arXiv:2302.04606v1 [cs.LO] 9 Feb 2023

关于发现有趣的组合整数序列

Martin Svato˘s[1] Peter Jung[1] Jan T´oth[1] Yuyi Wang[2,3]

Ond˘rej Ku˘zelka[1]

[1]布拉格捷克理工大学，捷克共和国[2]中车株洲研究所，中国[3]苏黎世联邦理工学院，瑞士{svatoma1, jungpete, tothjan2, ondrej.kuzelka}@fel.cvut.cz yuyiwang920@gmail.com

2023年1月

摘要

我们研究生成具有组合解释的有趣的整数序列这一问题。为此，我们提出了一种两步法。第一步，我们生成用于定义某些组合对象的一阶逻辑句子，例如无向图、排列、匹配等。第二步，我们使用提升的一阶模型计数算法，生成用于计算第一步生成的一阶逻辑公式所编码对象数量的整数序列。例如，如果一阶逻辑句子定义的是排列，那么生成的整数序列就是阶乘数序列 $n!$ 我们通过证明自动生成的序列中有相当一部分实际上可以在整数序列在线百科全书（OEIS）中找到，同时还能生成许多其他与之相似、但未收录于 OEIS 且可能具有趣味性的序列，来展示我们的方法能够生成有趣的新序列。我们工作的一项关键技术贡献，是提出了一种生成一阶逻辑句子的方法；该方法通过舍弃大量会导致冗余整数序列的句子，能够大幅削减句子的搜索空间。

1 引言

本文关注的是整数序列。顾名思义，整数序列 就是一列整数 $a_0, a_1, a_2, \ldots$，其中 $a_i \in \mathbb{Z}$ 对所有 $i \in \mathbb{N}$都成立。整数序列是基础的数学对象，几乎在数学的各个领域随处可见：从枚举组合学（在其中，整数序列用于统计具有特定性质的对象），到数学分析（在其中，整数序列通过泰勒级数定义函数），以及许多其他领域。甚至还有一部名为《在线整数序列百科全书》（OEIS）的百科全书，[1] 其前身为离线版本，由 Neil Sloane 于1964年创立（OEIS基金会，2023年）。截至2023年1月，其中收录了超过35.9万条整数序列。OEIS 收录的序列既受到专业数学家关注，也受到业余数学爱好者关注。

[1]https://oeis.org ；关于 OEIS 在数学中的作用的通俗介绍，也可参阅《广达》杂志上的文章 e: https://www.quanta magazine.org/neil-sloane-connoisseur-of-number-sequences-20150806/。

OEIS 数据库的一种典型用法如下。假设你正在研究一个组合问题，想要统计具有某种特定性质的 n 个顶点的无向图，例如所有顶点度数都等于 3。你成功计算出若干较小 n 值对应的这类图的数量，于是开始琢磨是否有人研究过同一数列。接着，你把算出的数值输入 OEIS 主页的搜索框并点击搜索。之后，所有 OEIS 中的匹配结果都会显示出来；如果你运气好，其中一个结果会告诉你一些关于这个问题的有趣信息——也许已经有人解决了它，或者至少算出了该序列的更多项。[2]

数列是如何进入 OEIS 的？被认为有趣的数列由用户手动提交至 OEIS。在这里，什么算是有趣的，在很大程度上显然是主观的。然而，这也是 OEIS 的一项局限——首次研究某个数列的人，在 OEIS 中查找它时并不能得到多少帮助。许多相当自然的数列并未收录于 OEIS。例如，正如 (Barv´ınek et al., 2021) 所指出的，OEIS 收录了关于用 2 种颜色进行适当着色的 2-正则图的计数数列，却没有收录关于用 3 种颜色进行适当着色的 2- 正则图的计数数列。OEIS 中还有许多类似的有趣数列缺失，这些数列或许对某些用户有用。这也正是我们开展本文工作的动机：我们开发了一种自动化方法，用于发现可能有趣的整数序列。

我们关注组合数列，即对满足某些给定性质、大小为 n 的对象进行计数的数列；这类数列是枚举组合学的研究对象 (Stanley, 1986年)。此类组合数列的例子包括对以下对象进行计数的数列： n元集合的子集、具有n 个顶点的图、具有 n 个顶点的连通图、具有 n 个顶点的树、没有不动点的 n 元排列等。具体而言，我们关注由可用一阶逻辑句子描述的结构所构成的组合数列。

研究用一阶逻辑表达的组合计数问题有几个优点。首先，尽管有时可能需要花些功夫，但人类用户能够理解一阶逻辑句子。例如，句子 $\forall x\ \neg R(x,\ x) \wedge \forall x \forall y\ R(x,\ y) \Rightarrow R(y,\ x)$ 可以理解为对无环无向图进行编码。其次，尽管计算一阶逻辑句子的模型数量通常难以处理（Beame 等人，2015），但一阶逻辑中存在一些经过充分研究的非平凡片段，可以高效地进行计数（Van den Broeck，2011；Van den Broeck、Meert 和 Darwiche，2014；Kuˇzelka，2021），而且已有快速实现（Van den Broeck，2011；van Bremen 和 Kuˇzelka，2021）。[3] 这意味着我们能够快速计算相应的组合数列。[4]

我们的方法分为两个阶段。首先，我们从一个易处理的片段中生成一阶逻辑句子。其次，我们计算每个生成句子对应的序列，并筛除会产生冗余序列的句子。事实证明，第一步至关重要。正如我们将在本文后文通过实验展示的那样，如果我们以朴素的方式生成句子，也就是说，如果尝试生成所有长度不超过 k 且语法形式各异的句子，我们就必须计算数量极其庞大的序列，其中大多数都是冗余的，以至于根本无法找到有趣的序列。因此，本文主要着重介绍我们系统中生成句子的组件。

本文其余部分安排如下。第2节介绍一阶逻辑的预备知识。第3节介绍我们构建句子数据库及相应整数序列的方法，并在第4节对其进行评估。第5节介绍相关工作，最后作出总结。

[2]例如，对于所有顶点度数均为 3 的无向图，OEIS 中的一个匹配项是序列 A002829：具有 $2n$ 个节点的三价（或三次）标记图的数量。[3]https://github.com/UCLA-StarAI/Forclift, https://www.comp.nus.edu.sg/~tvanbr/software/fastwfomc.tar.gz[4]对用于计数组合对象的整数序列进行计算复杂性研究，是枚举组合学中一个活跃的研究方向，参见例如（Pak，2018年）。

见第 6 节。

2 预备知识

我们使用一阶逻辑中不含函数的子集。该语言由有限集常量 Δ、有限集变量 $\mathcal{V}$以及有限集谓词 $\mathcal{P}$ 定义。一个原子式 的形式为 $P(t_1, \ldots, t_k)$，其中 $P \in \mathcal{P}$ 且 $t_i \in \Delta \cup \mathcal{V}$。一个文字 是原子式或其否定。一个公式 是一个原子式和一个文字。更复杂的公式可以通过逻辑联结词组合已有公式构成，也可以在其外加上全称量词（$\forall x$）或存在量词（$\exists x$），其中 $x \in \mathcal{V}$。公式中的变量 x 称为 $free$，如果该公式没有对 x进行量化。若公式不含自由变量，则称其为 $sentence$。若公式不含变量，则称其为基。

按照计算机科学领域的惯例，我们采用有限论域上的赫布兰德语义（Hinrichs and Genesereth，2006）。我们用 HB 表示赫布兰德基，即所有基原子的集合。我们用 ω 表示可能世界，即 HB 的任意子集。假定可能世界中的元素为真，其他所有元素为假。若句子 ϕ 在 ω 中成立，则可能世界 ω 是该句子的一个模型（记作 $\omega \models \phi$）。

2.1 加权一阶模型计数

为计算组合整数序列，我们采用加权一阶模型计数（WFOMC）问题（Van den Broeck 等人，2011年）。

定义 1。（加权一阶模型计数）设 ϕ 是某种关系语言 $\mathcal{L}$ 上的一个句子。令 HB 表示 $\mathcal{L}$ 在某个大小为 $n \in \mathbb{N}$ 的论域上的赫布兰德基。令 $\mathcal{P}$ 为语言 $\mathcal{L}$ 的谓词集合，并令 $\mathsf{pred} : \mathsf{HB} \mapsto \mathcal{P}$ 将每个原子式映射到其对应的谓词符号。令 $w : \mathcal{P} \mapsto \mathbb{R}$ 和 $w : \mathcal{P} \mapsto \mathbb{R}$ 为一对权重函数，分别为 $\mathcal{L}$ 中的每个谓词赋予正权重和负权重。我们定义

$$\mathsf{WFOMC}(\phi, n, w, \overline{w}) = \sum_{\omega \subseteq \mathsf{HB}: \omega \models \phi} \prod_{l \in \omega} w(\mathsf{pred}(l)) \prod_{l \in \mathsf{HB} \setminus \omega} \overline{w}(\mathsf{pred}(l)).$$

例 1. 考虑句子

$$\phi = \forall x \, \neg E(x, x)$$

以及权重 $w(E) = \overline{w}(E) = 1$。由于所有权重均为单位权重，我们只需计算 ϕ 的模型数量。我们可以将该句子解释如下：语言中的每个常量都是一个顶点。每个原子式 $E(A, B) \in \mathsf{HB}$ 且满足 $A, B \in \Delta$ 表示一条从 A 到 B 的边。此外，该句子禁止自反原子，即自环。总之，ϕ 的模型将是 n 个顶点上的所有无自环有向图。因此，我们得到

$$\mathsf{WFOMC}(\phi, n, w, \overline{w}) = 2^{n^2 - n}.$$

例 2. 考虑句子

$$\phi = \exists x \; Heads(x)$$

以及权重 $w(Heads) = 4$、$w(\neg Heads) = 1$。现在，我们可以将每个论域元素视为一次抛硬币的结果。该句子要求至少有一次抛硬币的结果为“注意力头”（存在一个常量 $A \in \Delta$，使得 $Heads(A)$ 是模型中的一个元素）。假设我们有 $i > 0$

个模型中的“注意力头”。那么，模型的权重将为 $4^i \cdot 1^{n-i} = 4^i$，并且将有 $\binom{n}{i}$ 个这样的模型。因此，

$$\mathsf{WFOMC}(\phi, n, w, \overline{w}) = \sum_{i=1}^{n} 4^i \cdot \binom{n}{i} = 5^n - 1.$$

2.2 二变量片段中的加权一阶模型计数

为了使计算易于处理，我们将每个句子中的变量数限制为最多两个。这种语言称为$\mathbf{FO}^2$，它允许在关于域大小的多项式时间内计算加权一阶模型计数（Van den Broeck，2011年；Van den Broeck、Meert 和 Darwiche，2014年）。我们将简要介绍这一可处理性结果。

以提升式方式计算加权一阶模型计数时，我们力求尽可能避免对问题进行基化。一阶句子的基化通常会使问题规模呈指数级增长，并且必然导致许多对称的子问题。

示例3。 考虑句子

$$\phi = \forall x \ (Smokes(x) \Rightarrow Cancer(x)).$$

在大小为 $n \in \mathbb{N}$ 的论域上对该句子进行基化，将产生由 n 个蕴含式构成的合取。每个蕴含式都有三个模型，这些模型中的原子式与其他蕴含式模型中的原子式完全不同。此外，不同蕴含式的模型之间存在双射。总的来说，我们本可以用简单得多的方式计算模型计数。对于某个特定常量，会有三个不同的模型。由于有 n 个常量，最终的模型计数将为 3^n。

为避免在此类对称实例上重复计算，我们旨在将加权一阶模型计数问题分解为相互独立的部分，使每个部分只需计算一次。待计算其加权一阶模型计数的逻辑句子的单元使这种分解成为可能。

定义2（单元）。 一个一阶逻辑[2] 公式 ϕ 的单元，是由 ϕ 中的原子式构成、且仅使用一个变量的极大一致合取。

示例4。 考虑公式

$$\phi = Smokes(x) \wedge Friends(x, y) \Rightarrow Smokes(y).$$

因此共有四个单元：

$$\begin{aligned}
C_1(x) &= Smokes(x) \wedge Friends(x, x), \\
C_2(x) &= \neg Smokes(x) \wedge Friends(x, x), \\
C_3(x) &= \neg Smokes(x) \wedge \neg Friends(x, x), \\
C_4(x) &= Smokes(x) \wedge \neg Friends(x, x).
\end{aligned}$$

为简化加权一阶模型计数的计算，我们按如下方式对单元进行条件化：

$$\begin{aligned}
\psi_{ij}(x, y) &= \phi(x, y) \wedge \phi(y, x) \wedge C_i(x) \wedge C_j(y), \\
\psi_k(x) &= \phi(x, x) \wedge C_k(x).
\end{aligned}$$

然后我们计算

$$r_{ij} = \mathsf{WMC}(\psi_{ij}(A,B), w', \overline{w}'),$$
$$w_k = \mathsf{WMC}(\psi_k(A), w, \overline{w}),$$

其中，WMC 只是 WFOMC 的命题版本，A，$B \in \Delta$ 并且权重（w'，w'）与（w，w）相同，但条件化单元中出现的原子式除外。这些权重设为 1，因为一元和二元自反原子的权重已计入 w_k 项中。注意，$r_{ij} = r_{ji}$。

现在，假设有 p 个不同的单元，我们可以写成

$$\mathsf{WFOMC}(\phi, n, w, \overline{w}) = \sum_{\mathbf{k} \in \mathbb{N}^p : |\mathbf{k}| = n} \binom{n}{\mathbf{k}} \prod_{i,j \in [p] : i < j} r_{ij}^{(\mathbf{k})_i (\mathbf{k})_j} \prod_{i \in [p]} r_{ii}^{\binom{(\mathbf{k})_i}{2}} w_i^{(\mathbf{k})_i}. \tag{1}$$

然而，上述方法仅适用于全称量化的一**阶逻辑**^{2} 句子。为了消除输入公式中的存在量词，我们可以使用专门针对加权一阶模型计数的斯科伦化（Van den Broeck、Meert 和 Darwiche，2014年）。该过程通过引入新的（新鲜）斯科伦谓词 Sk 来消除存在量词，其中 $w(Sk) = 1$且 $w(Sk) = -1$。

例 5。考虑公式

$$\phi = \forall x \exists y \; E(x,y),$$
$$\psi = \forall x \forall y \; \neg E(x,y) \vee Sk(x).$$

成立的是

$$\mathsf{WFOMC}(\phi, n, w, \overline{w}) = \mathsf{WFOMC}(\psi, n, w', \overline{w}'),$$

其中

$$w(E) = w'(E), \qquad w'(Sk) = 1,$$
$$\overline{w}(E) = \overline{w}'(E), \qquad \overline{w}'(Sk) = -1.$$

我们请读者参阅(Van den Broeck、Meert 和 Darwiche，2014年)以了解论证　.

结合专门的斯科伦化过程，根据公式（1），对于任意 一**阶逻辑**^{2} 句子，都可以在关于 n 的多项式时间内计算加权一阶模型计数。

2.3 带计数量词的二变量片段中的加权一阶模型计数

尽管一**阶逻辑**^{2} 的语言允许在多项式时间内计算加权一阶模型计数，但其表达能力自然相当有限。寻找更大的逻辑片段，同时仍能保持多项式复杂度，是当前活跃的研究课题。扩展一**阶逻辑**^{2} 并保留其易处理性的一种方法是加入计数量词。这种语言称为$\mathbf{C}^2$，Kuˇzelka（2021）证明了它具有易处理性。

计数量词是传统存在量词的一种推广。对于变量 $x \in \mathcal{V}$，我们允许使用形式为 $\exists^{=k} x$ 的量词，其中 $k \in \mathbb{N}$。[5] 含有计数量词的公式，其满足性可按自然方式定义。例如，$\exists^{=k} x \; \psi(x)$ 在 ω 中成立，当且仅当恰有 k 个常量$\{ A_1, A_2, \ldots, A_k \} \subseteq \Delta$ 满足 $\omega \models \psi(A_i)$，当且仅当 $1 \leq i \leq k$。

[5]Kuˇzelka（2021）实际上证明了更一般形式的计数量词具有易处理性，即 $\exists^{\bowtie k} x$，其中 $\bowtie \in \{ <, \leq, =, \geq, > \}$。然而，带不等式的计数并不容易扩展——事实上，即使是等式，在计算上也颇具挑战——因此我们只在计数量词中使用等式。

为了处理计数量词，Kuˇzelka（2021）建议在许多取值点（论域）上反复计算加权一阶模型计数。随后将这些值用于多项式插值。我们则使用符号权重；就我们的所有目的而言，这与多项式插值等价。我们直接得到原本通过插值所要得到的多项式。[6]

当我们要计算含有计数量词的句子的加权一阶模型计数时，首先要做的是将计数量词转换为传统存在量词。这可以通过另一种称为基数约束的句法结构来实现。我们允许公式包含形如 $(|P| = k)$，[7] 的原子公式，其中 $P \in \mathcal{P}$ 是谓词，且 $k \in \mathbb{N}$。直观地说，基数约束强制句子的所有模型中恰好包含 k 个谓词 P 的原子。

示例 6. 考虑以下句子

$$\phi = \forall x \exists^{=1} y\ E(x, y),$$
$$\phi' = (\forall x \exists y\ E(x, y)) \wedge (|E| = n).$$

那么有

$$\mathsf{WFOMC}(\phi, n, w, \overline{w}) = \mathsf{WFOMC}(\phi', n, w, \overline{w})$$

对于任意权重$(w, \overline{w})$。

利用示例 6 中所示的转换等方法，句子 ϕ 的$\mathbf{C}^2$ 加权一阶模型计数可化简为以下句子的加权一阶模型计数

$$\phi' = \psi \wedge \bigwedge_{i=1}^{m} (|P_i| = k_i),$$

其中 ψ 是一个一**阶逻辑**[2] 句子。然后，对于每个基数约束 $(|P_i| = k_i)$，我们定义 $w'(P_i) = x_i$，其中 x_i 是一个新的符号变量。对于未出现在任何基数约束中的谓词 $Q \in \mathcal{P}$，我们保持其正权重不变，即 $w'(Q) = w(Q)$。

最后，我们已经可以计算 $\mathsf{WFOMC}(\psi, n, w', \overline{w})$。结果将是一个多元多项式，其变量是为每个基数约束引入的符号变量。不过，只有其中一个单项式包含原始$\mathbf{C}^2$ 句子的实际加权一阶模型计数信息。[8] 也就是说，该单项式

$$A \cdot \prod_{i=1}^{m} x_i^{e_i}$$

使得 $e_i = k_i$ 对每个基数约束 $(|P_i| = k_i)$ 都成立。

现在，我们可以报告原始$\mathbf{C}^2$ 句子 ϕ的最终加权一阶模型计数结果。不过，在处理基数约束时，我们仍须考虑那些被符号变量替代的正权重。因此，

$$\mathsf{WFOMC}(\phi, n, w, \overline{w}) = A \cdot \prod_{i=1}^{m} w(P_i)^{k_i}.$$

[6]在定义 1 中，我们仅针对实值权重定义了加权一阶模型计数。不过，将其扩展到（多元）多项式是很自然的，也不会造成任何问题。[7]与计数量词类似，基数约束可以推广为 $(|P| \bowtie k)$，其中 $\bowtie \in \{<, \leq, =, \geq, >\}$。完整论述请参见（Kuˇzelka，2021）。[8]使用带有 $k > 0$ 的计数量词 $\exists^{=k}$ 时，我们还需要注意避免重复计数，详见（Kuˇzelka，2021）。

示例 7。 考虑句子

$$\phi = \forall x \exists^{=1} y \; E(x, y),$$

大小为 $n = 5$ 且 $w(E) = w(E) = 1$的论域。让我们计算 加权一阶模型计数$(\phi, n, w, \overline{w})$。

首先，我们去掉计数量词：

$$\phi' = (\forall x \exists y \; E(x, y)) \wedge (|E| = 5)$$

其次，我们引入一个符号变量 x，作为 E 谓词的正权重：

$$w'(E) = x$$

最后，我们计算 加权一阶模型计数$(\forall x \exists y \; E(x, y) \;\; n, \overline{w'}, w)$，并提取其中 x 的五次幂对应项的系数：

$$\mathsf{WFOMC}(\phi, n, w, \overline{w}) = 3125.$$

让我们检验所得结果。我们可以将公式 ϕ 解释为一个有向图，其中每个顶点恰有一条出边。对于每个顶点，它都有 n 个顶点可以与之相连。因此，这样的图共有 n^n 个。对于 $n = 5$，我们得到 $5^5 = 3125$。

3 构建序列数据库

我们的目标是构建一个数据库，其中包含一阶逻辑句子，以及由这些句子生成的相应整数序列。我们不打算让数据库中的句子面面俱到。对于任意整数序列，可能有许多句子都能生成它[9] 而我们只希望每个整数序列对应一个句子。如果多个句子生成相同的整数序列，我们就称它们为冗余。我们通常会尽量避免生成冗余句子。

数据库分两步构建。第一步，我们生成一阶逻辑句子；第二步，我们计算这些句子的模型数量所对应的整数序列。本节将按相反顺序介绍这两个步骤。首先，我们介绍如何从一阶逻辑句子计算出这些序列，我们将其称为组合谱；这可以通过现有的提升推理方法完成。然后，我们介绍一种全新的句子生成算法，力求尽可能少地生成冗余句子。

3.1 计算整数序列

给定一个一阶逻辑句子，我们需要计算一个数列，使其第 k 项为论域大小为 k 的关系句子的模型计数。使数列项非零的域大小集合称为该句子的 谱 。逻辑句子 ϕ 的谱 是指作为某个有限模型的大小而出现的自然数集合 ϕ （B¨orger、Gr¨adel 和 Gurevich，2001）。由于我们所求的数列在某种意义上建立在谱的基础之上，而且也可以将该数列描述为原句子的组合解释结果，因此我们将该数列称为该句子的组合谱。

定义 3（组合谱）。 逻辑句子 ϕ 的组合谱，记作$\mathfrak{S}(\phi)$，是句子 ϕ 在大小依次为 1、2、3、4..... 的有限论域上的模型计数序列。

[9]显然，如果允许任意谓词名称，我们甚至可以说，有无穷多个句子会生成相同的数列。

例8。 再次考虑句子

$$\phi = \exists x \; Heads(x).$$

因此，HB 的所有子集（空集除外）都是 ϕ 的模型。因此，对于大小为 n的论域，将有 $2^n - 1$ 个模型，即

$$\mathfrak{S}(\phi) = 1, 3, 7, 15, \ldots$$

组合谱可以使用加权一阶模型计数算法计算。在本文中，我们使用了自己实现的 (van Bremen 和 Kuˇzelka，2021) 中的算法。该算法是当前最先进的算法，运行时间关于 n 为多项式级，适用于一**阶逻辑**^{2}。我们将其与自己实现的 (Kuˇzelka，2021) 中的归约结合使用，这些归约使我们能够在关于 n为多项式级的时间内计算任意$\mathbf{C}^2$ 句子的谱。

3.2 生成一阶逻辑句子

一般而言，我们旨在生成具有以下句法形式的句子：[10]

$$\bigwedge_{\substack{Q_1 \in \{\forall, \exists, \exists^{=1}, \ldots, \exists^{=K}\}, \\ Q_2 \in \{\forall, \exists, \exists^{=1}, \ldots, \exists^{=K}\}}} \bigwedge_{i=1}^{M} Q_1 x Q_2 y \; \Phi_i^{Q1,Q2}(x, y) \wedge \bigwedge_{Q \in \{\forall, \exists, \exists^{=1}, \ldots, \exists^{=K}\}} \bigwedge_{i=1}^{M'} Q x \; \Phi_i^{Q}(x) \tag{2}$$

其中，每个 $\Phi^{Q1}{}_{,i}{}^{Q2}(x, y)$都是一个无量词析取式，仅包含逻辑变量$x$ 和 y；同样，每个 $\Phi_i^Q(x)$ 都是一个无量词析取式，仅包含逻辑变量 x。整数 K、 M 和 M' 是参数。

具有形式（2）的句子示例如下：

- $\forall x \exists^{=1} y \; R(x, y) \wedge \forall x \exists^{=1} y \; R(y, x)$,
- $\forall x \; \neg R(x, x) \wedge \forall x \forall y \; \neg R(x, y) \vee R(y, x)$.

这里，第一个句子定义了双射（即排列），第二个句子定义了无环无向图。

注 1. 我们将稍微宽泛地使用术语，并用子句 来指称形式为 $Q_1 x Q_2 y \; \Phi^{Q1}{}_{,i}{}^{Q2}(x, y)$ 和 $Qx \; \Phi_i^Q(x)$的量化析取式，尽管子句 一词通常仅用于全称量化析取式。

[10]文献中至少有两种适用于 $\mathbf{C}^2$ 的斯科特范式（Gr¨adel 和 Otto，1999年；Pratt-Hartmann，2009年），这使我们可以使用更少的量词前缀。然而，这些范式并非为组合计数而设计——它们的设计目的仅是保证 $\mathbf{C}^2$ 句子与其范式之间的等可满足性，并不保证组合等价。因此，在本文中它们无法直接为我们所用。

3.2.1 我们是否覆盖了 $\mathbf{C}^2$ 的全部?

一个自然的问题是：如果我们将句子限制为形式 (2)，能否得到所有可能的$\mathbf{C}^2$ 句子的组合谱? 答案似乎是否定的，接下来我们会作出解释；但在我们看来，这一点并不重要，因为本文为自己设定的任务并非生成$\mathbf{C}^2$ 句子的所有组合数列——无论如何，这都是不可行的，因为作为$\mathbf{C}^2$ 句子的谱所生成的不同整数序列有无穷多个。[11] 相反，我们希望在有限的时间预算内，尽可能生成更多有趣的 整数序列。

现在我们简要说明，为什么形式为（2）的句子并不能保证我们能够找到所有 $\mathbf{C}^2$ 组合谱。首先，我们不能依赖（Gr¨adel 和 Otto，1999；Pratt-Hartmann，2009）中的范式，因为它们并非旨在保持模型计数。尽管（Ku˘zelka，2021）中提出的转换能够将任意 $\mathbf{C}^2$ 句子的模型计数计算，归约为对形式为（2）的句子的计算，但它要求某些谓词具有负权重。我们不允许生成的句子中包含负权重，因为这会使对这些句子进行事后组合解释变得显著更加困难。

3.2.2 遍历句子空间

我们使用标准的广度优先搜索算法来遍历 $\mathbf{C}^2$ 句子的空间。该算法从空句子开始。在搜索树的每一层中，它生成所有可通过向前一层生成的某个句子添加一个文字而获得的句子。该文字可以添加到现有子句中，也可以作为一个新子句添加到句子中；在后一种情况下，还需要在其前加上量词。

例 *9.* 假设我们有句子 $\varphi = \forall x \exists y\ R(x,y)$，并希望对其进行扩展。还假设我们的语言中只有二元谓词 R，且允许使用的量词只有 $\forall$ 和 $\exists$（为简单起见）。要扩展 φ，我们可以先选择向子句 $\forall x \exists y\ R(x,y)$ 添加一个新的 R-文字。共有 *8* 种添加方式，得到以下句子：$\varphi_1 = \forall x \exists y (R(x,y) \vee R(x,x))$, $\varphi_2 = \forall x \exists y (R(x,y) \vee R(x,y))$, $\varphi_3 = \forall x \exists y (R(x,y) \vee R(y,x))$, $\varphi_4 = \forall x \exists y (R(x,y) \vee R(y,y))$, $\varphi_5 = \forall x \exists y (R(x,y) \vee \neg R(x,x))$, $\varphi_6 = \forall x \exists y (R(x,y) \vee \neg R(x,y))$, $\varphi_7 = \forall x \exists y (R(x,y) \vee \neg R(y,x))$, $\varphi_8 = \forall x \exists y (R(x,y) \vee \neg R(y,y))$。第二种方法是创建一个新的单文字子句，并将其添加到 φ 中。在这种情况下，我们有以下情况：$\varphi_9 = \forall x \exists y\ R(x,y) \wedge \forall x\ R(x,x)$, $\varphi_{10} = \forall x \exists y\ R(x,y) \wedge \exists x\ R(x,x)$, $\varphi_{11} = \forall x \exists y\ R(x,y) \wedge \forall x\ \neg R(x,x)$, $\varphi_{12} = \forall x \exists y\ R(x,y) \wedge \forall x\ \neg R(x,x)$，接下来是以下类型的句子：$\forall x \exists y\ R(x,y) \wedge Q_1 x Q_2 y\ R(x,y)$, $\forall x \exists y\ R(x,y) \wedge Q_1 x Q_2 y\ R(y,x)$, $\forall x \exists y\ R(x,y) \wedge Q_1 x Q_2 y\ \neg R(x,y)$, 以及 $\forall x \exists y\ R(x,y) \wedge Q_1 x Q_2 y\ \neg R(y,x)$，其中 $Q_1, Q_2 \in \{\forall, \exists\}$。

从这个例子可以看出，即使句子的一阶语言中仅包含一个二元谓词，分支因子仍然很大。然而，如果我们实际计算这些句子的组合谱，就会发现其中许多句子是冗余的（我们将在下一小节中给出这个术语的精确定义）。此外，如果我们能够在不计算这些句子的谱的情况下，检测出哪些句子是冗余的，就能节省大量时间。这是因为，尽管组合谱的计算在 n 中是多项式时间的，但计算开销仍然很大。而且，如果我们能够从搜索中移除一些句子，同时保证仍会生成所有非冗余句子，就能节省更多时间。在本节余下部分，我们

[11] 这一点很容易看出。事实上，即使是$\mathbf{FO}^2$ 句子也会生成无穷多个整数序列。例如，考虑形如$\varphi_k\ \varphi_k = \exists x \bigvee_{i=1}^{k} U_i(x)$的句子。它们的组合谱为 $\mathfrak{S}(\varphi_k) = \left((2^k)^n - 1\right)_{n=1}^{\infty}$。因此，即使对于这些简单的句子，我们也有无穷多个组合谱——每个 $k \in \mathbb{N}$对应一个。

将介绍这类技术——有些技术仅将句子标记为冗余，在这种情况下，我们不会计算它们的组合谱；有些技术还会将句子标记为可安全删除，在这种情况下，我们甚至不会用它们来生成新句子。当我们要指代那些冗余但不可安全删除的句子时，将使用不可安全删除 这一术语。

3.2.3 序列冗余意味着什么?

给定一个句子集合 $\mathcal{S}$，如果存在另一个句子$\varphi' \in \mathcal{S}$，且 $\mathfrak{S}(\varphi) = \mathfrak{S}(\varphi')$，那么句子 $\varphi \in \mathcal{S}$ 就被视为冗余，也就是说，另一个句子会生成相同的整数序列。由于判断两个句子是否具有相同的组合谱在计算上很困难（详见附录），我们只会寻找两个句子生成相同谱的充分条件。

除了上述冗余概念外，如果存在另外两个句子 φ'、φ''，使得 $\mathfrak{S}(\varphi) = \mathfrak{S}(\varphi') \cdot \mathfrak{S}(\varphi'')$，我们也将句子 $\varphi \in \mathcal{S}$ 视为冗余，其中乘积 $\cdot$ 是逐元素计算的。其理由是，当这种情况发生时，φ 的模型集合很可能对应于 φ' 和 φ'' 的模型集合的笛卡尔积中的元素（或者至少两者之间存在双射），这在组合学上并不十分有趣。[12]

3.2.4 检测冗余句子

现在我们已经解释了冗余句子的含义，可以继续探讨如何检测一个句子是否冗余，以及如果冗余，它是否也可安全删除。首先要强调的是，本节介绍的方法无法保证检测出所有冗余。另一方面，这些方法是可靠的——它们不会把非冗余句子标记为冗余。有些技术会将句子标记为冗余，但不会提供冗余的见证，即具有相同组合谱的其他句子。对于那些能够保证见证句子更短（文字数量更少）的技术，情况就是如此；更短的句子必定更早生成，因此我们知道，删去较长的冗余句子不会影响搜索的完备性。

我们的句子生成算法所使用的剪枝方法总结于表1。其中一些方法相当直接，无需在此作进一步论证。

称为同构句子的方法，是对数据挖掘文献中已知的枚举非同构模式的方法的一种直接扩展（例如参见（Nijssen and Kok，2001））。主要区别在于，检查同构时，我们允许对谓词重新命名（相关细节在附录中给出，以确保完备性）。

称为可分解句子的方法基于以下观察；这一观察在提升推理文献等领域广为人知（Van den Broeck，2011）：设 $\varphi = \varphi_1 \wedge \varphi_2$ 是一个一阶逻辑句子。如果 φ_1 和 φ_2 使用互不相交的谓词集，那么不难证明 $\mathfrak{S}(\varphi) = \mathfrak{S}(\varphi_1) \cdot \mathfrak{S}(\varphi_2)$，其中乘积是逐元素计算的；并且，$\mathfrak{S}(\varphi_1)$ 和 $\mathfrak{S}(\varphi_2)$ 分别只在由 φ_1 和 φ_2 中所含谓词构成的语言上计算。

虽然名为 置换论元 的方法本身可能不需要更详细的解释，但为了帮助大家更好地理解，我们仍将在此通过一个例子加以说明。假设我们有两个句子：$\varphi_1 = \forall x \exists y\ E(x,\ y)$ 和 $\varphi_2 = \forall x \exists y\ E(y,\ x)$。第一个句子可以解释为对不存在出度为 0 的顶点的有向图进行建模，第二个句子则可以解释为对不存在入度为 0 的顶点的有向图进行建模。这种解释基于我们将 $E(x,\ y)$ 理解为

[12]毕竟，人们总是可以在后处理中创建这样的序列，并在需要时将其解释为相应笛卡尔积中的元素。

表1：用于生成非冗余句子的算法所采用的剪枝技术 s.

剪枝技术名称	**思想简述**
使用下文所述技术检测出的句子可安全删除。	
同构句子	如果一个句子可以由另一个句子得到，则这两个句子是同构的通过重命名变量和谓词名称。详情请见附录。
可分解句子	如果句子 φ 可以写成两个合取 $\varphi = \varphi' \wedge \varphi''$ 的谓词集互不相交的合取，那么 φ 就是冗余的。请参见正文中的论证。
永真式与矛盾式	任何包含恒真（即重言式）子句的句子都是冗余的——存在一个更短的句子，具有相同的组合谱。所有不可满足句子（矛盾式）产生由零组成的相同组合谱，因此也冗余 .
否定	如果两个句子可以通过对某些谓词的所有出现的文字取反而变为同构，那么它们会生成相同的组合谱。例如， $\mathfrak{S}(\forall x \exists y\ R(x,y)) = \mathfrak{S}(\forall x \exists y\ \neg R(x,y))$.
置换论元	论元翻转是对谓词 R 进行的一种变换，它将所有出现的 $R(x,y)$ 替换为 $R(y,x)$，并将所有出现的 $R(y,x)$ 替换为 $R(x,\ y)$。如果两个句子 φ 和 φ' 可以通过以下方式变为同构论元翻转，那么它们生成相同的组合谱，即，$\mathfrak{S}(\varphi) = \mathfrak{S}(\varphi')$。
通过下文所述技术检测到的句子不可安全删除。我们不会计算这些句子的组合谱，也不会将它们存储在数据库中。	
平凡约束	假设句子 φ 包含形式为 $\forall x$的子句 $U(x)$ 或 $\forall x \forall y\ R(x,\ y)$，我们称之为平凡约束。那么该句子 φ' 由 φ 去掉平凡约束，并将所有 U 或 R的出现分别替换为*true*后，具有相同的组合谱，如 φ。
自反原子	如果二元文字 R 在句子 φ 中只以 $R(x,x)$、$\neg R(x,x)$ $R(y,y)$ 或 $\neg R(y,y)$ 的形式出现，并且 φ 至少含有两个文字，那么该句子是冗余的。 论证请参见正文。
包含	如果句子 φ 包含两个子句 $Q_1 x Q_2 y\ \alpha(x,\ y)$，并且 $Q_1 x Q_2 y\ \beta(x,\ y)$，具有相同的量词前缀，并且如果存在一个子-替换 θ： $\{x,y\} \to \{x,y\}$ 使得 $\alpha\theta \subseteq \beta$，那么 φ 是冗余的——由删除 $Q_1 x Q_2 y\ \beta(x,\ y)$ 而从 φ 得到的句子 φ' 生-成相同的组合谱。
单元图同构	请参见正文。

一条从 x 到 y的边，不过我们也可以将其解读为一条从 y 到 x 的边，这不会改变该句子的组合谱（组合谱并不取决于我们如何解读句子）。推广这一观察后，我们会发现，仅在某些谓词的参数顺序上有所不同的句子（例如上文的 φ_1 和 φ_2 ）必然会生成相同的组合谱。

接下来，我们进一步介绍一种称为自反原子的方法。如果句子 φ 中某个二元谓词 R的原子仅以 $R(x,x)$ 或 $R(y,y)$ 的形式出现，那么所有基原子 $R(i,j)$（其中 i 和 j 是域元素，且 $i \neq j$）都不受 φ约束。因此， $\mathfrak{S}(\varphi) = \mathfrak{S}(\varphi') \cdot \mathfrak{S}(\varphi'')$，其中 $\varphi' = \forall x\ \neg R(x,x)$，而 φ''是通过将 $R(x,x)$ 的所有出现替换为$U_R(x)$，并将 $R(y,y)$ 的所有出现替换为 $U_R(y)$ 得到的句子，其中 U_R 是一个新谓词。这里， φ' 涵盖了参数为 $i \neq j$的原子 $R(i,j)$ 的所有可能配置。因此，这样的句子 φ 是冗余的。

方法永真式与矛盾式、否定、平凡约束和包含无需进一步解释，表1中已有说明。唯一剩下的方法，即单元图同构，将在下一小节中介绍。

3.2.5 单元图同构

表1中称为单元图同构的最后一种剪枝方法，依赖于提升推理领域中的一个概念。这个概念最初旨在更高效地计算公式（1）。在（van Bremen 和 Kuˇzelka，2021）中，作者引入了一种称为单元图的特殊结构，以帮助更快地计算加权一阶模型计数。

定义 4 (单元图**)。** 句子 ϕ 的单元图 G_ϕ 是一个完全图(V, E)，满足

1. V是单元标签 $\{1, 2, \ldots, p\}$的集合，
2. 每个节点 $i \in V$都有一个标签 w_i，
3. 每条边（包括自环）均从节点 i 指向 j，并带有标签r_{ij}。

从公式（1）可以看出，加权一阶模型计数的计算完全由项r_{ij}和 w_k决定。即使加入计数量词，这一点仍然不变，因为此时只是进一步搜索公式（1）的符号结果，以寻找特定的单项式。因此，计算完全由单元图决定，其中包含所有r_{ij} 和 w_k 值。

基于这一观察，我们提出了一种基于两个单元图同构的剪枝技术。如果两个句子的单元图同构，那么它们的加权一阶模型计数结果将相同，因而它们的组合谱也将相同。下面我们将这些论断形式化。

我们首先讨论所有权重均为实数的简单情形。这足以将该剪枝方法应用于$\mathbf{FO}^2$ 句子；随后，我们将其扩展到具有符号权重的情形，以便正确处理来自$\mathbf{C}^2$的句子。

首先，我们定义单元图同构的含义。

定义*5*（具有实值权重的单元图同构）。设 G 和 G' 是两个单元图，其中每条边 $\{i,j\} \in E$（G）（分别为 $\{i',j'\} \in E$（G'））都标记有实值权重 r_{ij}（分别为 $r'_{i'j'}$），且每个顶点 $i \in V$（G）（$i' \in V$（G'））都标记有实数 w_i（$w'_{i'}$）。如果存在一个双射 $f:\ V$（G）$\rightarrow \rightarrow V$（$G'$），使得对于所有 i、$j \in V$（G），$w'_i = w_{f(i)}$ 且 $r'_{ij} = r_{f(i)、f(j)}$，则称 G 和 G' 同构。

为了利用这一同构，我们将利用 C1 的以下有趣性质。

注释 1. 再次令 p 表示方程(1) 中的单元数，并令

$$f : [p] \to [p]$$

是一个双射。那么以下等式成立：

$$\sum_{\boldsymbol{k}\in\mathbb{N}^p:|\boldsymbol{k}|=n} \binom{n}{\boldsymbol{k}} \prod_{i,j\in[p]:i<j} r_{ij}^{(\boldsymbol{k})_i(\boldsymbol{k})_j} \prod_{i\in[p]} r_{ii}^{\binom{(\boldsymbol{k})_i}{2}} w_i^{(\boldsymbol{k})_i} =$$
$$\sum_{\boldsymbol{k}\in\mathbb{N}^p:|\boldsymbol{k}|=n} \binom{n}{\boldsymbol{k}} \prod_{i,j\in[p]:i<j} r_{f(i),f(j)}^{(\boldsymbol{k})_i(\boldsymbol{k})_j} \prod_{i\in[p]} r_{f(i),f(i)}^{\binom{(\boldsymbol{k})_i}{2}} w_{f(i)}^{(\boldsymbol{k})_i}.$$

换言之，在保持权重结构不变的同时置换单元，不会改变最终结果。

接下来，我们给出一个结果，以证明可以对一**阶逻辑**^{2} 句子使用单元图同构方法。

定理 1. </style}令和为两个一阶逻辑^2 句子，其权重分别为$(w,\overline{w})$ </style}和(w', w')，并令G_ϕ 和 G_ψ 分别为它们的单元图。若 G_ϕ 与 G_ψ同构，则

$$\mathsf{WFOMC}(\phi, n, w, \overline{w}) = \mathsf{WFOMC}(\psi, n, w, \overline{w})$$

对于任意域大小 $n \in \mathbb{N}$，均成立。

证明概要。 证明基于以下观察：令 f 为保权重的双射 f： $V(G_\phi) \to V(G_\psi)$；根据单元图同构的定义，这样的双射必然存在。考虑根据句子的单元图计算加权一阶模型计数的公式，即公式（1）。如果我们对单元索引（公式中的i和 j）应用双射 f，就会将计算 ϕ 的加权一阶模型计数的公式转换为计算 ψ 的公式（同样是因为 f 是保权重的双射）。由注释 1 可知，这两个公式必然相同，因此，对于任意域大小 n， ϕ 和 ψ 的加权一阶模型计数必然相等。

□

接下来，我们将单元图同构方法扩展到$\mathbf{C}^2$ 句子。为此，我们首先需要扩展单元图和单元图同构的定义。

定义 6 (带基数约束的单元图**)。** 句子 ϕ 的单元图 G_ϕ 是一个二元组 (G, C)，由以下部分组成：

1. 一个完全图 $G = (V, E)$，满足

   (a) V是单元标签 $\{1, 2, \ldots, p\}$的集合；*(b)* 每个节点 $i \in V$都有一个标签 w_i；*(c)* 每条从节点 i 到 j 的边（包括自环）都有一个标签 r_{ij}。

   这里，权重 w_i 和 r_{ij} 通常是多元多项式。

2. 表示基数约束的单项式集合 C。

例10. 考虑句子

$$\phi = (\forall x\ \neg E(x,x)) \wedge (\forall x \forall y\ \neg E(x,y) \vee E(y,x)) \wedge (|E| = 10)$$

它对具有 *5* 条边的无向图建模。只有一个单元与 ϕ 这个句子一致：$\neg E(x,x)$。正如我们在示例 *7* 中所见，要编码基数约束 $|E| = 5$，我们需要引入符号权重 $w(E) = x$。因此，单元图由权重 $w_1 = 1$、$r_{1,1} = 1 + x^2$ 所构成的图，以及表示基数约束的单元素集合 $C = \{x^{10}\}$ 构成。

现在我们可以给出带基数约束的 $\mathbf{FO}^2$ 句子的单元图同构定义（这正是编码$\mathbf{C}^2$ 句子所需的全部内容）。

定义*7*（单元图同构：具有符号权重和基数约束的图）。设 (G, C) 和 (G', C') 是两个带基数约束的单元图，其中每条边 $\{i,j\} \in E(G)$（分别为 $\{i',j'\} \in E(G')$）都标记有一个多元多项式 r_{ij}（分别为 $r'_{i'j'}$），并且每个顶点 $i \in V(G)$（$i' \in V(G')$）都标记有一个多元多项式 $w_i\,(w'_{i'})$。如果存在一个双射 $f: V(G) \to V(G')$，以及另一个双射 g，将多项式 (G, C) 中出现的变量映射到多项式 (G', C') 中出现的变量，并满足以下条件，我们就称 G 和 G' 同构：

1. $w'_i = g(w_{f(i)})$,
2. $r'_{ij} = g(r_{f(i),f(j)})$ *for all* $i, j \in V(G)$,
3. $C' = g(C)$.

上述定义比$\mathbf{FO}^2$ 句子的单元图定义更复杂，因为我们需要确保在找到单元图的同构时，不会“破坏”基数约束。

最后，我们可以正式证明，单元图同构也可用于$\mathbf{C}^2$ 句子。

定理2. </style}令 和 为两个$\mathbf{C}^2$ 句子， ϕ' 和 ψ' 为它们编码得到的带基数约束的$\boldsymbol{FO}^2$ 句子。令$(G_{\phi'}, C_{\phi'})$ 和$(G_{\psi'}, C_{\psi'})$ 分别为它们对应的带约束单元图。若$(G_{\phi'}, C_{\phi'})$ 与$(G_{\psi'}, C_{\psi'})$同构，则

$$\mathsf{WFOMC}(\phi, n, w, \overline{w}) = \mathsf{WFOMC}(\psi, n, w, \overline{w})$$

对于任意域大小 $n \in \mathbb{N}$ 以及任意权重(w, w)。¯

证明。 证明是定理 1 证明的直接扩展。 □

因此，只需从单元图同构所诱导的每个等价类中输出一个句子即可。然而，正如表1中所述，单元图同构的句子并非可安全删除，这意味着组合谱只针对诱导等价类中的一个成员进行计算，但所有成员都会用于进一步扩展搜索空间。

4 实验

本节通过实验评估第3.2节所述构建整数序列数据库的技术能否在合理时间内有效运行。此外，我们将仔细考察一些生成的$\mathbf{C}^2$ 句子，它们的组合谱似乎出现在 OEIS 中。最后，我们还会研究一些组合谱未出现在 OEIS 中的句子。

4.1 填充整数序列数据库

我们分别对$\mathbf{FO}^2$ 句子和$\mathbf{C}^2$ 句子的生成器进行了两项独立实验。我们将每个句子的组合谱计算时间限制设为五分钟；这些实验的结果分别见图1和图2，对应$\mathbf{FO}^2$ 和$\mathbf{C}^2$, 。这些实验旨在评估我们提出的剪枝技术所产生的影响。

我们从一个基线方法开始，该方法仅过滤掉同构句子（采用模式挖掘文献中使用的标准同构概念，该概念不考虑谓词重命名（Nijssen and Kok，2001）），并剪除可分解句子——这些是任何合理方法都可能实现的基本功能。随后，我们在基线方法上加入了永真式与矛盾式。以类似的方式，我们按照以下顺序，在前一种方法的基础上逐一添加单项剪枝技术：同构句子、否定、置换论元、自反原子、包含、平凡约束和单元图同构。可以看出，我们的方法将运行时间和生成的句子数量都降低了几个数量级。

这些剪枝技术通过两种方式帮助扩大填充数据库的规模。朴素方法（例如基线）虽然能快速生成大量句子，但很快就会耗尽所有可用内存；可安全删除技术则显著降低了内存需求。所有剪枝技术都会消耗一些计算时间，但与计算组合谱所需的时间相比，这点开销微不足道；计算组合谱是这项任务中最耗时的部分，分别参见图1c和图1b。由于剪枝方法（包括不可安全删除的方法）减少了组合谱的计算次数，因此很快就能抵消其开销；这一点从图1b和图2b中可以清楚看出，这两幅图展示了填充数据库所需的估算[13] 时间。

图1a和图2a还显示了唯一组合谱数量的下界，[14]即用相同数量的唯一整数序列填满数据库所需的最少非冗余句子数。

关于实验设置的详细信息，请参阅附录 B.3；关于剪枝技术的更多实验，请参阅附录 C。

4.2 初始数据库构建

除了比较所提出的剪枝方法优势的实验外，我们还使用算法生成了一个组合数列初始数据库。为此，我们让句子生成器在一台配备 500 GB RAM、128 个处理器（我们使用了多线程）的机器上运行五天，收集句子及其组合谱。计算一个数列的组合谱时限为五分钟。

最终得到的数据库包含超过 26,000 个唯一整数序列。对于数据库中的每个序列，我们都查询了 OEIS，以确定该序列是否与 OEIS 中已有的某个序列相匹配。我们发现其中有 301 个序列已收录于 OEIS——这占我们生成序列的 $\approx$1.2%。这个比例看似不高，但绝不可忽视。此外，我们的目标主要是生成新的序列。表2展示了几个生成出来且恰好已收录于 OEIS 的有趣序列。

一个有趣序列的例子是表2中的最后一个序列。OEIS 中没有该序列的组合刻画。我们可以从$\mathbf{C}^2$ 句子中得到这样的刻画

[13]由于未使用全部剪枝技术的方法会生成数量极高的句子（其中大多数都是冗余的），计算它们的谱将耗费数千小时。因此，对于这些方法，我们仅通过计算随机样本中的句子谱来估算运行时间。[14]计算包含五个以上文字的句子的谱十分耗时，因此我们只通过随机抽取较长句子来估算上层级别的结果。因此，我们仅展示已完成全面评估的级别的谱下界。

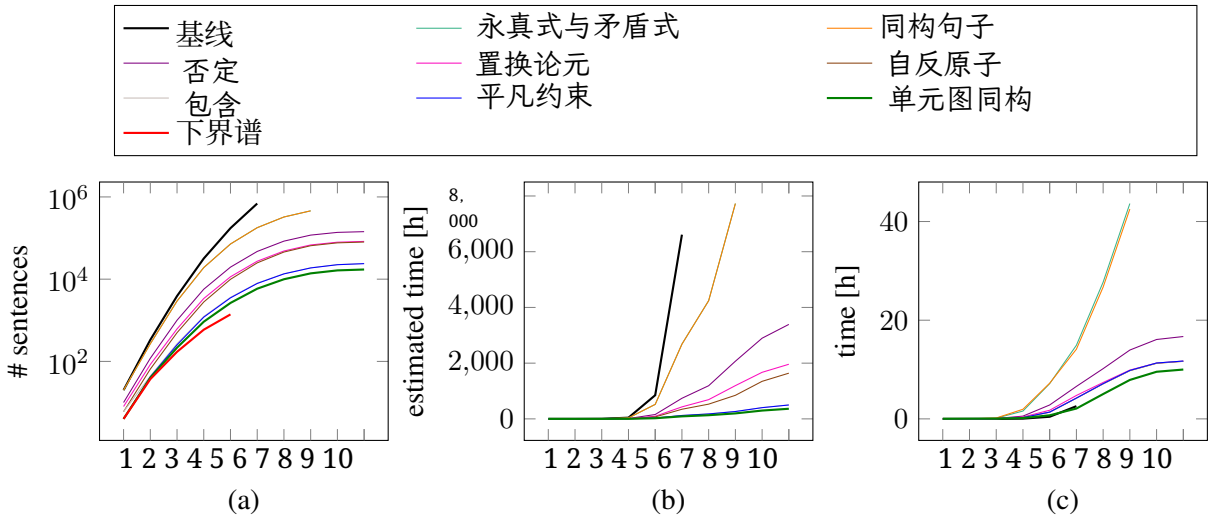

图1：累计 $\mathbf{FO}^2$ 句子数量 (a)、填充数据库的预计时间 (b)，以及生成句子所需的时间 (c)，最多包含 x 个文字。每个子句最多五个文字，每个句子最多两个子句，一个一元谓词和一个二元谓词。

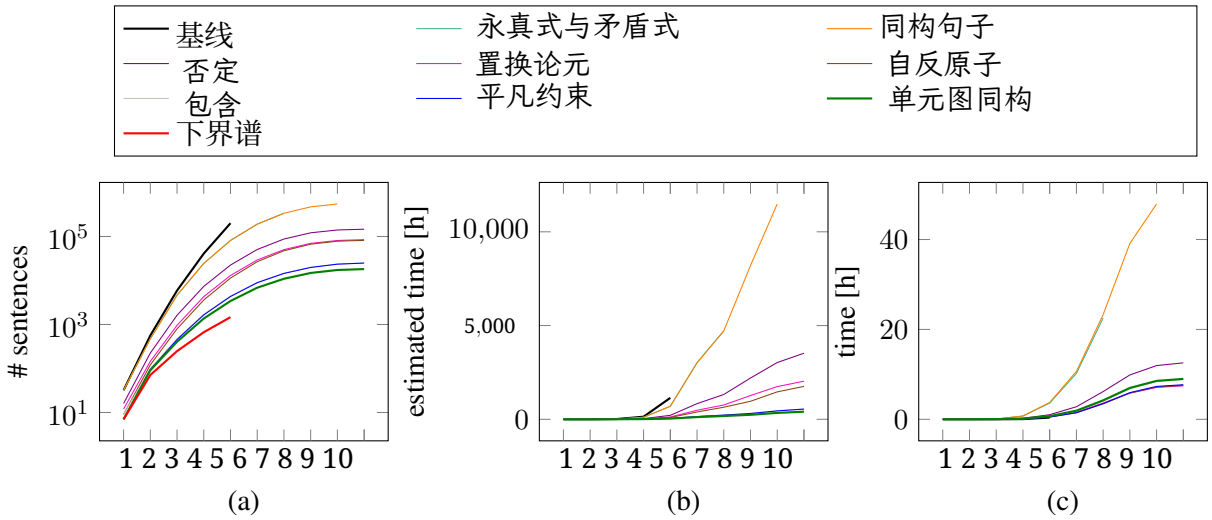

图2：$\mathbf{C}^2$ 句子的累计数量 (a)、填充数据库的预期时间 (b)，以及生成最多含有 x 个文字的句子所需的时间。每个子句最多含五个文字，每个句子最多含两个子句，一个一元谓词和一个二元谓词，$k \leq 1$。

表2：由我们的算法生成的句子的组合谱序列示例，其中一些也出现在 OEIS 中。

句子	OEIS 编号	OEIS 名称
$(\forall x \exists^{=1} y B(x,\ y)) \wedge (\forall x \exists^{=1} y B(y,\ x)) \wedge (\forall x \forall y B(x,x) \vee B(x,y) \vee \neg B(y,x))$	A85	n个自反排列的数量 字母，也称为对合；数量 个单元的标准杨表数量。
$(\forall x \exists^{=1} y\ B(x,\ y)) \wedge (\forall x \exists^{=1} y\ B(y,\ x))$	A142	阶乘数：$n! = 1 \cdot 2 \cdot 3 \cdot 4 \cdot ... \cdot n$ （对称群 S_n 的阶，数量 n 个字母的排列）。
$(\forall x B(x,\ x)) \wedge (\forall x \exists^{=1} y \neg B(x,\ y)) \wedge (\forall x \exists^{=1} y \neg B(y,\ x))$	A166	错排数，也称 rencontres 数或 derangements：n 的排列数 没有不动点的元素。
$(\forall x \forall y B(x,\ y) \vee \neg B(y,\ x)) \wedge (\exists x B(x,\ x)) \wedge (\forall x \exists^{=1} y \neg B(x,\ y))$	A1189	阶为 n 的排列数，阶 恰为 2。
$(\forall x \forall y\ U(x) \vee B(x,y)) \wedge (\forall x \forall y\ \neg U(x) \vee B(y,\ x))$	A47863	双色标记图的数量 其中黑色节点仅与 白色节点相连，反之亦然。
$(\forall x\ \ B(x,x)) \wedge (\forall x \exists y\ \ \neg B(x,\ y)) \wedge (\forall x \exists y\ \neg B(y,\ x))$	A86193	$n \times n$ 个元素取自.....的矩阵数量 $\{0,1\}$ 没有零行、没有零列 且主对角线为零。
$(\forall x \forall y\ \ U(x) \vee \neg U(y) \vee B(x,\ y)) \wedge (\forall x \exists^{=1} y\ \neg B(x,\ y))$	A290840	$a(n) = n! \cdot [x^n] \frac{\exp(n \cdot x)}{1 + LambertW(-x)}$.

生成它的：[15]$(\forall x \forall y\ U(x) \vee \neg U(y) \vee \neg B(x,\ y)) \wedge (\forall x \exists^{=1} y\ B(x,\ y))$。这可以解释如下：我们正在计算由一个函数 $b: [n] \to [n]$ 和一个集合 $U \subseteq [n]$构成的配置，且满足：如果 $y = b(x)$ 且 $y \in U$ 那么 $x \in U$。虽然这可能算不上一个深刻的组合问题，但它为当前序列提供了一种组合解释——如果没有这个数据库，我们就无法找到它。

接下来，我们讨论几个由我们的算法构造、可以说是自然的组合数列示例，这些数列尚未收录于 OEIS。第一个示例是数列 0, 0, 6, 72, 980, 15360, . . .，它由句子 $(\forall x \neg B(x,x)) \wedge (\exists x \forall y \neg B(y,x)) \wedge (\forall x \exists^{=1} y\ B(x,y))$ 生成。我们可以将其解释为统计函数 $f: [n] \to [n]$ 的数量，这些函数没有不动点，且像不等于 $[n]$。另一个示例是数列 1, 7, 237, 31613, 16224509, 31992952773, . . .，它对应于句子 $(\forall x \exists y\ B(x,y)) \wedge (\exists x \forall y\ B(x,y) \vee B(y,x))$，统计具有 n 个顶点的有向图的数量，其中每个顶点的出度都非零，且存在一个顶点，它通过出边或入边与所有其他顶点（包括自身）相连。还有一个示例是数列 1, 5, 127, 12209, 4329151, 5723266625, . . .，对应于句子 $(\forall x \exists y\ B(x,y)) \wedge (\exists x \forall y\ B(x,y))$。它统计每个顶点出度都非零、且至少有一个顶点的出度为 n 的有向图数量；这也等于没有零行且至少有一行全为 1 的二进制矩阵数量。这些示例对应数据库中较简单的结构；其中还有其他更复杂（也更难解释）的结构。例如，我们的算法还构造了另一个数列 0, 3, 43, 747, 22813, 1352761, . . .，它由句子 $(\forall x \neg B(x,x)) \wedge (\forall x \forall y \neg B(x,y) \vee B(y,x)) \wedge (\exists x \forall y \neg B(x,y) \vee \neg U(y)) \wedge (\exists x \exists y\ B(x,y))$ 给出，统计至少有一条边、顶点以两种颜色标记（红色和

[15]为了便于阅读，我们将谓词 B 替换为其否定，这不会改变谱。

黑色（红色对应于 $U(x)$，黑色对应于 $\neg U(x)$）的无环无向图数量，且其中至少有一个顶点未与任何红色顶点相连（注意，这个顶点本身也可以是红色）。我们还可以继续列举类似的数列，但我们认为这里展示的少数几个例子已足以说明我们的系统所构建的数据库中可能找到哪些类型的数列。

5 相关工作

据我们所知，此前尚无关于组合数列自动生成的研究。不过，有些研究在某些方面与本文工作有所交集。与本文最密切相关的是关于提升推理的研究（Poole，2003；Gogate 和 Domingos，2011；Van den Broeck，2011；Van den Broeck、Meert 和 Darwiche，2014；Beame 等人，2015；Kuˇzelka，2021）；没有提升推理，就不可能开展这项工作。我们直接使用了其中的算法（尽管经过了重新实现），以及（van Bremen 和 Kuˇzelka，2021）提出的单元图概念。同构句子的检测与（van Bremen 等人，2021）中介绍的技术类似，不过，该研究聚焦于命题逻辑问题，而我们在此将这些技术用于包含一阶逻辑句子的问题。此外，也有关于数学自动发现的研究，例如（Colton，2002；Davies 等人，2021），以及数据库 http://sequencedb.net；但据我们所知，枚举组合学领域尚无与我们的工作类似的研究。在组合学与人工智能交叉领域，最接近的一系列研究是（Suster 等人，2021）和（Totis 等人，2023）。然而，这些研究并未尝试生成新的数列或新的组合学成果，因为它们的主要目标是解决教科书式的组合问题——这同样是一个极具挑战性的问题。最后，近期还有一些研究将 OEIS 数列用作程序合成的输入，例如（D'Ascoli 等人，2022；Gauthier、Olˇs´ak 和 Urban，2023）。这类研究的目标与我们的目标正交。值得探究的是，如果将 OEIS 中没有收录的组合数列输入这些系统，能否得到有趣的合成程序。

6 结论

我们介绍了一种构建具有组合解释的整数序列数据库的方法，并用它生成了一个小型初始数据库，其中包含超过 2.6 万个互不重复的序列；其中相当一部分似乎已经被研究过，这表明我们能够自动生成有趣的整数序列。我们的方法包含两个关键组成部分：一种现有的提升推理算法（van Bremen 和 Kuˇzelka，2021），用于从一阶逻辑句子计算序列；以及一种新的生成一阶句子的方法，它能成功剔除数量庞大的冗余句子。

A 实现细节

我们使用 Java 实现了第 3.2 节所述的句子生成器，并采用了表1中的所有剪枝技术，依赖项如下：Sat4J [16], supertweety[17], 和 Prover9[18]。生成句子的算法伪代码见算法1；输出的句子 包含所有生成的

[16]https://www.sat4j.org/ [17] https://github.com/supertweety/LogicStuff [18] https://www.cs.unm.edu/~mccune/prover9/

算法1 $\mathbf{C}^2$ 句子生成器的伪代码

参数：文字上限为 ML，子句上限为 MC，一元和二元谓词的数量分别为 UP和 BP
输出：句子

```
句子 ← ∅
隐藏 ← ∅
层 ← {∅}
对 i ∈ [1, 2, 3, ..., ML × MC] 执行
    下一层 ← ∅
    对 句子 ∈ 层执行
        对 子项 ∈细化(句子, ML, MC, UP, BP) 执行
            如果 is-redundant(子项, 句子 ∪ hidden) 则
                如果 is-safe-to-delete(子项, 句子 ∪ hidden) 则
                    丢弃句子子项
                else
                    下一层 ← 下一层 ∪{子项}
                    隐藏 ←隐藏 ∪{子项}
                结束 if
            else
                下一层 ←下一层 ∪{子项}
                句子 ← 句子 ∪{子项}
            结束如果
        结束循环
    结束循环
    层 ← 下一层
结束循环
返回句子
```

$\mathbf{C}^2$ 句子，随后这些句子会被插入数据库，并计算其组合谱。

为避免在检查一个句子是否为重言式、矛盾式或两者皆非时陷入停滞，我们为 Prover9 的执行设置了一个较短的时间限制，即 30秒，并将其有效地用作软过滤器。一般而言，该检查是一个不可判定问题；对于 $\mathbf{C}^2$，它是 NEXPTIME 完全的（Pratt-Hartmann，2005年）。

加权一阶模型计数的计算，包括单元图构建和与 $\mathbf{C}^2$ 相关的归约，均使用 Julia编程语言实现（Bezanson等人，2017年）。我们使用了 Nemo.jl 软件包（Fieker等人，2017年）来进行多项式表示和运算。

A.1 OEIS 命中项

表3显示　　在第4.2节所述的初始数据库构建过程中发现的所有 OEIS 命中项　　.

B 进一步细节

本节讨论尚待解决的技术问题。

B.1 组合等价的复杂度

判断两个句子是否具有相同的组合谱，并不比检查它们是否等价更容易，原因如下。假设其中一个句子是矛盾式。检查另一个句子是否具有相同的组合谱，即，，.....，等价于检查它是否也是矛盾式。这只是一个复杂度下界，但已经表明，检查$\mathbf{C}^2$ 句子的组合谱是否等价是NEXPTIME 困难的；这一结论源自 Pratt-Hartmann（2005年）关于$\mathbf{C}^2$ 可满足性判定复杂度的经典结果。判断两个$\mathbf{C}^2$ 句子生成的组合谱是否相同，其确切复杂度仍是一个有趣的开放问题。

B.2 句子同构

这里，我们给出两个$\mathbf{C}^2$ 句子同构的形式化定义。我们只考虑第3.2节所述形式的句子：

$$\bigwedge_{\substack{Q_1 \in \{\forall, \exists, \exists^{=1}, \ldots, \exists^{=K}\}, \\ Q_2 \in \{\forall, \exists, \exists^{=1}, \ldots, \exists^{=K}\}}} \bigwedge_{i=1}^{M} Q_1 x Q_2 y \; \Phi_i^{Q1,Q2}(x,y) \wedge \bigwedge_{Q \in \{\forall, \exists, \exists^{=1}, \ldots, \exists^{=K}\}} \bigwedge_{i=1}^{M'} Q x \; \Phi_i^{Q}(x). \tag{3}$$

其中每个 $\Phi^{Q1}{}_{,i}{}^{Q2}(x,y)$ 和 $\Phi_i^Q(x)$ 都是文字的析取。

如果满足以下条件之一，我们就称子句 $Q_1 x Q_2 y \; \Phi(x,y)$ 与子句 $Q_1' x Q_2' y \; \Psi(x,y)$ 同构：

1. 如果 $Q_1 = Q_1' = Q_2 = Q_2'$ 并且存在一个双射 $f: \{x,y\} \to \{x,y\}$，使得 $f(\Phi(x,y))$ 的文字集合与 $\Psi(x,y)$ 的文字集合相同。
2. 如果 $Q_1 = Q_1' \neq Q_2 = Q_2'$，并且 $\Phi(x, \; y)$ 的文字集合与$\Psi(x, \; y)$ 的文字集合相同。

如果两个形式为 (2) 的句子中，一个句子的每个子句都能在另一个句子中找到一个与之同构的子句，我们就称这两个句子同构。

最后，我们将同构扩展为允许谓词重命名的同构：如果存在一个从两个形式为 (3) 的句子的谓词之间的双射，使得根据上述同构定义，这两个句子同构，那么我们就称这两个句子在谓词重命名下同构。

B.3 实验设置

本节包含复现第4节中实验所需的所有细节。具体而言，图1中可视化的 **一阶逻辑**^{2}实验采用了以下语言限制：每个子句最多包含五个文字，每个句子最多包含两个子句，且最多包含一个一元谓词和一个二元谓词。

这些限制同样适用于图2中展示的第二个实验，该实验涉及$\mathbf{C}^2$with $k \leq 1$，也就是说，仅限于 $\forall x \exists^{k=1} y$、 $\exists^{k=1} x$等类型的量词。我们还禁止由两个量词构成的元组，其中一个为 $\exists^{=k} x$ 形式，另一个则为 $\exists^{=l} y$ 或 $\exists y$，也就是说， $\exists x \exists^{=k} y$、 $\exists^{=k} x \exists y$、 $\exists^{=k} x \exists^{=l} y$。

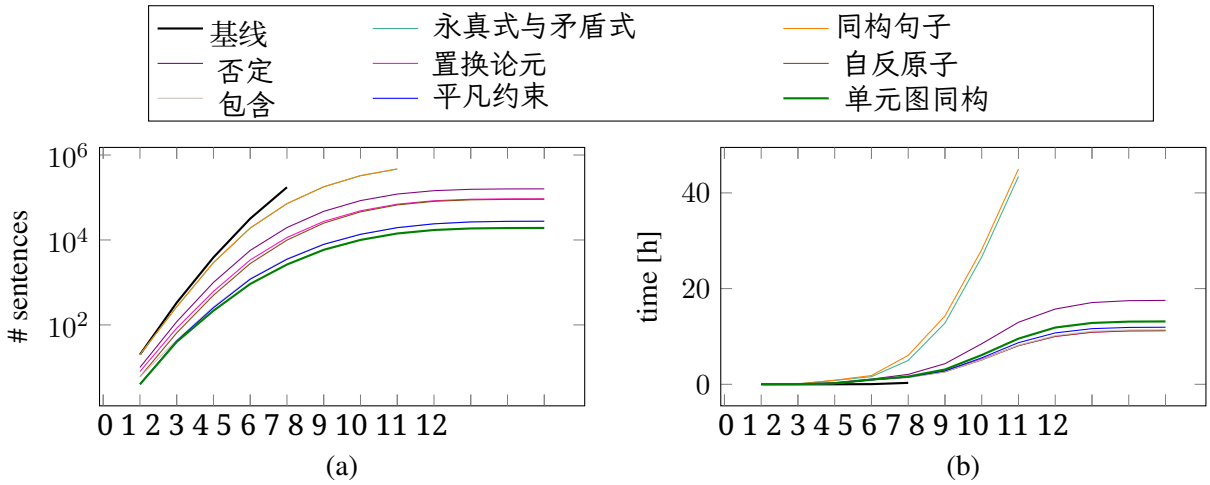

图 3：最多包含 x 个文字时，$\mathbf{FO}^2$ 句子的累计数量（a）以及生成句子所需的时间（b）。每个子句最多包含六个文字，每个句子最多包含两个子句、一个一元谓词和一个二元谓词。

这三种组合的扩展性都不理想，因此我们无法在用于填充数据库的五分钟时限内计算出它们的组合谱。出于同样的原因，我们将含有计数量词的子句中的文字数量限制为至多一个。

每个句子生成器都使用 51GB 内存运行，计算时间为 48 小时。曲线缺失的部分表示相应的生成器超出了其中一项限制。

C 进一步评估剪枝技术

我们又进行了两项实验，重点纯粹在于所提出的剪枝技术，而非整数序列数据库的构建。

第一项实验涉及生成每个子句最多含六个文字的 $\mathbf{FO}^2$ 和 $\mathbf{C}^2$ 句子；其余设置与附录 B.3 中相同。我们没有计算这些句子的组合谱，因此只在图 3 和图 4 中分别展示 $\mathbf{FO}^2$ 和 $\mathbf{C}^2$ 句子生成器所生成句子的数量及运行时间。

最后，我们采用了这样的设置：两个一元谓词和两个二元谓词，子句数最多为三个，每个子句最多含两个文字；图 5 和图 6 分别展示了 $\mathbf{FO}^2$ 和 $\mathbf{C}^2$ 的结果。由于分支因子高于此前的设置，这最后一种设置最为困难。

实验清楚地证实了所提出的剪枝技术的有效性。

参考文献

Barv´ınek, J.; van Bremen, T.; Wang, Y.; Zelezn´y, F.; and Kuzelka, O. 2021. 使用提升推理进行自动猜想，发现 p-递归。在 Katzouris, N., and Artikis, A., eds., 归纳逻辑编程——第 *30* 届国际会议，*ILP 2021*，论文集，《计算机科学讲义》第 13191 卷，17–25。Springer。

Beame, P.; Van den Broeck, G.; Gribkoff, E.; and Suciu, D. 2015. 对称加权一阶模型计数。载于第 *34* 届 *ACM SIGMOD-SIGACT-SIGAI* 数据库系统原理研讨会论文集，PODS '15，313–328。美国纽约州纽约市：计算机协会。

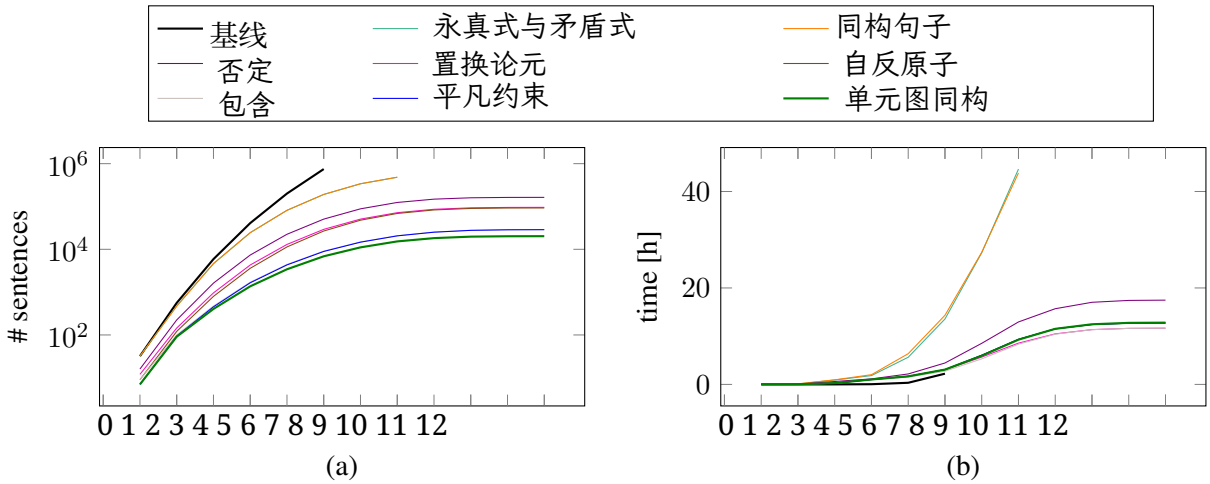

图 4：最多含 x 个文字时，$\mathbf{C}^2$ 句子的累计数量（a）以及生成句子所需的时间（b）。每个子句最多含六个文字，每个句子最多含两个子句，包含一个一元谓词和一个二元谓词，$k \leq 1$。

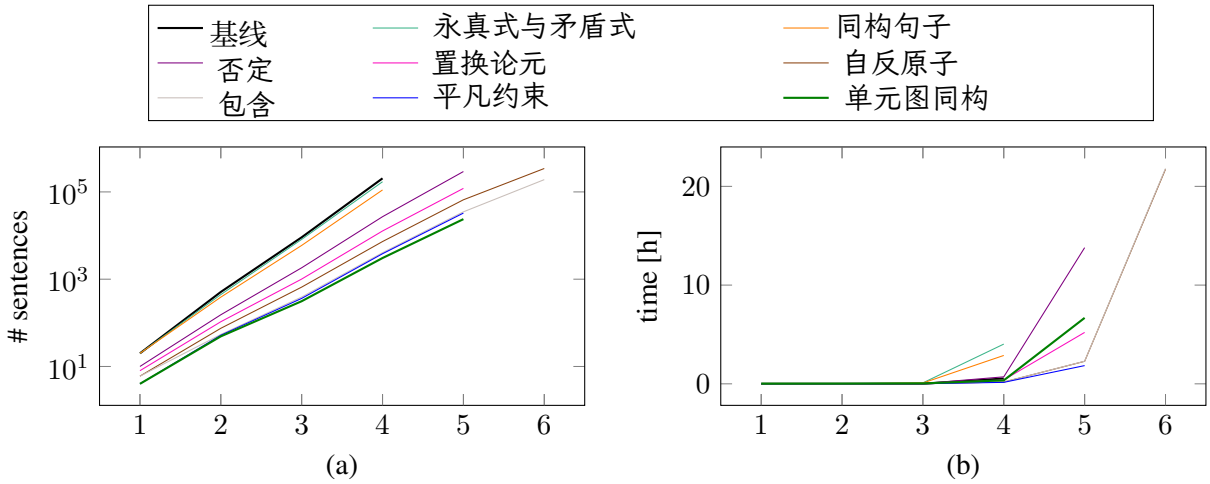

图 5：最多含 x 个文字时，$\mathbf{FO}^2$ 句子的累计数量 (a) 以及生成句子所需的时间 (b)。每个子句最多含两个文字，每个句子最多含三个子句，并包含两个一元谓词和两个二元谓词。

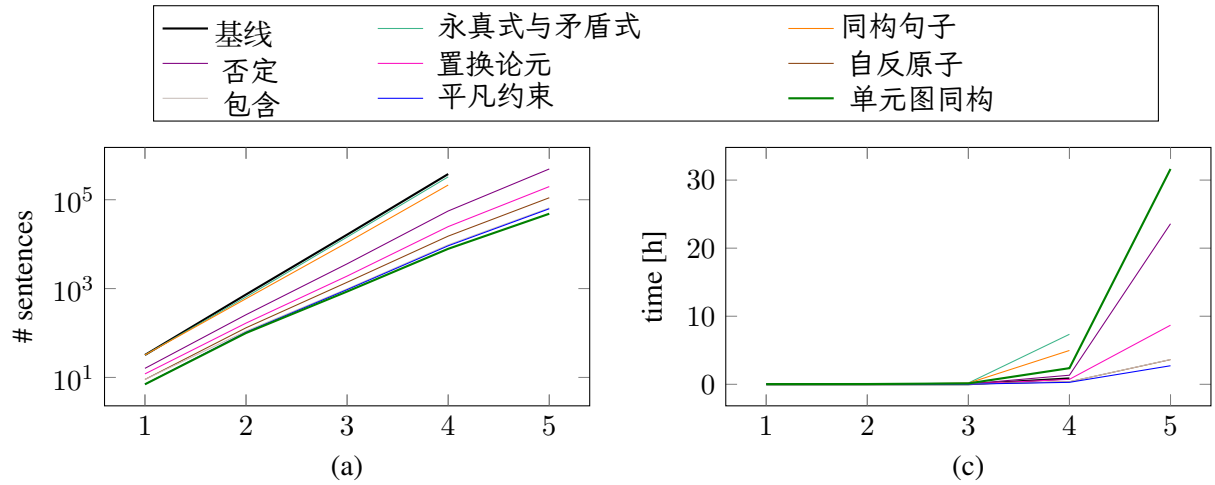

图 6：最多包含 x 个文字时，累计生成的 $\mathbf{C}^2$ 个句子数量（a）以及生成句子所需的时间（b）。每个子句最多包含两个文字，每个句子最多包含三个子句、两个一元谓词和两个二元谓词，$k \leq 1$。

Bezanson, J.; Edelman, A.; Karpinski, S.; 和 Shah, V. B. 2017. Julia：一种数值计算的新方法。*SIAM Review*59(1):65–98. B¨orger, E.; Gr¨adel, E.; 和 Gurevich, Y. 2001. 经典判定问题。Springer Science & Business Media. 48. Colton, S. 2002. 用于定理生成的 HR程序。载于国际自动演绎会议，285–289。Springer。D'Ascoli, S.; Kamienny, P.-A.; Lample, G.; 和 Charton, F. 2022. 用于递推关系预测的深度符号回归。载于 Chaudhuri, K.; Jegelka, S.; Song, L.; Szepesvari, C.; Niu, G.; 和 Sabato, S. 编，《第*39*届国际机器学习会议论文集》，第162卷《机器学习研究论文集》，4520–4536。PMLR。Davies, A.; Veli˘ckovi´c, P.; Buesing, L.; Blackwell, S.; Zheng, D.; Toma˘sev, N.; Tanburn, R.; Battaglia, P.; Blundell, C.; Juh´asz, A.; 等。2021. 通过引导人类直觉推动数学发展。*Nature* 600(7887):70–74。Fieker, C.; Hart, W.; Hofmann, T.; 和 Johansson, F. 2017. Nemo/Hecke：用于 Julia 编程语言的计算机代数与数论软件包。载于*2017*年*ACM*国际符号与代数计算研讨会论文集，ISSAC'17，157–164。美国纽约州纽约市：ACM。Gauthier, T.; Ol˘s´ak, M.; 和 Urban, J. 2023. Alien coding。Gogate, V., 和 Domingos, P. M. 2011. 概率定理证明。载于 Cozman, F. G., 和 Pfef- fer, A. 编，《*UAI 2011*，第*27*届人工智能不确定性会议论文集，西班牙巴塞罗那，*2011*年*7*月*14*日至*17*日》，256–265。AUAI Press。Gr¨adel, E., 和 Otto, M. 1999. 关于二变量逻辑。*Theoretical Computer Science* 224(1-2):73–113。

Hinrichs, T., 和 Genesereth, M. 2006. Herbrand逻辑。技术报告 LG-2006-02，斯坦福大学，斯坦福，加利福尼亚州。http://logic.stanford.edu/reports/LG-2006-02.pdf。Kuˇzelka, O. 2021. 带计数量词的二变量片段中的加权一阶模型计数。人工智能研究杂志70:1281–1307。Nijssen, S., 和 Kok, J. N. 2001. 多重关系上的快速关联规则。载于 Nebel, B. 编，《第十七届国际人工智能联合会议论文集，*IJCAI 2001*，美国华盛顿州西雅图，*2001*年*8*月*4*日至*10*日》，891–896。摩根·考夫曼。OEIS基金会。2023. 在线整数数列百科全书，电子版发布于 http://oeis.org。Pak, I. 2018. 枚举组合学中的复杂度问题。载于《国际数学家大会论文集：里约热内卢 *2018*》，3153–3180。世界科技出版公司。Poole, D. 2003. 一阶概率推理。载于 Gottlob, G., 和 Walsh, T. 编，《*IJCAI-03*，第十八届国际人工智能联合会议论文集，墨西哥阿卡普尔科，*2003*年*8*月*9*日至*15*日》，985–991。摩根·考夫曼。Pratt-Hartmann, I. 2005. 带计数量词的二变量片段的复杂度。逻辑、语言与信息杂志369–395。Pratt-Hartmann, I. 2009. 带计数量词的二变量片段的数据复杂度。*Infor-mation and Computation*207(8):867–888。St anley, R. P. 1986. 什么是枚举组合学？载于枚举组合学。Springer。1–63。Suster, S.; Fivez, P.; Totis, P.; Kimmig, A.; Davis, J.; De Raedt, L.; 和 Daelemans, W. 2021. 将概率文字题映射为可执行表示。载于*2021*年自然语言处理实证方法会议论文集，3627–3640。Totis, P.; Davis, J.; De Raedt, L.; 和 Kimmig, A. 2023. 面向组合计数的提升推理。人工智能研究杂志76:1–58。van Bremen, T., 和 Kuˇzelka, O. 2021. 利用单元图加速双变量逻辑的提升。载于第三十七届人工智能不确定性会议论文集，《机器学习研究论文集》第161卷，1393–1402。PMLR。van Bremen, T.; Derkinderen, V.; Sharma, S.; Roy, S.; 和 Meel, K. S. 2021. 用于组合实例模型计数的对称分量缓存。载于第三十五届*AAAI*人工智能会议，3922–3930。AAAI出版社。Van den Broeck, G.; Taghipour, N.; Meert, W.; Davis, J.; 和 De Raedt, L. 2011. 通过一阶知识编译实现提升概率推理。载于 Walsh, T. 编，《第二十二届国际人工智能联合会议（*IJCAI*）论文集》，2178–2185。AAAI出版社/国际人工智能联合会议。Van den Broeck, G.; Meert, W.; 和 Darwiche, A. 2014. 加权一阶模型计数的斯科伦化。载于《第十四届知识表示与推理原理国际会议论文集》，KR’14，111–120。AAAI出版社。

Van den Broeck, G. 2011年。《提升式概率推理的一阶知识编译的完备性》。载于第*24*届神经信息处理系统国际会议论文集，NIPS'11，1386–1394。美国纽约州雷德胡克：Curran Associates公司。

表3：更多 OEIS 匹配结果示例。

句子	OEIS 编号	OEIS 名称
$(\exists^{=1}x\ U(x))$	A27	正整数。也称为 nat- 自然数 ，非负整数或 计数数，但这些术语有歧 义。
$(\forall x\ U(x) \vee \neg U(x)) \wedge (\forall x\ \neg U(x) \vee U(x))$	A79	2 的幂： $a(n) = 2^n$。
$(\forall x \exists^{=1} y\ B(x,y)) \wedge (\forall x \exists^{=1} y\ B(y,x)) \wedge (\forall x \forall y\ B(x,\ y) \vee \neg B(y,\ x))$	A85	n 个字母上的自反排列数量 ，也称为对合；数量 由 n 个单元组成的标准杨表。
$(\forall x \exists y\ \neg B(y,x)) \wedge (\exists x \exists y\ \neg B(x,y)) \wedge (\forall x \exists^{=1} y\ \neg B(x,\ y))$	A142	阶乘数： $n! = 1*2*3*4*...*n$ （对称群 S_n 的阶， n 个字母的排列数）。
$(\forall x \exists^{=1} y\ B(x,y)) \wedge (\forall x \exists^{=1} y\ B(y,x)) \wedge (\forall x \forall y\ B(x,\ y) \vee B1(x,\ y))$	A165	双 阶乘 of even 数： $(2n)!! = 2^n * n!$。
$(\forall x \neg U(x) \vee \neg B(x,\ x)) \wedge (\forall x \exists^{=1} y\ \neg B(x,\ y))$	A169	含有 n 个节点的带标号根树数量 节点： n^{n-1}。
$(\forall x \exists^{=1} y\ B(x,y)) \wedge (\forall x \exists^{=1} y\ B(y,x)) \wedge (\exists^{=1} x\ B(x,\ x))$	A240	错排数：排列的数量 恰有一个不动点的 $[n]$。
$(\forall x\ U(x) \vee U1(x))$	A244	3 的幂： $a(n) = 3^n$。
$(\forall x \forall y \quad B(x,\ y) \vee \neg B(x,\ x)) \wedge (\forall x \exists^{=1} y\ \neg B(y,\ x))$	A248	指数型生成函数 $exp(x * exp(x))$ 的展开。
$(\forall x \exists^{=1} y\ B(x,y)) \wedge (\forall x \exists^{=1} y\ B(y,x)) \wedge (\forall x \forall y\ B(x,\ x) \vee \neg B(x,\ y) \vee \neg B(y,\ x))$	A266	指数型生成函数的展开 $\frac{exp(-\frac{x^2}{2})}{1-x}$.
$(\exists x \exists y B(x,\ y)) \wedge (\forall x \exists^{=1} y\ B(x,\ y))$	A312	$a(n) = n^n$；带标记映射的数量 从 n 个点映射到自身的映射（自函数- 函数）。
$(\forall x\ U(x) \vee U1(x)) \wedge (\forall x\ U(x) \vee U2(x))$	A351	5的幂： $a(n) = 5^n$。
$(\forall x \exists^{=1} y\ B\ (x,\ y)\ ) \wedge (\forall x \exists^{=1} y\ B\ (y,\ x)\ ) \wedge (\forall x \forall y\ \neg B\ (x,\ x)\ \vee U\ (x)\ \vee B\ (x,\ y)\ )$	A354	指数型生成函数 $\frac{exp\ (-x)}{1-2*x}$ 的展开式 。
$(\forall x \exists y\ U\ (x)\ \vee U1\ (y)\ ) \wedge (\forall x \forall y\ U1\ (x)\ \vee \neg U1\ (y)\ \vee U2\ (x)\ )$	A400	6 的幂： $a\ (n)\ = 6^n$。
$(\forall x\ U(x) \vee U1(x) \vee U2(x))$	A420	7 的幂： $a(n) = 7^n$。
$(\forall x \exists y\ B(x,y)) \wedge (\forall x\ B(x,x) \vee U(x)) \wedge (\forall x \exists^{=1} y\ B(y,\ x))$	A522	有序 k元组的总数（$k = 0..n$），从一个 n-中选取的不同元素 元素集： $a(n) = \sum_{k=0..n} \frac{n!}{k!}$。
$(\forall x\ U(x) \vee U1(x) \vee U2(x)) \wedge (\forall x\ U(x) \vee U3(x))$	A1020	11 的幂： $a(n) = 11^n$。
$(\forall x\ U(x) \vee U1(x) \vee U2(x) \vee U3(x))$	A1024	15 的幂。
$(\forall x \forall y \quad B(x,\ y) \vee \neg B(y,\ x)) \wedge (\exists x B(x,\ x)) \wedge (\forall x \exists^{=1} y\ \neg B(x,\ y))$	A1189	阶为 n 的排列数，阶 恰为 2。
$(\forall x \exists^{=1} y\ \neg B(y,\ x)) \wedge (\exists^{=1} x \forall y B(x,\ y))$	A1804	$a(n) = n! * C(n,\ 2)$。
$(\forall x \exists^{=1} y\ B(x,y)) \wedge (\forall x \exists^{=1} y\ B(y,x)) \wedge (\forall x\ \neg B(x,\ x) \vee U(x) \vee U1(x))$	A1907	指数型生成函数的展开 $\frac{exp\ (-x)}{(1-4*x)}$.

表3——接上一页

句子	OEIS 编号	OEIS 名称
$(\forall x\ U(x) \vee U1(x)) \wedge (\exists^{=1}x\ \neg U(x)) \wedge (\exists^{=1}x\ \neg U1(x))$	A2378	长方形数（也称 promic、pronic 或 heterome-cic）数： $a(n) = n*(n+1)$.
$(\forall x\ B(x,\ x) \vee \neg B(x,\ x))$	A2416	$a(n) = 2^{n^2}$。
$(\exists x\ \neg B(x,\ x)) \wedge (\exists x \forall y\ \neg B(y,\ x))$	A5019	具有 n X n (0,1)矩阵的数量，其中 1-宽度为 1。
$(\forall x\ U(x) \vee U1(x)) \wedge (\forall x \forall y\ \neg U(x) \vee U2(y))$	A5056	$a(n) = 3^n + 2^n - 1$。
$(\forall x \forall y\ U(x) \vee U1(y)) \wedge (\exists x\ U(x)) \wedge (\forall x \forall y\ U(x) \vee U2(y))$	A5367	$a(n) = 2*(2^n+1)*(2^{n+1}-1)$】【。
$(\forall x \forall y\ B(x,\ y) \vee \neg B(y,\ x))$	A6125	$a(n) = 2^{\frac{n*(n-1)}{2}}$.
$(\forall x \forall y\ B(x,\ y) \vee \neg B(y,\ x)) \wedge (\forall x \exists y\ B(x,\ y)) \wedge (\exists x B(x,\ x))$	A6129	$a(0)$、$a(1)$、$a(2)$、.....满足 $\sum_{k=0..n} a(k)*binomial(n,k) = 2^{binomial(n,2)}$，针对 $n \geq 0$.
$(\forall x\ \neg B(x,\ x)) \wedge (\forall x \forall y\ \neg B(x,\ y) \vee B(y,x)) \wedge (\forall x \forall y\ \neg B(x,y) \vee U(x))$	A6896	$a(n)$ 是层级线性序列的数量 定义在 n 个有标记因子上的模型，允许二元交互（但不允许更高阶交互）；或者节点从一个{v1}-集合中选取的简单标记从一个含有 n 个元素的集合中选取节点构成的图。
$(\forall x \forall y\ U(x) \vee B(x,y)) \wedge (\forall x \forall y\ B(x,y) \vee \neg B(y,\ x))$	A6898	$a(n) = \sum_{k=0..n} C(n,k)*2^{\frac{k*(k+1)}{2}}$.
$(\forall x \forall y\ B(x,\ x) \vee B(y,\ y)) \wedge (\forall x \exists^{=1}y\ \neg B(x,\ y))$	A7778	$a(n) = n^{n+1}$。
$(\forall x \exists^{=1}y\ B(x,y)) \wedge (\forall x \exists^{=1}y\ B(y,x)) \wedge (\forall x\ B(x,\ x) \vee U(x) \vee U1(x))$	A10845	$a(n) = 3*n*a(n-1)+1, a(0) = 1$。
$(\forall x \exists y\ U(x) \vee B(x,\ y)) \wedge (\forall x \exists^{=1}y\ B(y,x)) \wedge (\exists^{=1}x\ \neg U(x)) \wedge (\exists^{=1}x\ \neg B(x,\ x))$	A11379	$a(n) = n^2*(n+1)$。
$(\forall x\ U(x) \vee U1(x) \vee U2(x)) \wedge (\forall x\ \neg U(x) \vee U3(x))$	A11557	10的幂： $a(n) = 10^n$。
$(\forall x\ \neg B(x,\ x)) \wedge (\forall x \exists y\ B(x,\ y) \vee U(x) \vee U1(y)) \wedge (\forall x \forall y\ \neg B(x,y) \vee B(x,x))$	A20515	$a(n) = 4^n - 2^n + 1$。
$(\forall x \exists y\ U(x) \vee U1(y)) \wedge (\exists x\ U(x) \vee U2(x) \vee U3(x))$	A20518	第10圆分多项式在 2 的幂处。
$(\exists x\ U(x) \vee U1(x)) \wedge (\exists x\ \neg U(x) \vee U2(x))$	A20540	$a(n) = 8^{n+1} - 2^{n+2}$。
$(\forall x \forall y\ U(x) \vee U(y) \vee U1(x) \vee \neg U1(y))$	A27649	a(n) = $2*(3^n) - 2^n$。
$(\forall x \forall y\ U(x) \vee U1(y)) \wedge (\forall x \forall y\ \neg U(x) \vee U2(y))$	A33484	$a(n) = 3*2^n - 2$。
$(\forall x \exists^{=1}y\ B(x,y)) \wedge (\forall x \exists^{=1}y\ B(y,x)) \wedge (\forall x \forall y\ B(x,\ x) \vee U(x) \vee \neg U(y))$	A33540	$a(n+1) = n*(a(n)+1)$，适用于 $n \geq 1$, $a(1) = 1$。
$(\forall x \exists^{=1}y\ B(x,\ y)) \wedge (\forall x\ B(x,\ x) \vee U(x)) \wedge (\forall x\ \neg B(x,\ x)) \wedge (\exists^{=1}x \forall y\ \neg B(y,\ x))$	A37184	像中恰有 1 个节点的函数有向图。

表3——接上一页

句子	OEIS 编号	OEIS 名称
$(\exists x B(x,\ x)) \wedge (\exists x \exists y B(x,\ y)) \wedge (\forall x \exists^{=1} y\ B(y,\ x))$	A45531	黏性函数的数量：自映射函数从 $[n]$ 出发且具有不动点的函数。
$(\forall x \exists^{=1} y\ B(x,y)) \wedge (\forall x \exists^{=1} y\ B(y,x)) \wedge (\exists x \exists y B(x,\ y) \vee U(x) \vee U1(x))$	A47053	$a(n) = 4^n * n!$.
$(\forall x \forall y\ B(x,\ y) \vee B(y,\ x))$	A47656	$a(n) = 3^{\frac{n^2-n}{2}}$.
$(\forall x \forall y\ U(x) \vee B(x,y)) \wedge (\forall x \forall y \neg U(x) \vee B(y,\ x))$	A47863	双色节点标记图的数量，其中黑色节点仅与白色节点相连，反之亦然。相连，反之亦然。
$(\forall x \exists y \neg B(x,\ y)) \wedge (\forall x \exists y \neg B(y,\ x))$	A48291	没有, $1\ n$ X n 矩阵的数量，其中零行或零列的矩阵。
$(\forall x\ U(x) \vee U1(x) \vee U2(x)) \wedge (\forall x \forall y \neg U(x) \vee \neg U1(y))$	A48473	$a(0) = 1, a(n) = 3 * a(n-1) + 2;\ a(n) = 2 * 3^n - 1$。
$(\exists x \neg B(x,\ x)) \wedge (\forall x \exists^{=1} y\ B(x,\ y))$	A48861	$a(n) = n^n - 1$。
$(\forall x\ B(x,\ x)) \wedge (\forall x \exists y\ B(x,\ y))$	A53763	$a(n) = 2^{n^2-n}$。
$(\forall x \forall y\ (B(x,\ y) \vee B1(x,\ y)) \wedge (\exists x B(x,\ x)) \wedge (\forall x \neg B1(x,\ x))$	A53764	$a(n) = 3^{n^2-n}$。
$(\forall x \exists y \neg B(y,\ x))$	A55601	无 n X 个 n 二进制矩阵的数量，其中没有零行。
$(\forall x \exists^{=1} y\ B(x,\ y)) \wedge (\forall x\ B(x,\ x) \vee U(x)) \wedge (\exists x \neg U(x))$	A55869	$a(n) = (n+1)^n - n^n$。
$(\forall x \exists^{=1} y\ B(x,\ y)) \wedge (\exists^{=1} x\ B(x,\ x))$	A55897	$a(n) = n * (n-1)^{n-1}$。
$(\forall x\ U(x) \vee U1(x)) \wedge (\exists x\ U(x)) \wedge (\exists x\ U1(x))$	A58481	$a(n) = 3^n - 2$.
$(\forall x\ U(x) \vee U1(x)) \wedge (\exists x\ \neg U(x)) \wedge (\exists^{=1} x \neg U1(x))$	A58877	n 个顶点的标号无环有向图的数量节点且恰好包含 $n-1$ 个入度为零。
$(\forall x \forall y\ U(x) \vee U1(y)) \wedge (\exists x\ \neg U(x) \vee U2(x))$	A59153	$a(n) = 2^{n+2} * (2^{n+1} - 1)$。
$(\forall x\ \neg B(x,\ x) \vee \neg B1(x,\ x)) \wedge (\forall x\ B1(x,x) \vee U(x))$	A60757	$a\ (n)\ = \sqrt[2]{4^n}$。
$(\forall x \forall y\ B\ (x,\ y)\ \vee\ B1\ (x,\ y)\ ) \wedge (\forall x \forall y\ B\ (x,\ y)\ \vee B2\ (x,\ y)\ )$	A60758	$a(n) = 5^{n^2}$。
$(\forall x \exists y\ U(x) \vee U1(y)) \wedge (\exists x \neg U(x))$	A60867	$a(n) = (2^n - 1)^2$。
$(\forall x \exists y\ B(x,\ y)) \wedge (\exists x \forall y\ \neg B(x,\ y)) \wedge (\forall x \exists^{=1} y \neg B(y,\ x))$	A61190	$a(n) = n^n - n$。
$(\forall x \exists^{=1} y\ B(x,y)) \wedge (\forall x \exists^{=1} y\ B(y,x)) \wedge (\forall x \neg B(x,\ x) \vee U(x)) \wedge (\exists^{=1} x \neg U(x))$	A62119	$a\ (n)\ = n!\ *\ (n-1)\ .$
$(\exists x \exists y U(x)\ \vee\ B(x,\ y)) \wedge (\forall x \exists^{=1} y \neg B(x,\ y))$	A62971	$a(n) = (2 * n)^n$。
$(\forall x \exists y\ B(x,\ y)) \wedge (\exists x \forall y\ \neg B(y,\ x) \vee \neg B(x,\ y)) \wedge (\forall x \exists^{=1} y\ B(y,\ x))$	A66052	对称群中的排列数量群 S_n 的阶为 ≥ 3。
$(\forall x \forall y\ U(x) \vee B(x,y)) \wedge (\forall x \exists y \neg U(x) \vee B(y,\ x)) \wedge (\forall x \exists^{=1} y\ B(y,\ x))$	A66068	$a(n) = n^n + n$。

表3 – 承接上一页

句子	OEIS 编号	OEIS 名称
$(\forall x \forall y \ U(x) \vee U1(y)) \wedge (\exists x \ U(x)) \wedge (\forall x \forall y \neg U1(x) \vee U2(y))$	A68156	生成函数：$\frac{(x+2)*(x+1)\,(x-1)}{*(x-2)} = \sum_{n\geq 0} a(n) * \frac{x^n}{2}$。$a(n) = \sum_{k=0..n} binomial(n,k) * k^n$。
$(\forall x \forall y \ U(x) \vee B(x, y)) \wedge (\exists x \exists y \neg B(x, y)) \wedge (\forall x \exists^{=1} y \neg B(y, x))$	A72034	
$(\forall x \neg B(x, x)) \wedge (\forall x \forall y \neg B(x, y) \vee B(y, x)) \wedge (\forall x \forall y \neg B(x, y) \vee U(x) \vee U(y))$	A79491	分子 $\sum_{k=0..n} \frac{binomial(n,k)}{2^{\frac{k*(k-1)}{2}}}$
$(\forall x \forall y \ U(x) \vee \neg U(y) \vee U1(x) \vee U2(y))$	A81626	$2 * 6^n - 4^n$。
$(\forall x \forall y \ B(x, y) \vee B1(x, y)) \wedge (\forall x \forall y \ B(x, y) \vee \neg B1(y, x))$	A81955	$a(n) = 2^r * 3^s$ 其中 $r = \frac{n\,(n+1)}{2}$ 和 $s = \frac{n(n-1)}{2}$。
$(\forall x \forall y \ U(x) \vee U1(x) \vee U2(y)) \wedge (\forall x \forall y \ U(x) \vee \neg U2(y))$	A83319	$4^n + 3^n - 2^n$。
$(\forall x \forall y \ U(x) \vee U1(x) \vee U2(y)) \wedge (\forall x \ U(x) \vee U2(x))$	A83320	$a(n) = 5^n + 4^n - 3^n$。
$(\forall x \ U(x) \vee U1(x)) \wedge (\forall x \exists y \ U(x) \vee \neg U1(y)) \wedge (\exists x \ U(x))$	A83323	$a(n) = 3^n - 2^n + 1$。
$(\forall x \ \neg B(x, x)) \wedge (\forall x \forall y \ \neg B(x, y) \vee B(y, x)) \wedge (\forall x \forall y \neg B(x, y) \vee B1(x, y)) \wedge (\forall x \forall y \ B1(x, y) \vee \neg B1(y, x))$	A83667	反对称二元关系的数量 在由 n 个带标签的点组成的集合上。
$(\forall x \forall y \ U(x) \vee U1(y) \vee U2(y)) \wedge (\forall x \ U(x) \vee \neg U1(x))$	A85350	二项式变换 多重伯努利数 A027649。
$(\forall x \forall y \ U(x) \vee U(y) \vee U1(x) \vee U2(y))$	A85352	展开式 $\frac{1-4x}{(1-5x)(1\ -6x)}$.
$(\forall x \exists^{=1} y \ B(x, y)) \wedge (\forall x \ B(x, x) \vee U(x)) \wedge (\forall x \ U(x) \vee U1(x))$	A85527	$a(n) = (2n+1)^n$。
$(\forall x \ \neg U(x) \vee B(x, x)) \wedge (\forall x \exists^{=1} y \neg B(x, y))$	A85528	a(n) = $(2 * n + 1)^{n+1}$。
$(\forall x \forall y \ B(x, y) \vee B1(x, y)) \wedge (\exists x B(x, x)) \wedge (\forall x \ \neg B1(x, x)) \wedge (\forall x \exists^{=1} y \ B1(x, y))$	A85532	$(2n)^{n+1}$。
$(\forall x \ B(x, x)) \wedge (\forall x \exists y \ \neg B(x, y)) \wedge (\forall x \exists y \neg B(y, x))$	A86193	元素属于以下集合的 n X n 矩阵数量 0，1，且没有零行、没有零列，并且主对角线为零的矩阵。
$(\forall x \neg B(x, x)) \wedge (\forall x \exists y \ B(x, y))$	A86206	由 n 个元素为 n 的矩阵组成的数量 $0, 1$，且没有零行，主对角线为零。
$(\forall x \exists^{=1} y \ B(x, y)) \wedge (\forall x \exists^{=1} y \ B(y, x)) \wedge (\forall x \forall y \neg B(x, x) \vee U(y)) \wedge (\exists^{=1} x \ U(x))$	A86325	设 $u(1) = 0, u(2) = 1, u(k) = u(k-1) + {}^{u}{}_{(k-2)\ k-2}$；则 $a(n) = n! * u(\{v12)$。
$(\forall x \forall y \ U(x) \vee B(y, y)) \wedge (\exists x \exists y U(x) \vee U1(x) \vee B(x, y))$	A88668	GF(2) 上满足.....条件的 n×n 矩阵的数量 特征多项式 $x^{n-1} * (x-1)$。
$(\forall x \ U(x) \vee \neg B(x, x)) \wedge (\exists^{=1} x \neg U(x)) \wedge (\forall x \exists^{=1} y \ B(x, y))$	A89205	$a(n) = n^n * (n-1)$。
$(\forall x \exists y \ U(x) \vee U1(y)) \wedge (\forall x \forall y \ U1(x) \vee U2(y))$	A92440	$a(n) = 2^{(n+1)} - 2^{n+1} + 1$。

表3——续前页

句子	OEIS 编号	OEIS 名称
$(\forall x \forall y\ U(x) \vee \neg U(y)) \wedge (\forall x \forall y\ U(x) \vee U1(x) \vee U2(y))$	A93069	$a(n) = (2^n+1)^2 - 2$.
$(\forall x \forall y\ B(x, y) \vee \neg B(y, x)) \wedge (\forall x \forall y\ B(x, x) \vee U(y)) \wedge (\exists^{=1} x\ \neg B(x, x))$	A95340	所有标记 图中节点的总数为 n 个。
$(\forall x \forall y\ U(x) \vee B(x, y)) \wedge (\forall x \exists^{=1} y\ \neg B(y, x)) \wedge (\exists^{=1} x \forall y B(x, y))$	A98916	n X n (0,1)矩阵的积和式 其第 ij 项等于零，当且仅当（$i = 1$，$j = 1$)、($i = 1$，$j = n$)、($i = n$，$j = 1$) 以及（$i = n$，$j = n$）。
$(\forall x \forall y\ B(x, y) \vee \neg B(y, x)) \wedge (\forall x \forall y\ B(x, y) \vee B1(x, y))$	A99168	a（n）$= 3^n * 5^{binomial(n,2)}$。
$(\forall x \forall y\ U(x) \vee U1(x) \vee U2(y)) \wedge (\forall x \forall y\ \neg U(x) \vee U2(y))$	A99393	$a(n) = 4^n + 2^n - 1$。
$(\forall x\ U(x) \vee U1(x)) \wedge (\forall x \exists y\ \neg U(x) \vee \neg U1(y)) \wedge (\exists x\ \neg U(x))$	A101052	n 的优先排列数量 带标签元素仅允许 $k \leq 3$ 个等级 。
$(\forall x \forall y\ B(x, y) \vee B1(x, y)) \wedge (\forall x \forall y\ \neg B(x, y) \vee \neg B(y, x))$	A109345	$a(n) = 5^{\frac{n^2-n}{2}}$.
$(\forall x\ \neg B(x, x)) \wedge (\forall x \forall y\ \neg B(x, y) \vee B1(x, y)) \wedge (\forall x \forall y\ B1(x, y) \vee B(y, x))$	A109354	$a(n) = 6^{\frac{n^2-n}{2}}$.
$(\forall x\ \neg B(x, x)) \wedge (\forall x \forall y\ \neg B(x, y) \vee B(y, x) \vee B1(x, y)) \wedge (\forall x \forall y\ B1(x, y) \vee B(x, y))$	A109493	$a(n) = 7^{\frac{n^2-n}{2}}$.
$(\forall x\ \neg B(x, x)) \wedge (\forall x \forall y\ \neg B(x, y) \vee B1(x, y)) \wedge (\forall x \forall y\ B1(x, y) \vee B1(y, x))$	A109966	$a(n) = 8^{\frac{n^2-n}{2}}$.
$(\forall x \exists^{=1} y\ B(x, y)) \wedge (\forall x \forall y\ B(x, x) \vee \neg B(y, y))$	A110567	$a(n) = n^{n+1} + 1$。
$(\forall x \forall y\ B(x, y) \vee \neg B(y, x)) \wedge (\exists x B(x, x))$	A122743	标准化多项式的数量，次数为 n，在 $GF(2)[x, y]$ 中。
$(\forall x \forall y\ B(x, y) \vee \neg B(y, x)) \wedge (\exists x \exists y B(x, y))$	A126883	$a(n) = (2^0) * (2^1) * (2^2) * (2^3) ... (2^n) - 1 = 2^T(n) - 1$ 其中 $T(n) = A000217(n)$ 是第 n个三角数。
$(\forall x\ \neg B(x, x)) \wedge (\forall x \forall y\ \neg B(x, y) \vee B(y, x)) \wedge (\forall x \forall y\ \neg B(x, y) \vee U(x)) \wedge (\forall x \forall y\ U(x) \vee \neg U(y))$	A126884	$a(n) = (2^0) * (2^1) * (2^2) * (2^3) ... (2^n) + 1 = 2_n^T + 1$（参见 A000217）。
$(\forall x \exists^{=1} y\ B(x, y)) \wedge (\forall x \exists^{=1} y\ B(y, x)) \wedge (\forall x \forall y\ B(x, x) \vee U(y))$	A127986	$a(n) = n!\ + 2^n - 1$。
$(\exists^{=1} x\ \neg B(x, x))$	A128406	$a(n) = (n+1) * 2^{n*(n+1)}$。
$(\forall x \exists y\ U(x) \vee U1(y)) \wedge (\forall x \exists y\ \neg U1(x) \vee U2(y)) \wedge (\exists x\ \neg U2(x))$	A128831	每个元素均为 n 元组的数量 从 1、2、3 的子集中选取，使得 所有 n 元素的交集为空集。

表3 – 接上一页

句子	OEIS 编号	OEIS 名称
$(\forall x \forall y \quad B(x, \ y) \ \vee \ \neg B(y, \ x)) \ \wedge$ $(\forall x \ B(x, x) \ \vee U(x))$	A132727	$a(n) = 3 * 2^{n-1} * a(n-1)$，当 $a(0)$ 时 $= 1$。
$(\forall x \forall y \quad U(x) \ \vee \ U1(x) \ \vee \ B(x, y)) \ \wedge$ $(\forall x \ U(x) \ \vee U1(x))$	A133460	$3^n * 2^{n^2}$
$(\forall x \exists y \quad B(x, \ y)) \wedge \ (\exists x B(x, \ x)) \wedge$ $(\exists x \ \neg B(x, \ x)) \wedge (\forall x \exists^{=1} y \ B(y, \ x))$	A133798	$a(n) = A002467(n) - 1$。
$(\forall x \forall y \quad U(x) \ \vee \ U(y) \ \vee \ B(x, y)) \ \wedge$ $(\forall x \ B(x, \ x))$	A134485	三角形 $A134484(n, \ k) =$ 的行和 $2^{[n(n-1)-k(k-1)]} * C(n, \ k)$。
$(\forall x \forall y \quad U(x) \ \vee \ U1(x) \ \vee \ \neg U1(y)) \ \wedge$ $(\forall x \ U(x) \ \vee U2(x))$	A135160	$a(n) = 5^n + 3^n - 2^n$。
$(\forall x \forall y \ U(x) \ \vee B(x, y)) \wedge (\forall x \forall y \ U(x) \ \vee$ $B(y, \ x))$	A135748	$a\ (n) = \sum_{k=0..n} binomial\ (n, \ k) \ * 2^{k^2}$。
$(\forall x \forall y \quad U(x) \ \vee B(x, \ y) \ \vee B(y, \ x)) \ \wedge$ $(\forall x \forall y \ \neg U\ (x) \ \vee \neg B\ (x, \ y)\ )$	A135755	$a\ (n) = \sum_{k=0..n} C\ (n, \ k) \ * 3^{\lfloor \frac{k*(k-1)}{2} \rfloor}$。
$(\forall x \forall y \quad B(x, \ y) \ \vee \ \neg B(x, \ x)) \ \wedge$ $(\forall x \forall y \ \neg B(x, \ x) \ \vee B(y, \ x))$	A135756	$a\ (n) = \sum_{k=0..n} C\ (n, k) \ * 2^{k*(k-1)}$。
$(\forall x \forall y \ \neg U(x) \ \vee \neg B(y, x))$	A136516	$a(n) = (2^n + 1)^n$。
$(\forall x \exists y \quad U(x) \ \vee U1(y)) \ \wedge (\exists x \quad U(x) \ \vee$ $\neg U1(x) \ \vee U2(x))$	A145641	其二进制表示为 n 1、 n 0和 n 1的串接。
$(\forall x \forall y \quad U(x) \ \vee \ \neg U(y) \ \vee \ U1(x)) \ \wedge$ $(\forall x \ U(x) \ \vee U2(x))$	A155588	$a\ (n) = 5^n + 2^n - 1^n$。
$(\forall x \ U\ (x) \ \vee U1\ (x)\ ) \ \wedge \ (\forall x \exists y \ \neg U\ (x) \ \vee$ $\neg U1\ (y) \ \vee U2\ (y)\ )$	A155597	$a\ (n) = 6^n - 2^n + 1$。
$(\forall x \exists y \ U(x) \ \vee U1(y) \ \vee U2(y))$	A155599	$a(n) = 8^n - 2^n + 1^n$。
$(\forall x \forall y \quad U(x) \ \vee \ \neg U(y) \ \vee \ U1(x)) \ \wedge$ $(\forall x \ \neg U1(x) \ \vee U2(x))$	A155602	$4^n + 3^n - 1$。
$(\forall x \quad U(x) \ \vee \ U1(x)) \ \wedge (\forall x \exists y \quad U(x) \ \vee$ $U2(y)) \ \wedge (\exists x \ U(x))$	A155611	$6^n - 3^n + 1$。
$(\forall x \quad U(x) \ \vee \ U1(x) \ \vee \ U2(x)) \ \wedge$ $(\exists x \forall y U(x) \ \vee \neg U1(y))$	A155612	$7^n - 3^n + 1$。
$(\forall x \exists y \quad U\ (x) \ \vee U1\ (y)\ ) \ \wedge \ (\exists x \quad U\ (x) \ \vee$ $U2\ (x)\ )$	A155629	$a(n) = 8^n - 4^n + 1^n$。
$(\forall x \exists^{=1} y \ B(x, y)) \wedge (\forall x \exists^{=1} y \ B(y, x)) \ \wedge$ $(\forall x \forall y \quad B(x, \ y) \ \vee \ B1(x, \ y)) \ \wedge$ $(\exists^{=1} x \ \neg B1(x, \ x))$	A161937	间接等距变换的数量，这些变换是 $(n-1)$ 维立方体的面上的错排。
$(\forall x \forall y \ B(x, \ x) \ \vee \neg B(y, \ x))$	A165327	指数生成函数: $\sum_{n>0} 2^{n(n-1)} * exp(2^n * x) * \frac{x^n}{n!}$。
$(\exists x \quad U(x) \ \vee \ U1(x)) \ \wedge (\exists x \quad \neg U(x)) \ \wedge$ $(\exists x \ U(x) \ \vee \neg U1(x))$	A170940	$4^n - 2^n - 2$。
$(\forall x \forall y \quad U\ (x) \ \vee U1\ (y)\ ) \ \wedge \ (\exists x \quad U\ (x)\ ) \ \wedge$ $(\forall x \ \neg U1\ (x) \ \vee U2\ (x)\ )$	A171270	a(n) 是唯一满足 $m =$ 的数 m $pi\ (1^{\frac{1}{n}}) \ + pi\ (2^{\frac{1}{n}}) \ + ... + pi\ (m^{\frac{1}{n}})$。
$(\forall x \exists y \quad \neg B(x, \ y) \ \vee \ \neg B(y, \ x)) \ \wedge$ $(\exists x \exists y \ \neg B(x, \ y))$	A173403	A002416 的逆二项变换。

表3 – 接上一页

句子	OEIS 编号	OEIS 名称
$(\exists x \forall y \quad B(x,\ y) \ \vee \ \neg B(y,\ y)) \ \wedge$ $(\exists x \exists y \ \neg B(x,\ y)) \ \wedge (\forall x \exists^{=1} y \ B(y,\ x))$	A176043	$a(n) \ = (2 * n - 1) \ * (n-1)^{n-1}$。
$(\forall x \exists y \ \neg U(x) \ \vee B(x,\ y))$	A180602	$(2^{n+1} - 1)^n$。
$(\forall x \exists y \ U\ (x) \ \vee U1\ (y)\ ) \ \wedge \ (\forall x \exists y \ U1\ (x) \ \vee$ $U(y))$	A191341	$a(n) \ = 4^n - 2 * 2^n + 3$。
$(\forall x \forall y \ U(x) \vee B(x,y)) \wedge (\forall x \forall y \ B(x,y) \vee$ $U1(y))$	A196460	指数型生成函数：$A(x) = \sum_{n \geq 0} (1+2^n)^n * exp((1+2^n) * x) * \frac{x^n}{n!}$.
$(\forall x \forall y \quad U(x) \ \vee \ B(x,\ y)) \ \wedge$ $(\forall x \forall y \ \neg B(x,\ y) \ \vee B1(x,\ y))$	A202989	指数生成函数：$\sum_{n \geq 0} 3^{\binom{n}{2}} * exp(3^n * x) * \frac{x^n}{n!}$
$(\forall x \ U\ (x) \ \vee U1\ (x)\ ) \ \wedge \ (\forall x \exists y \ \neg U\ (x) \ \vee$ $B(x,\ y))$	A202990	指数生成函数：$\sum_{n \geq 0} 3^n * 2^{n^2} * exp(-2 * 2^n * x) * \frac{x^n}{n!}$.
$(\forall x \forall y \quad B(x,\ y) \ \vee \ B1(x,\ y)) \ \wedge$ $(\forall x \exists y \ \ B(x,\ y)) \ \wedge \ (\forall x \exists y \ \ \neg B(x,\ y) \ \vee$ $\neg B1(x,\ y))$	A202991	指数生成函数：$\sum_{n \geq 0} 3^n * exp\ (-2 * 3^n * x) \ * \frac{x^n}{n!}$.
$(\forall x \forall y \quad B(x,\ y) \ \vee \ B(y,\ x)) \ \wedge$ $(\exists x \exists y \ \neg B(x,\ y))$	A206601	$3^{\frac{n(n+1)}{2}} - 1$.
$(\forall x \exists^{=1} y \ B(x,y)) \ \wedge (\forall x \exists^{=1} y \ B(y,x)) \ \wedge$ $(\forall x \forall y \ \neg B(x,\ x) \ \vee B(x,\ y) \ \vee \neg B(y,\ y))$	A212291	n 元素的排列数 至多有一个不动点。
$(\forall x \ U\ (x) \ \vee U1\ (x)\ ) \ \wedge \ (\forall x \exists y \ \neg U\ (x) \ \vee$ $U2(y)) \ \wedge (\exists x \ \neg U2(x))$	A212850	n X 3 数组的数量，其行被- 排列 0..2，且所有行中均不存在列 j 大于所有行中的列 j-1。
$(\forall x \forall y \quad B(x,\ y) \ \vee \ \neg B(y,\ x)) \ \wedge$ $(\forall x \forall y \ B(x,\ x) \ \vee \neg B(y,\ y))$	A217994	$a(n) \ = 2^{\frac{2+n+n^2}{2}}$.
$(\forall x \quad \neg B(x,\ x)) \ \wedge \ (\forall x \forall y \quad \neg B(x,\ y) \ \vee$ $B(y,\ x)) \ \wedge \ (\forall x \forall y \quad \neg B(x,\ y) \ \vee U(x) \ \vee$ $\neg U(y))$	A226773	选择一个简单标号图的方式数 在 n 个节点上，然后选择一个子集 其连通分支。
$(\forall x \forall y \quad B(x,\ y) \ \vee \ B1(x,\ y)) \ \wedge$ $(\forall x \exists y \quad B(x,\ y)) \ \wedge \ (\forall x \exists y \quad B(y,\ x)) \ \wedge$ $(\exists x \exists y B(x,\ y))$	A230879	2-packed n X n 矩阵的数量。
$(\forall x \exists y \ B(y,\ x) \ \vee B1(x,\ y))$	A241098	$(4^n - 1)^n$。
$(\forall x \ \neg B(x,x)) \wedge (\forall x \forall y \ \neg B(x,y) \vee U(x) \vee$ $U1(x)) \ \wedge (\forall x \forall y \ U(x) \ \vee \neg B(y,x))$	A243918	$a(n) \ = \sum_{k=0..n} binomial(n,k) \ * (1 + 2^k)^k$。
$(\exists x \forall y \quad B(x,\ y) \ \vee \ \neg B(y,\ x)) \ \wedge$ $(\forall x \ B(x,\ x)) \ \wedge (\forall x \exists^{=1} y \ \neg B(x,\ y))$	A246189	$[n]$ 上最小环长为 2 的自映射数量。
$(\forall x \forall y \ U(x) \ \vee B(x,y)) \ \wedge (\forall x \exists y \ U(x) \ \vee$ $\neg B(y,\ x))$	A251183	$a(n) \ = \ \sum_{k=0..n} binomial(n,\ k) \ * (-1)^{n-k} * (2^k + 1)^k$。
$(\forall x \forall y \ U(x) \vee B(x,y)) \wedge (\forall x \forall y \ B(x,y) \vee$ $U1(x))$	A251657	$a(n) \ = (2^n + 3)^n$。
$(\forall x \forall y \ \ U(x) \ \vee \ U1(y)) \ \wedge (\exists x \ \ U(x) \ \vee$ $U2(x)) \ \wedge (\exists x \ U1(x))$	A267816	第 n 次迭代的十进制表示—— 从一个亮起的“规则 221”元胞自动机开 始，该自动机初始时只有一个亮起的 （黑色）单元。

表3——续前页

句子	OEIS 编号	OEIS 名称
$(\forall x \exists y \quad B(x,\ y) \ \vee \ B1(x,\ y)) \wedge (\forall x \ B(x,\ x)) \wedge (\forall x \exists^{=1} y \ \neg B(x,\ y)) \wedge (\forall x \exists^{=1} y \ \neg B1(x,\ y))$	A281997	$a(n) = (n-1)^n * n^n$。
$(\forall x \exists y \ B(x,\ y)) \wedge (\forall x \exists y \ \neg B(x,\ y)) \wedge (\forall x \exists y \ B(y,\ x)) \wedge (\forall x \exists y \ \neg B(y,\ x))$	A283624	不含全为 0 的 $1 \ n$ X n 矩阵的数量行或列，其中所有元素均为相同 .
$(\forall x \exists y \ B(x,\ y)) \wedge (\exists x \forall y \ \neg B(y,\ x) \vee U(y)) \wedge (\forall x \forall y \ U(x) \vee \neg U(y))$	A287065	$n \times n$ 的支配集数量车图。
$(\exists x \forall y \quad B\ (x,\ y) \ \vee \ U\ (y)\ ) \wedge (\forall x \exists^{=1} y \ B\ (y,\ x)\ ) \wedge (\exists^{=1} x \ \neg B\ (x,\ x)\ )$	A317637	$a(n) = n * (n+1) * (n+3)$】【。
$(\forall x \forall y \quad B(x,\ y) \ \vee \ \neg B(y,\ x)) \wedge (\forall x \exists y \ B(x,\ y))$	A322661	跨越 n 个顶点的带自环图数量带标签顶点。
$(\forall x \forall y \ U(x) \vee B(x,y)) \wedge (\forall x \exists y \ \neg U(x) \vee \neg B(y,\ x))$	A324306	生成函数：$\sum_{n \geq 0} \frac{(2^n+1)^n * x^n}{(1+2^n * x)^{n+1}}$。
$(\forall x \forall y \ U(x) \vee B(x,y)) \wedge (\forall x \forall y \ B(x,y) \vee B1(x,\ y))$	A326555	$a(n) = (2^n + 3^n)^n$ 用于 $n \geq 0$。
$(\forall x \ U(x) \vee U1(x) \vee U2(x)) \wedge (\exists x \ U(x)) \wedge (\exists x \ U1(x))$	A337418	每个位置处集合的数量（采用Hausdorff度量几何）位于两个……之间由完全二部图定义的集合 $K(3,n)$（其中 n 至少为 3）缺少两条条边，且移除的边并不与 3 点部分中的同一个顶点相连，而是与另一部分中的同一个顶点相连。
$(\forall x \forall y \ U(x) \ \vee B(x,y)) \wedge (\exists x \ U(x)) \wedge (\forall x \exists y \ \neg B(y,\ x))$	A337527	生成函数：$\sum_{n \geq 0} \frac{(2^n+1)^n * x^n}{(1+(2^n+1) * x)^{n+1}}$.
$(\forall x \ U\ (x) \ \vee U1\ (x)\ ) \wedge (\forall x \forall y \ \neg U\ (x) \ \vee B(x,\ y))$	A337851	$a(n) = (2^n + 2)^n$。
$(\forall x \forall y \ U(x) \vee B(x,y)) \wedge (\forall x \exists y \ B(x,y) \vee U1(x))$	A337852	$a(n) = (2^{n+1} + 1)^n$。