如果再添加一个，就会得到 256 个多项式，每个都表示一个“算子”。可以看出，此时我们从字母（pqr）中得到 8 个粒子（原子）。

$$x \wedge y \wedge z \mapsto pqr \to 1.$$

$$x \wedge y \wedge \neg z \mapsto pq(r+1) \to 2.$$

$$x \wedge \neg y \wedge z \mapsto p(q+1)r \to 3.$$

$$\neg x \wedge y \wedge z \mapsto (p+1)qr \to 4.$$

$$x \wedge \neg y \wedge \neg z \mapsto p(q+1)(r+1) \to 5$$

$$\neg x \wedge y \wedge \neg z \mapsto (p+1)q(r+1) \to 6$$

$$\neg\, x \wedge \neg y \wedge z \mapsto (p+1)(q+1)r \to 7$$

$$\neg x \wedge \neg y \wedge \neg z \mapsto (p+1)(q+1)(r+1) \to 8$$

尽管如此，PBNF-变换仍将使用与真值表相同的方法来确定复合命题的真值。

例 3.16. 我们可以找到一些三元算子，因此对于一种“弱”双射，例如在 Church 12 中再次出现的 $[p,q,r]\in\mathcal{OP}_*$，那里称为条件析取，它与 $[r,q,p]\in\mathcal{OP}_{**}$ 对偶。我们知道，二元算子的一个生成元是 $(p+1)(q+1)=pq+p+q+1$。要得到完整的三元多项式，我们乘以 $(r+1)$：$pqr+pq+pr+qr+p+q+r+1=(p+1)(q+1)(r+1)$。 [3] 中的数字串表示 $[p,q,r]=11100010$，这意味着我们可以像上面那样，将单位正方形中的区域从 1 到 8 编号，从而构造该多项式。由此得到

(3.13)
$$[p,q,r]\mapsto pqr+pq(r+1)+p(q+1)r+(p+1)(q+1)r=pq+qr+r;$$

(3.14)
$$N([p,q,r])=11100010$$

这个三元运算符的特点是它具有自对偶性，并且只需两个常量（$[p,q,r],\ t,\ f$）便是完备的。我们稍后会对此进行说明。

例 3.17. 拉回的其他三元联结词包括少数运算符和多数运算符，它们互为补，通常写作（$Mpqr$），当至少两个输入为 1 时，结果为 1。它具有

$$\begin{aligned}(p\wedge q)\vee(p\wedge r)\dot{\vee}(q\wedge r)&\mapsto(p^S+1)(q^S+1)+1\\&=((pq+1)(pr+1)+1)+1)(qr+1)+1\\&=(pqr+pq+pr+1)(qr+1)+1=pq+pr+qr.\end{aligned}$$

我们得到等式

$$pq+pr+qr=1\Leftrightarrow\ 111,110,101,011.$$

因此这里可以使用 PBNF变换，而少数三元运算符只需将读取位置移到 $T=0$。

例 3.18. 我们还可以在 $+pqr$ 的三元奇偶联结词中找到另一种拉回；当值为 1 的输入数量为奇数时，它将返回 1。对于一个四位字符串，输入为 10 和 01，此处的算子是异或或 $|$，但在其他地方通常写作 $+$。适用的输入为 111、100、010、001，它们对应的区域为 1、5、6、7，因此我们得到

$$pqr+p(q+1)(r+1)+(p+1)q(r+1)+(p+1)(q+1)r=p+q+r.$$

因此我们注意到，这里的映射类型与二元 $|$（$p+q$）相同。由于布尔性质 $x+x=0$，选取值 $p+q+r$ 会是一个不错的猜测。

我们继续举更多形如 $S_{c(k,3)\cdot}$ 的复合命题示例。

例 3.19。我们证明

$$(p\wedge q)\wedge r=p\wedge q\wedge(q\wedge r).$$

这映射为$(pq)r=p(qr)$，我们可以使用实数的结合律。

例 3.20。 我们还证明了条件联结词下的传递性

$$(p \Rightarrow q) \wedge (q \Rightarrow r) \dot{\Rightarrow} (p \Rightarrow r)$$

这映射为 $p^S(q^S + 1) + 1$ 和 $p^S = (p(q + 1) + 1)(q(r + 1) + 1) = p(q + 1) + q(r + 1) + 1$

$$q^S = [p(r + 1) + 1)]$$
$$p^S(q^S + 1) + 1 = p^S([p(r + 1) + 1)] + 1) + 1$$
$$= p^S p(r + 1) + 1$$
$$[p(q + 1) + q(r + 1) + 1]p(r + 1) + 1$$
$$= p(q + 1)(r + 1) + pq(r + 1) + p(r + 1) + 1$$
$$= p(q + 1)(r + 1) + pq(r + 1) + p(r + 1) + 1$$
$$= pqr + pq + pr + p + pqr + pq + pr + p + 1 = 1$$

4. 采用一次与二次多项式的扩展陈述演算

这里我们考虑一个二值函数：单一命题或复合命题变量的值属于 $(y_1, y_2, \ldots\ldots, y_n)$，而算子的值属于 $(x_1, x_2, \ldots\ldots, x_n)$。

定义 4.1。 $f: \mathcal{D}^n \longrightarrow (\mathbb{Z}_n, \mathbb{Z}_n)$ 是定义域 $\mathcal{D}$ 上以 n 个自变量（算子）为参数的函数；空类算子（零算子）可以取值 $y_0, y_1, \ldots\ldots, y_n \in \mathbb{Z}n$，而对于 (1,2)-算子，则取一个数字串，即形如 $(x_1, x_2, \ldots\ldots, x_n)$ 的向量，通常简写为 $(x_1 x_2 \ldots x_n)$，甚至写作 $x_1 x_2 \ldots x_n$，其中 $x_n \in \mathbb{Z}n$ ；我们称之为值域。

我们在这里给出了 $n \in \mathbb{N}$ 的定义，但通常只考虑两个真值 $y_0 = 0$ 和 $1 = y_1$ ，使得 $y_0, y_1 \in \mathbb{Z}_2$ ，并且 $x \in \mathbb{Z}_2$ 和 $|x_n| = 1$，2，3，4。然而，也存在丰富得多的推广， [2] 甚至可以推广到连续统。

例 4.2。如果我们考虑析取，就得到

$$f(S_{c(1,0)})(1, 0) = f((\vee)(1, 0)) = (x_1, x_2, x_3, x_4)(1, 0) = 1110$$

因此，参数 $\vee$ 被映射为一个由四个 x 组成的字符串，然后在此处对其求值（1，0）；但如果只显示数字串，我们像以前一样使用 $N(\vee) = (x_1, x_2, x_3, x_4)$（0，1）$= (0001)$。我们也对命题求值，例如

$$f(p \wedge (p \Rightarrow q)) \mapsto 1/0.$$

这里的参数是（$p, q, \wedge, \Rightarrow$）$\in D.$。通常，结果由逻辑规则、公理或真值表决定，但在这里，我们转而将其变换到多项式空间 $g(p, q, a)$

$$g(f(S_{c(k,l)})) \mapsto (g(p, q, a) \mapsto 1, 0 \quad \text{or} \quad g(p, q, 1)$$

在这种情况下，我们有

$$g(f(p \wedge (p \Rightarrow q))) \mapsto pq^S = p(p(q + 1) + 1) = pq \mapsto p \wedge q.$$

因此我们得到 $f(p \wedge (p \Rightarrow q)) = p \wedge q$ 也可以只进行计数

$$f(p \wedge (p \Rightarrow q)) = (1100)(1010)^S = (1100)(1011) = 1000$$

这里在对条件命题(1011)进行代换后使用逐分量乘法，这与使用多项式相同。之后我们会看到更多这样的应用。我们也将写作 $g(S_{c(k,l)})$，而不写作 $g(f(S_{c(k,l)}))$。

接下来，我们定义语句演算中将要使用的概念，即通过单位划分来呈现的几何视角。

定义 4.3。当 j 为变量数时，单位被划分为 $k = 2^j$ 个区域。只有一个变量($j = 1$)时，设为 x，单位被划分为两个区域（$k = 2$）： x 和（非 x） $\neg x$。幂集 $\mathcal{P}(X_1)$包含 2^k个成员，因此 $\mathcal{P} = 0,\ x,\ \neg x,$ 1 和 $|\mathcal{P}| = 4.$。我们有 $x, \neg x$，它们将单位划分为 $1 = (x \wedge \neg x)$。有两个变量 $j = 1,$ 2（x 和 y）时，我们得到 $2^2 = 4$个将单位划分的区域： $x \wedge y,\ x \wedge \neg y,\ \neg x \wedge y$和 $\neg x \wedge \neg y.$。幂集的大小为 $|(P)| = 16.$

定义 4.4. 算子空间 $\mathcal{OP}$ 中有 2^{2^j} 个类型为 $op_{\boldsymbol{x}}$ 的成员，它们各自表示一个与命题字母相连或位于陈述字母之间的算子。这里我们考虑一元算子（$j = 1$），共4个，以及二元算子（$j = 2$），共计16个。

现在我们希望通过PBNF-变换，在一个 $\mathcal{OP}$-空间与一个多项式族之间建立联系。[9]

定义 4.5. 多项式之家 $\mathcal{H}$，包含多项式形式，即符号，其中每个符号都通过其值域（$x_1,\ x_2,\ .....,\ x_n$）与算子空间 $\mathcal{OP}$ 中的一个成员相连。

定义 4.6. 多项式族都将是布尔代数的类型，该布尔代数为

$$\mathcal{B} =< B, \wedge, \vee, ', 0, 1 >; \wedge, \vee, ' \in op_{\mathcal{B}} \tag{4.1}$$

其中 $\wedge$ 与 $\vee$ 是 $\mathcal{B}$中的二元算子， $'$ 是一元算子，0，1是被指定的不同元素，因此 $B \in S_c$ 是这些算子在其上作用的非空集。这些二元算子可交换，分配律成立，但结合律不成立，因此例如一般情形下（$x \wedge y$） $\vee\, x \neq x \wedge$（$y \vee x$），除非 $x = y.$

定义 4.7。我们在此考虑的符号与 $j = 1,$ 2 是 p 和 q 的二次多项式，它们通过模 2 下的加法和乘法构成一个布尔环，其中这些多项式是形如 $a_1pq + a_2p + a_3q + a_4$ 和 $a_n = 0,$ 1 的有限和。 p 和 q 在多项式中都具有

[9]这种思维方式在数学中得到了广泛应用。例如，在几何学中，圆心位于原点、半径为 1 的圆可映射为代数方程 $x^2 + y^2 = 1$。微分算子也可以映射为代数形式，此时 D_xD_y 变成简单的多项式 $\xi\eta$。其思路是将问题转移到一个更容易处理的空间；在这个例子中，就是布尔多项式代数。

几何意义，即作为划分单位的面积，定义并集和非空交集；同时也表示不同于 p 和 q 的面积，这些面积还具有数论意义，表示一个数字串 $p = 1100$ 和 $q = 1010.$ 。我们还考虑只含有 p 或 q 中线性多项式的更简单系统，用于一元算子，构成一个仅定义了加法的布尔群。这些多项式同样是形如 $a_1pq + a_2p + a_3q + a_4$ 和 $a_n = 0$，1 的有限和，但这里 $a_1 = a_{2|3} = 0.$

在下一个定义中，我们通过在加法和乘法之外加入矩阵运算，来考虑布尔环的一种扩展。稍后我们将用这一概念来获得一种略有不同的视角。这并非指元素取自布尔环的矩阵，而是指布尔环中的矩阵，以及该环中的元素。

定义 4.8. 通过加入矩阵乘法和加法，可以扩展布尔环。接着我们考虑 $\begin{bmatrix} x_1x_2 \\ x_3x_4 \end{bmatrix}$，其中使用加法和乘法，并对 2 取模。我们还使用转置，它可以看作是改变数字串，例如 $p^T(1100) = q(1010)$ 或 $\begin{bmatrix} x_1x_3 \\ x_2x_4 \end{bmatrix}$。

命题 4.9. 若要使一个 2×2-矩阵为布尔矩阵，且其元素为布尔元素，则必须满足

$$\begin{bmatrix} x_1x_2 \\ x_3x_4 \end{bmatrix} + \begin{bmatrix} x_1x_2 \\ x_3x_4 \end{bmatrix} = 0$$

and

$$\begin{bmatrix} x_1x_2 \\ x_3x_4 \end{bmatrix} \cdot \begin{bmatrix} x_1x_2 \\ x_3x_4 \end{bmatrix} = \begin{bmatrix} x_1x_2 \\ x_3x_4 \end{bmatrix}$$

我们发现，以下矩阵算子 $p, p', q, q', I, \wedge, \uparrow, \iota_0$ 均为布尔的，并且在左乘和右乘下都对加法与乘法封闭。

证明。 经检验，我们发现选择器 $p, p', q, q', I, \wedge, \uparrow, \iota_0$ 作为矩阵满足这些标准，可以加入布尔环。异或()未通过检验，结果为 $()^2 = I$，因为它是 I 的镜像映射。第一个平凡算子 $\left[\begin{smallmatrix} 0 & 0 \\ 0 & 0 \end{smallmatrix}\right]$ 是布尔的，但第二个$\left[\begin{smallmatrix} 1 & 1 \\ 1 & 1 \end{smallmatrix}\right]$不是。在范式算子中， $\wedge$ 作为矩阵是布尔的，但 $\vee$不是。在导出算子中， $\uparrow$是布尔的， $\downarrow$ 不是，其余算子也都不是。所有矩阵算子元素在转置、加法以及左乘和右乘下也都封闭。这并非限制，因为我们通常只需要单侧乘法。□

这里我们已经展示了一个这样的例子，其中使用矩阵作为生成元。

定理 4.10. 矩阵定义使得有可能以选择器作为命题演算表述的单联结词本原来开始。

证明。 我们从四个矩阵运算开始，此处我们使用 p 作为选择器。

$$\begin{cases} \boldsymbol{x}_p \mapsto \left(\begin{smallmatrix} 1 & 1 \\ 0 & 0 \end{smallmatrix}\right) \\ \boldsymbol{x}_{q=p^T} \mapsto \left(\begin{smallmatrix} 1 & 0 \\ 1 & 0 \end{smallmatrix}\right) \\ \boldsymbol{x}_{p'} \mapsto \left(\begin{smallmatrix} 0 & 0 \\ 1 & 1 \end{smallmatrix}\right) \\ \boldsymbol{x}_{p\times q} \mapsto \left(\begin{smallmatrix} 0 & 0 \\ 0 & 0 \end{smallmatrix}\right) \end{cases} \tag{4.2}$$

现在我们有通常的选择器 p, q 和 p'。我们可以相加，以得到1（它并非布尔矩阵） $p + p' = p + p + 1 = 1$ ，它们之所以被定义，是因为 p、 p' 和0都作为布尔矩阵存在，因此它们也可以映射到多项式。回想一下，我们并不要求用我们的原始项来构造平凡算子，只要这些以后能够被定义即可，见 Church， [3]中第129页§ 24。在此之后从 p开始，通过将我们已定义的分量相加与相乘，我们便得到完整的多项式，从而得到$pq + p + q + 1 \in \mathcal{H}$(g($p, q,$ 1)), 并且可以用与之前相同的方法构造任意算子。有 $p, q,$0和1就足够了，这表明了选择器与平凡算子的重要性。□

5. 线性与第一个多项式范式定理的证明

布尔多项式在以下迭代意义下是线性的 p， $q \in S_{c(k,l)}$ （命题形式、 c=经典、 k=算子（联结词）计数、 l=文字总数）。

定义 5.1. 我们用 $\star$表示某个空间或某个算子

$$g(S_{c(k,l)} \star S_{c(m,n)}) = g(g(S_{c(k,l)}) \star g(S_{c(m,n)})) = g(p, q, a)g(\star)g(p, q, a) \tag{5.1}$$

$$= (a_1pq + a_2p + a_3q + a)(0, 1) = (0|1)\big|g(p, q, a);\ a \wedge a_{j=1,2,3} = 0/1. \tag{5.2}$$

例 5.2. 设合取复合语句 $S_{c(2,3)}$ 由两个条件子语句 $S^1_{c(1,2)}$ 和 $S^2_{c(1,2)}$ 构成，且为 $(p \Rightarrow q) \wedge (q \Rightarrow p)$。 那么我们有

$$g_{(p,q,1)}((p \Rightarrow q) \wedge (q \Rightarrow p)) = g(g(p \Rightarrow q) \wedge g(q \Rightarrow p))$$

$$(p(q + 1) + 1)g(\wedge)(p + 1)q + 1) = (p(q + 1) + 1) \cdot ((p + 1)q + 1)$$

$$= (p(q + 1) + 1) + ((p + 1)q + 1) = p + q + 1(1, 0) = 1(00, 11)|0(10, 01).$$

这里我们对 $1 \wedge 0$进行了求值；由于我们处于 g(p， q， 1) 中，因此得到数字串 (1001)。应用反函数 g^{-1}(p， q， 1)：

$$g^{-1}_{(p,q,1)}(p + q + 1) = p \Leftrightarrow q.$$

¯我们还看到，值域 $\mathcal{H}$ 是实值的。我们将多项式视为一个数；这并非代数中常用的那种含义——在那种情况下没有求值，并称为不定元——但这种看法不适用于此处。

注记 5.3。 这个迭代过程可以继续进行，直到所有子命题都处理完毕，且主联结词完成变换。我们把这一过程称为在 PBNF 中不使用任何推理规则。通常，我们会应用

代换，传统上它被视为一条推理规则。迭代过程或许能更清楚地说明我们的命题演算表述是如何运作的。我们还认为，这并非逻辑推理规则，而是数学领域中使用的线性，例如 $f(a+b) = f(a) + f(b)$。

映射 f 在奇异情形（$j=1$）下定义为

$$f(x_1, x_2) = (0, p, p+1, 1) \tag{5.3}$$

在布尔群 $\mathcal{B}_1 = \langle 0,\ p,\ p+1,\ 1$ 中；$+\rangle$. 在二元情形下，我们有一个布尔多项式矩阵环

$$\mathcal{B}_2 = \langle 0, \begin{bmatrix}00\\00\end{bmatrix} p, \begin{bmatrix}11\\00\end{bmatrix} q, \begin{bmatrix}10\\10\end{bmatrix} p+1, \begin{bmatrix}00\\11\end{bmatrix} q+1, \begin{bmatrix}01\\01\end{bmatrix} p+q, p+q+1, \begin{bmatrix}10\\01\end{bmatrix} \tag{5.4}$$

$$(p+1)q, (p+1)q+1, p(q+1), p(q+1)+1, \tag{5.5}$$

$$(p+1)(q+1)\begin{bmatrix}0\\01\end{bmatrix}, (p+1)(q+1)+1, pq, \begin{bmatrix}10\\00\end{bmatrix} pq+1, 1; +, \cdot, \rangle. \tag{5.6}$$

映射 $g^{-1}\colon \mathcal{H} \longrightarrow \mathcal{OP}$ 是纤维或拉回，由 $g(\boldsymbol{x})$的原像定义。请注意，为了封闭性，我们只要求矩阵乘法满足单侧乘法。

注记 5.4。这里我们必须谨慎，因为一元算子（$=,\ -,\ \neg,\ +$，我们将在下一篇文章中探讨）也可以定义为 $-:= p \wedge \neg p$和 $+:= p \vee \neg p$，这样我们就没有双射，因为

$$g^{-1}(p(p+1)) \mapsto -(00) \quad \text{and} \quad \iota_0(0000)$$

ι_1也同理。为避免这种情况，我们不会将这些二元多项式用于一元算子。

我们用 N 表示算子的数字串（不含逗号），例如，$N(p \Rightarrow q) \sim N(\Rightarrow) = (1011)$。两者的对应关系是 $g(1011) = p(q+1) + 1$，而纤维为 $g^{-1}(p(q+1)+1) = op_{\boldsymbol{x}(1011)}$.

定理 5.5。

$N(p.op_{\boldsymbol{x}}q)$	f中的串$(op_{\boldsymbol{x}})$	符号 $\in \mathcal{H}^*$
$(1)N(p \wedge q) = 1000$	$(1100)(1010) = (1000)$	pq
$(2_p)N(p,\ \neg p)$	$(1100),\ (0011)$	$p,\ p+1$
$(2_q)N(q,\ \neg q)$	$(1010), (0101)$	$q, q+1$
$(3)N(p \vee q) = 1110$	$1100 + 1010 + 1000 = 1110$	$(p+1)(q+1)+1$
$(4)N(p \Rightarrow q) = 1011$	$1000 + 1100 + 1111 = 1011$	$p(q+1)+1$
$(5)N(p \Leftarrow q) = 1101$	$1000 + 1010 + 1111 = 1101$	$(p+1)q+1$
$(6)N(p \Leftrightarrow q) = 1001$	$(1011)(1101) = 1001$	$p+q+1$
$(7)\ N(p \uparrow q) = 0111$	$(1100)(1010) + 1111 = 0111$	$pq+1$。
$(8)N(p \downarrow q) = 0001$	$1110 + 1111 = 0001$	$(p+1)(q+1)$。
$(9)N(p\|q) = 0110$	$1100 + 1010 = 0110$	$p+q$。

证明。为了找到这些多项式，我们从 pq 开始，将其命名为恒等多项式，并将其用作本原 多项式，同时假设

它映射为 (1， 0，0， 0)，我们以该算子为基础进行构造，并检验结果。我们按分量相乘（内积）

$$\langle p \cdot q \rangle = \langle 1100 \cdot 1010 \rangle = (1,0,0,0) \mapsto \wedge.$$

请注意，这不是标量积；结果是一个向量。现在，我们尝试通过相加 $p+q =$(1， 1， 0， 0) +(1， 0， 1，0) =(0， 1，1， 0) $\mapsto$ |来得到析取 $p \vee q$，结果是异或，因为 $1+1=0$，所以缺少交集。这意味着交集

$$\cap = pq \mapsto \wedge$$

是乘法，而并集

$$\cup \neq p+q \mapsto |, \quad \cup = p+q+pq \mapsto \vee$$

并非只是将 p和 q相加，而是通过使用我们最初采用的本原算子，将 (|+$\wedge$) $\mapsto p+q+pq =$(1， 1， 1， 0) 相加。我们也可以将其视为异或 |的定义。

$$p+q = (p \cap q') \cup (p' \cap q). \tag{5.7}$$

我们通常将 $pq+p+q$ 写作 $(p+1)(q+1)+1 \hookrightarrow \vee$，因为 $\vee$ 是齐次输入 $\left[\begin{smallmatrix}0\\0\end{smallmatrix}\right]$ 在 $\left[\begin{smallmatrix}1100\\1010\end{smallmatrix}\right]$ 中的补，求补的方式是加 1，即 ((0001) +(1111) = (1110))。我们喜欢在计算中使用这些粒子 (p+1) 和 (q+1)，因为当它们乘以与括号中的相同选择器时，两者均为 0 ($p(p+1) = 0$)。这样，我们证明了该命题，因为所有这些变换都可以逐一检验。我们注意到，使用了基本构件 $(0, p, q, p+1, q+1, 1)$，它们在 $\mathbb{Z}_2$ 中仅使用 $+$ 和 $\cdot$ 来填充表格。加 1 会对算子$+1 \leftrightarrow \neg op_x$取反；加上 p、q 和 1，则会逐分量相加 $p+q+1 =$(1, 1, 0, 0) +(1, 0, 1, 0)+(1, 1, 1, 1) =(0, 1, 1, 0) +(1, 1, 1, 1) =(1, 0, 0, 1)，再加上$1+1=0$，得到 $\Leftrightarrow$；或者也可以像上面那样使用乘法。这里我们注意到，所有形式都处于其补形式，因为除第一个以外，我们都加了 1。这表明可能存在一个不处于补形式的对偶空间。我们将在下一篇文章中讨论这种情况。要得到平凡算子 N(1)=1111 和 N(0)=0000 的多项式，我们可以取任意算子，并补上缺少的部分。□

例 5.6。 我们从 1110 开始，加上 0001 得到 1111，并且 h

多项式

$$(p+1)(q+1)+1+(p+1)(q+1) = 1.$$

我们还注意到，如果我们相加

$$N(op_{\boldsymbol{x}} + op_{\boldsymbol{x}}) = N(op_{\boldsymbol{x}}) + N(op_{\boldsymbol{x}}) = 0$$

因此我们看到

$$N : op_{\boldsymbol{x}} \longrightarrow x_1x_2x_3x_4; \quad x_n = 0,1$$

是一个线性算子，通过给出算子数字串来定义，并得到加法中性元作为结果。此外

$$op_{\boldsymbol{x}}^2 = op_{\boldsymbol{x}}$$

我们看到了幂等性。因此，在算子层面，这一切都是布尔的。

在下一篇文章中，除其他内容外，我们将讨论原始算子和多项式生成元。下一篇文章的其他内容包括：我们将在正规多项式族中加入另外三个对我们技术有用的多项式族，从而完成对命题演算的表述。对于线性多项式，我们将证明它们构成四元克莱因群的完整对称性，并且可以定义否定的典范分解[5]。对于二次多项式，我们将证明，等式（$p \uparrow p$） $= \neg p$在算子空间中是不可判定的，但在 PBNF 中是可判定的。我们将继续研究多项式族的原始算子和生成元。

参考文献

[1] 1978 Barkley Rosser J.《数学家逻辑学》，Dover，Bergmann, M.：《多值逻辑与模糊逻辑导论》，剑桥大学出版社，2008年。[3] Church, A.：《数理逻辑导论》，普林斯顿大学出版社，1956年。[4] Halmos, P.R：《布尔代数》，Martino 出版社，2013年。[5] Humberstone, L.：《迭代否定。》Theoria，61，1，1995年，第1—24页。[6] Kleene S.C.：《元数学导论》。D. Van Nostrand 公司，1952年。[7] Quine, W.V.：《数理逻辑》W.W. Norton 出版公司，1940年。[8] Stoll R. R.：《集合论与逻辑》Dover，1979年。[9] Tarski, A.：《逻辑学导论》，牛津大学出版社，2013年。[10] Veatch, W. S.《作为布尔代数的命题逻辑——一种全新视角，第*1*卷》，亚马逊出版社，2017年。

林奈大学电子邮件地址：
pelle.borgeke@lnu.se