arXiv:2304.00895v2 [math.CO] 5 Apr 2023

第九个戴德金数的计算

预印本

Christian Jäkel德累斯顿工业大学，01062，德国德累斯顿christian.jaekel(at)tu-dresden.de

摘要

本文介绍了一种计算第九个戴德金数的算法。其关键在于利用自由分配格中的矩阵乘法和对称性，并通过形式概念分析技术检测这些对称性。

Ke关键词： 戴德金数、自由分配格、形式概念分析、区间 s.

1 引言

戴德金数是一类增长迅速的整数序列，以理查德·戴德金的名字命名。其确定问题称为戴德金问题。令 d(n) 表示第 n 个戴德金数。关于以往成果的综述见 [11], , 其中还包含下表：

图1. 戴德金数（截至 $n = 8$）。整数序列 A000372。

n	d(n)	来源
0	2	Dedekind, 1897 [4]
1	3	
2	6	
3	20	
4	168	
5	7581	Church, 1940 [2]
6	7828354	Ward, 1946 [9]
7	2414682040998	Church, 1965 [3]
8	56130437228687557907788	Wiedemann, 1991 [10]

d(n) 有几种不同的解释。例如，d(n) 等于幂集格 $\mathfrak{B}_n := (2^{\{0,\ldots,n-1\}}, \subseteq)$ 中反链的数量，也等于以 n 个变量为自变量的单调布尔函数的数量，或具有 n 个生成元的自由分配格的元素数量。

因此，下一节将说明如何用数值表示自由分配格。随后，第 3 节将讨论（反）同构格区间，这是一种用于计算的工具。第 4 节和第 5 节讨论算法的理论方面，第 6 节则讨论实际方面，以计算第九个戴德金数。

2 自由分配格

令 $\mathbb{D}_n = (\mathrm{D}(n), \preceq)$ 表示由 n 个生成元生成的自由分配格。以下定理见于 [5]:

定理 1. $\mathbb{D}_n$ 的元素与从 $\mathfrak{D}^k$ 到 $\mathbb{D}_{n-k}$的单调映射之间存在一一对应关系。

推论 1. $\mathbb{D}_{n+1}$ 中的元素与数对(x, y) 中的元素 x, y之间存在一一对应关系，使得 $x \leq y$。

证明。 设 x 和 y 是 $\mathbb{D}_n$中的元素。我们有 $\emptyset$ 和{0}作为$\mathfrak{D}$的元素。将 $\emptyset$ 映射到 x ，即可单调地将{0}映射到满足 $x \leq y$ 的每个 y 。□

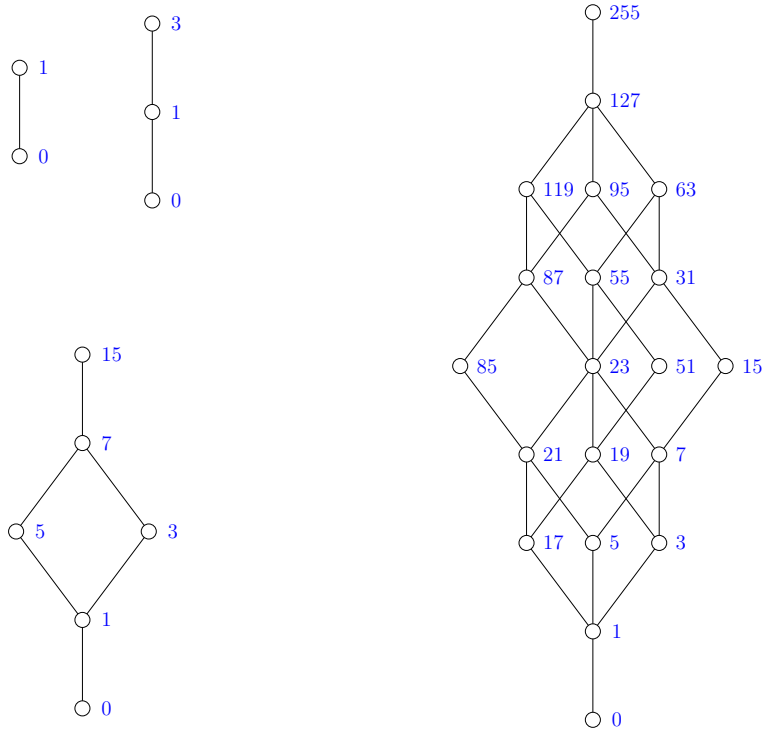
推论 1 提供了一种递归生成 $\mathbb{D}_n$ 中元素的方法。此外，这还给出了 D(n) 的数值表示，有助于高效实现用于计算 d(n) 的算法。为此，将 $\mathbb{D}_0$ 的两个元素分别表示为二进制数 0 和 1。底层序关系按位应用$0 \leq 1$ 。接下来，我们根据推论 1 构造有效数对，并将它们连接成 2 位数，即 00、01 和 11。这个过程可以继续迭代。

示例 1. 本示例展示了从 D(0) 开始生成 D(3) 的过程。

$$\begin{matrix}0\\1\end{matrix} \quad\rightarrow\quad \begin{matrix}00\\01\\11\end{matrix} \quad\rightarrow\quad \begin{matrix}0000\\0001\\0011\\0101\\0111\\1111\end{matrix} \quad\rightarrow\quad \begin{matrix}\text{0000.0000 0001.0011 0101.0101}\\\text{0000.0001 0001.0101 0101.0111}\\\text{0000。0011 0001。0111 0101。1111}\\\text{0000.0101 0001.1111 0111.0111}\\\text{0000.0111 0011.0011 0111.1111}\\\text{0000.1111 0011.0111 1111.1111}\\\text{0001.0001 0011.1111}\end{matrix}$$

图 2 展示了这些数如何构成自由分配格：将“按位或”作为并运算，将“按位与”作为交运算。我们不使用二进制数，而采用更方便的十进制解释，将其表示为整数值。从编程的角度来看，这种表示方式节省内存，而且按位运算速度非常快。

图 2. 具有 0、1、2 和 3 个生成元的自由分配格。



3 格、定义与记号

在本节中，我们始终考虑一个有限格 $\mathbb{L}=(L,\vee,\wedge,\bot,\top)=(L,\leq)$，它具有底元 $\bot$ 和顶元 $\top$。格之间的同构是保持并运算和交运算的双射，因此也保持序。对偶地，反同构 φ 是反转序的双射，因此也会反转并运算和交运算。

$$x\leq y\Leftrightarrow\varphi(y)\leq\varphi(x)\qquad x\vee y\Leftrightarrow\varphi(x)\wedge\varphi(y)\qquad x\wedge y\Leftrightarrow\varphi(x)\vee\varphi(y).$$

定义 1. 对于满足 $a\leq b$ 的 a、$b\in L$，我们定义区间 $[a,b]:=\{x\mid x\in L,\ a\leq x\leq b\}$。若存在将 I 映射到 J 上的同构，则称区间 I 与区间 J 同构，记为 $I\cong J$。类似地，通过相应反同构的存在性来定义反同构的区间 $I\cong_{\mathrm{a}}J$。如果两个区间 I 和 J 同构或反同构，则称它们等价，记为 $I\equiv J$。此外，$\mathrm{Int}(\mathbb{L})$ 表示 $\mathbb{L}$ 的所有区间构成的集合，$\#I$ 表示 I 的基数。

命题 1. 区间的等价是 $\mathrm{Int}(\mathbb{L})$ 上的等价关系。因此，所有区间构成的集合 $\mathrm{Int}(\mathbb{L})$ 可以按 $\equiv$ 取商。

证明。自反性由 $\cong$ 以及每个区间都与自身同构这一事实所保证。此外，由于 $\cong$ 和 $\cong_a$ 的对称性，对称性也成立。为推导传递性，我们考虑三种情况：

$$I\cong J\wedge J\cong K\Rightarrow I\cong K,\quad I\cong_{\mathrm{a}}J\wedge J\cong_{\mathrm{a}}K\Rightarrow I\cong K\quad I\cong J\wedge J\cong_{\mathrm{a}}K\Rightarrow I\cong_{\mathrm{a}}K.$$

□

对于 $[I]\in\mathrm{Int}(\mathbb{L})/_{\equiv}$，我们用 $\#[I]$ 表示 $[I]$ 的等价类的基数。最后，对于每个 $x\in I$，我们引入两个算子，用于解码相对于 I 的顶元和底元而言的区间长度：

$$\bot_{\mathrm{I}}(x):=\#[\bot_{\mathrm{I}},x]\text{ and }\top_{\mathrm{I}}(x):=\#[x,\top_{\mathrm{I}}],$$

其中 $\bot_{\mathrm{I}}$ 表示底元，$\top_{\mathrm{I}}$ 表示 I 的顶元。由于可以从外部上下文（包含算子参数的定义域）推知 I，因此本文其余部分将省略下标 I。

例 2。 这些是 $\mathrm{Int}(\mathbb{D}_2)/_{\equiv}$ 的等价类（参见图 *2*）。

$$\{[0,0],[1,1],[3,3],[5,5],[7,7],[15,15]\},\{[0,1],[1,3],[1,5],[3,7],[5,7],[7,15]\},$$
$$\{[0,3],[0,5],[3,15],[5,15]\},\{[1,7]\},\{[0,7],[1,15]\},\{[0,15]\}.$$

4 $\mathbb{D}_{n+1}$、$\mathbb{D}_{n+2}$ 和 $\mathbb{D}_{n+3}$ 的枚举

在本节中，我们应用定理 1，通过分别计算从 $\mathbb{D}_n^{\,1}$、$\mathbb{D}_n^{\,2}$ 和 $\mathbb{D}_n^{\,3}$ 到 $\mathbb{D}_n$ 的所有单调映射，推导 $\mathrm{d}(n+1)$、$\mathrm{d}(n+2)$、$\mathrm{d}(n+3)$ 的值。这是对下一节的预热，也概述了用于计算某些戴德金数的公式。

定理 2。 以下公式通过 $\mathbb{D}_n$ 计算 $\mathbb{D}_{n+1}$：

$$\mathrm{d}(n+1)=\sum_{a\in\mathrm{D}(n)}\top(a)=\sum_{I\in\mathrm{Int}(\mathbb{D}_n)}1=\#\,\mathrm{Int}(\mathbb{D}_n)=\sum_{[I]\in\mathrm{Int}(\mathbb{D}^n)/_{\equiv}}\#[I].$$

证明。如果将 $\mathbb{D}_n$ 的底元映射到 $a\in\mathrm{D}(n)$，那么任意 $b\in[a,\top]$ 都是顶元的有效像。因此，任意满足 $[a,b]\in\mathrm{Int}(\mathbb{D}_n)$ 的 a、$b\in\mathrm{D}(n)$ 也都是有效的选择（见图3左图）。我们不必对所有区间求和，而可以将求和范围限制在等价的区间上。□

定理3中的第一个方程已在 [1] 中给出。

定理3. 以下公式通过 $\mathbb{D}_n$枚举 $\mathbb{D}_{n+2}$：

$$\mathrm{d}(n+2)=\sum_{a,b\in\mathrm{D}(n)}\bot(a\wedge b)\cdot\top(a\vee b)=\sum_{I\in\mathrm{Int}(\mathbb{D}_n)}(\#I)^2=\sum_{[I]\in\mathrm{Int}(\mathbb{D}^n)/_{\equiv}}(\#I)^2\cdot\#[I].$$

证明. 若按图3中间所示选取 $a, b \in \mathrm{D}(n)$，则 $[a \vee b, \top]$ 中的每个元素以及 $[\bot, a \wedge b]$ 中的每个元素都分别可以作为顶元与底元。另一方面，图3右侧的图像表明，底元和顶元可以映射到每个区间 $[a, b]$，这意味着倒数第二个方程成立。由于同构区间与反同构区间的基数相等，最后一个等式成立。□

下一个定理说明如何用矩阵演算来表示 $\mathrm{d}(n+2)$ 的计算。为此，我们使用方阵的迹算子 Tr，其定义为所有对角元素之和。

定理4. 我们定义两个矩阵 $\alpha(a,b) := \bot(a \wedge b)$ 和 $\beta(a,b) := \top(a \vee b)$，其中 $a, b \in \mathrm{D}(n)$。有如下结论：

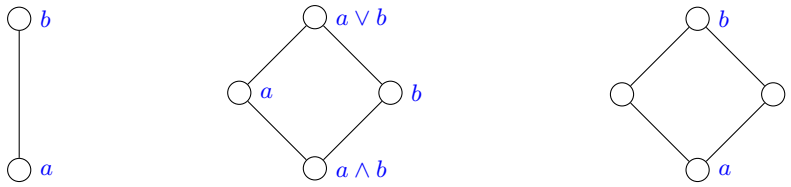
$$\mathrm{d}(n+2) = \mathrm{Tr}(\alpha \cdot \beta).$$

证明。令 $\gamma = \alpha \cdot \beta$ 为矩阵乘积。我们使用定理 3 中的第一个公式，并代入矩阵表达式。注意，α 和 β 是对称的。

$$\mathrm{d}(n+2) = \sum_{a \in \mathrm{D}(n)} \sum_{b \in \mathrm{D}(n)} \alpha(a,b) \cdot \beta(b,a) = \sum_{a \in \mathrm{D}(n)} \gamma(a,a) = \mathrm{Tr}(\gamma).$$

□

图 3. 定理 2 和定理3 的证明图示。



下一个定理的第一个公式与 [7] 中给出的公式相似。

定理5. 令 X 表示所有区间的集合 $\{[\bot, y] \mid y \in \mathrm{D}(n)\}$。以下公式通过 $\mathbb{D}_n$ 枚举 $\mathbb{D}_{n+3}$：

$$\begin{aligned}
\mathrm{d}(n+3) &= \sum_{y \in \mathrm{D}(n)} \sum_{a,b,c \in [\bot, y]} \bot(a \wedge b \wedge c) \cdot \top(a \vee b) \cdot \top(a \vee c) \cdot \top(b \vee c) \\
&= \sum_{[I] \in X/_{\cong}} \#[I] \cdot \sum_{a,b,c \in I} \bot(a \wedge b \wedge c) \cdot \top(a \vee b) \cdot \top(a \vee c) \cdot \top(b \vee c) \\
&= \sum_{I \in \mathrm{Int}(\mathbb{D}_n)} \sum_{a,b,c \in I} \top(a \vee b) \cdot \top(a \vee c) \cdot \top(b \vee c) \\
&= \sum_{[I] \in \mathrm{Int}(\mathbb{D}^n)/_{\equiv}} \#[I] \cdot \sum_{a,b,c \in I} \top(a \vee b) \cdot \top(a \vee c) \cdot \top(b \vee c) \\
&= \sum_{[I] \in \mathrm{Int}(\mathbb{D}^n)/_{\equiv}} \#[I] \cdot \sum_{a,b \in I} \Big(\sum_{c \in [a, \top_I]} \bot(b \wedge c) \Big)^2.
\end{aligned}$$

证明。按照图4左图的方式选取 y，然后选取 a，b，$c \in [\bot, y]$，即可推导出第一个公式。将 X 中同构的元素提取出来后，可得第二个方程。需要注意的是，选取 a，b，$c \in [\bot, y]$ 后，底元仍有一个自由度。若将底元和顶元映射到区间 $[x, y]$ 的边界上，就能消除这一自由度，并得到第三个方程。显然，随后可以提取出同构区间。由于反同构区间也可以提取出来，因此可得第四个方程。为说明其原因，我们对第三个恒等式[1]进行对偶化：

$$\mathrm{d}(n+3) = \sum_{[I] \in \mathrm{Int}(\mathbb{D}_n)} \sum_{a,b,c \in I} \bot(a \wedge b) \cdot \bot(a \wedge c) \cdot \bot(b \wedge c).$$

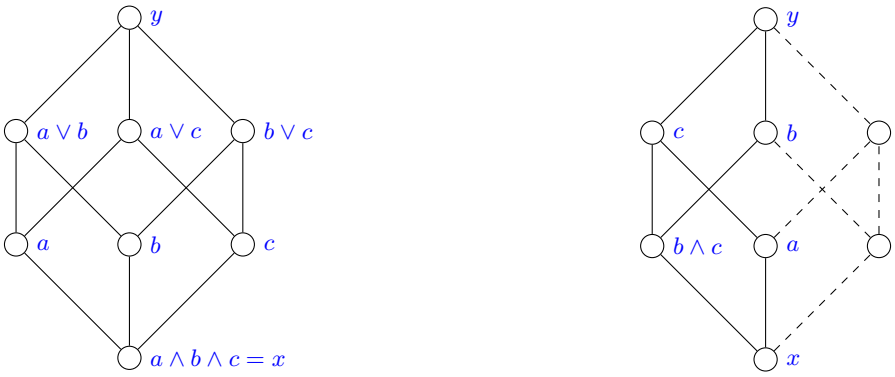
[1] 这意味着 a, b, c 必须放置在比图4所示位置高一个“层级”的位置。

由于一个反同构 同构会反转序，从而交换并运算与交运算；对于 $I \cong_{\mathrm{a}} J$，有 :

$$\sum_{a,b,c\in I} \bot(a\wedge b)\cdot\bot(a\wedge c)\cdot\bot(b\wedge c) = \sum_{a,b,c\in J} \top(a\vee b)\cdot\top(a\vee c)\cdot\top(b\vee c).$$

为得出最后一个方程，我们利用 $\mathfrak{J}^3 \cong \mathfrak{J}^2 \times \mathfrak{J}$。目标是应用 $\mathfrak{J}^3$ 中的垂直对称性（见图4右图）。若选定一个区间 I，则每个 a，$b \in I = [x, y]$ 都可以独立地按图示放置。计算出 $\sum_{c\in[a,y]} \#[x, b\wedge c]$ 后，由于 $\mathfrak{J}^2$ 所诱导的对称性，可以将结果平方。□

图4。 定理5证明所用的图。



与定理4类似，我们通过矩阵演算来表示 $\mathrm{d}(n+3)$ 的计算。

定理 6. 我们定义矩阵 $\alpha(a,b) := \top(a\vee b)$，其中 $a, b \in I \in \mathrm{Int}(\mathbb{D}^n)$。有 :

$$\mathrm{d}(n+3) = \sum_{[I]\in \mathrm{Int}(\mathbb{D}^n)/\equiv} \#[I]\cdot \mathrm{Tr}(\alpha^3).$$

证明。 我们使用定理3的第四个公式，并代入矩阵定义。

$$\mathrm{d}(n+3) = \sum_{[I]\in \mathrm{Int}(\mathbb{D}^n)/\equiv} \#[I]\cdot \sum_{a\in I}\sum_{b\in I}\alpha(a,b)\sum_{c\in I}\alpha(b,c)\alpha(c,a).$$

最内层求和 e xpresses α^2，对 b 求和得到 α^3，而对 a 求和则给出迹算子。 □

我们实现了支持图形处理器的定理 6，并在 Nvidia A100 图形处理器上运行，在 17.56 秒内计算出 d(8)。然而，由于涉及的矩阵最大维度为 7828354×7828354，该算法无法计算 d(9)。

5 $\mathbb{D}_{n+4}$的枚举

本节中，我们应用定理 1，通过计数从 $\mathfrak{J}^4$ 到 $\mathbb{D}_n$ 的所有单调映射，推导 $\mathrm{d}(n+4)$ 的值。

定理 7. 以下公式通过 $\mathbb{D}_n$ 枚举 $\mathbb{D}_{n+4}$:

$$\mathrm{d}(n+4) = \sum_{I\in\mathrm{Int}(\mathbb{D}_n)} X = \sum_{[I]\in\mathrm{Int}(\mathbb{D}^n)/\equiv} \#[I]\cdot X, \ \textit{with:}$$

$$
\begin{aligned}
X = & \sum_{a,b,c,d \in I} \sum_{\substack{e \in [a \vee b, \top] \\ f \in [a \vee c, \top] \\ g \in [a \vee d, \top] \\ h \in [b \vee c, \top] \\ i \in [b \vee d, \top] \\ j \in [c \vee d, \top]}} \top(e \vee f \vee h) \cdot \top(e \vee g \vee i) \cdot \top(f \vee g \vee j) \cdot \top(h \vee i \vee j) \\
= & \sum_{a,b,c,d \in I} \sum_{\substack{e \in [a \vee b \vee c, \top] \\ f \in [a \vee b \vee d, \top] \\ g \in [a \vee c \vee d, \top] \\ h \in [b \vee c \vee d, \top]}} \big(\#[a \vee b, e \wedge f] \cdot \#[a \vee c, e \wedge g] \cdot \#[a \vee d, f \wedge g] \\
& \qquad \cdot \#[b \vee c, e \wedge h] \cdot \#[b \vee d, f \wedge h] \cdot \#[c \vee d, g \wedge h] \big) \\
= & \sum_{a,b,c,d,e,f \in I} (\bot(a \wedge c \wedge d) \cdot \bot(b \wedge c \wedge e) \cdot \bot(b \wedge d \wedge f) \cdot \bot(a \wedge e \wedge f) \\
& \qquad \cdot \top(b \vee c \vee d) \cdot \top(a \vee c \vee e) \cdot \top(a \vee d \vee f) \cdot \top(b \vee e \vee f)).
\end{aligned}
$$

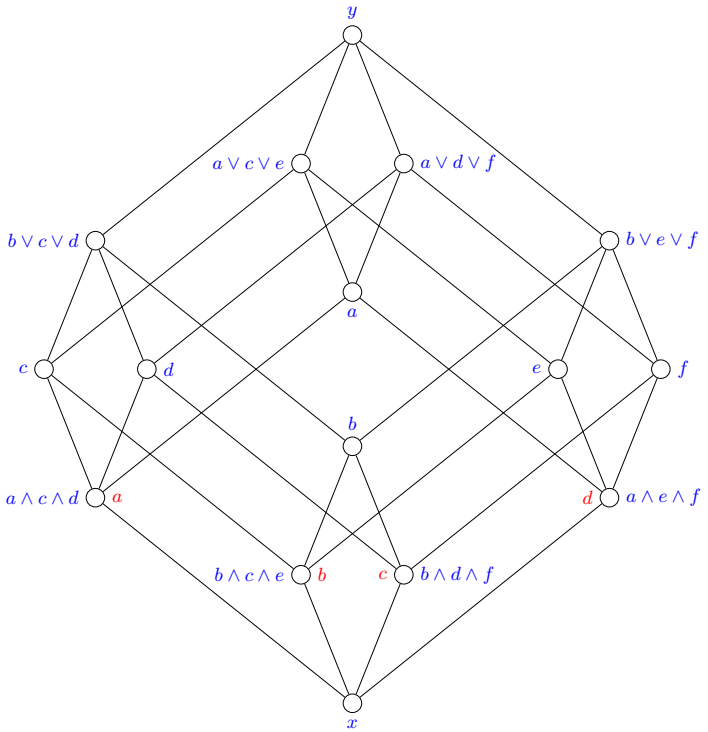
证明。 对于所有三个公式，底元和顶元属于 $\mathfrak{I}^4$ 映射到每个可能区间的上下界 $[x, y]$。但是，正如定理5的证明中所述，我们可以论证只需考虑等价区间。

在第一个方程中，值 a, b, c, d 按图 5 中红色标签所示选取。然后，我们遍历由二元并和区间顶端给出的全部六个范围。这样还剩下四个自由度，其范围从相应的三元并到顶元。

第二个方程对 a， b， c， d采用相同的选择。在接下来的步骤中，我们通过 e， f， g， h遍历由三元并和顶元给出的所有四个范围。这产生六个自由度，其范围从 a， b， c， d的二元并一直延伸到 e， f， g， h各自的二元交。

最后，我们可以如图 5 中蓝色标签所示，将 a, b, c, d, e, f置于相应位置。因此，从每个三元并到区间顶端，以及从区间底端到每个三元交，共有八个自由度。□

图5。将 $\mathfrak{I}^4$ 视为 $\mathfrak{I}^2 \times \mathfrak{I}^2$的图。



定理 7 的最后一个公式可以用矩阵演算表示。

定理 8. 对于 $I \in \mathrm{Int}(\mathbb{D}^n)$ 以及 $a, b, c, d \in I$，我们定义矩阵 α 和 β:

$$\alpha_{ab}(c,d) := \bot(a \wedge c \wedge d) \cdot \top(b \vee c \vee d) \text{ and } \beta_{ab}(c,d) := \bot(b \wedge c \wedge d) \cdot \top(a \vee c \vee d).$$

令 γ_{ab} 为 α_{ab} 与 β_{ab}的矩阵乘积。则有:

$$\mathrm{d}(n+4) = \sum_{[I] \in \mathrm{Int}(\mathbb{D}^n)/\equiv} \#[I] \cdot \sum_{a,b \in I} \mathrm{Tr}(\gamma_{ab}^2).$$

证明。 I将矩阵表达式代入定理 7 中的公式，对于 $d(n+4)$，可得:

$$\sum_{[I] \in \mathrm{Int}(\mathbb{D}^n)/\equiv} \#[I] \cdot \sum_{a,b \in I} \sum_{c,d \in I} \sum_{e,f \in I} \alpha(c,d)\beta(c,e)\alpha(e,f)\beta(d,f).$$

由于 α 和 β 对称，我们可以将两次矩阵乘法视为相同:

$$\gamma(d,e) = \sum_{c \in I} \alpha(d,c)\beta(c,e) \text{ and } \gamma(e,d) = \sum_{f \in I} \alpha(e,f)\beta(f,d).$$

所有指标都来自同一区间（区间 I），这意味着上述矩阵相等。从矩阵的角度出发，我们计算 γ 与自身乘积的对角线，并将其相加 $\sum_{d \in I} \sum_{e \in I} \gamma(d, e)\gamma(e, d)$。 □

观察图 5 可知， a 和 b 彼此独立。结合定理 7 中的公式，可得 a 与 b之间具有旋转对称性。只需考虑满足$a \leq b$的数对。对于严格小于 b 的 a，权重取二；而对于 $a = b$，权重取一。这使计算量减少了近一半。不过，还可以进一步简化。因此，对于每个 $I \in \mathrm{Int}(\mathbb{D}^n)$，令 $\varphi\colon I \to I$为一个同构/反同构。我们在 $I \times I$上引入一种关系。即 $(a, b) \sim (\tilde{a}, \tilde{b}) :\Longleftrightarrow$

$$\exists \varphi : I \to I,\ \varphi(a) = \varphi(\tilde{a}) \text{ and } \varphi(b) = \varphi(\tilde{b}).$$

令 $I \times I\,|_{\leq}$ 表示所有满足 $a \leq b$ 的数对 (a, b)。此外，权重算子 $\omega\colon I \times I\,|_{\leq} \to \{1, 2\}$ 在 $a \neq b$ 时等于 2，否则等于 1。

定理 9. 对于 $I \in \mathrm{Int}(\mathbb{D}^n)$ 以及 a、b、c、$d \in I$，我们定义矩阵 α 和 β:

$$\alpha_{ab}(c,d) := \bot(a \wedge c \wedge d) \cdot \top(b \vee c \vee d) \text{ and } \beta_{ab}(c,d) := \bot(b \wedge c \wedge d) \cdot \top(a \vee c \vee d).$$

令 γ_{ab} 为 α_{ab} 与 β_{ab} 的矩阵乘积。则有:

$$\mathrm{d}(n+4) = \sum_{[I] \in \mathrm{Int}(\mathbb{D}^n)/\equiv} \#[I] \cdot \sum_{[(a,b)] \in (I \times I|_{\leq})/\sim} \omega(a,b) \cdot \#[(a,b)] \cdot \mathrm{Tr}(\gamma_{ab}^2).$$

证明。 关系 $\sim$ 是一种等价关系，同一等价类中的所有数对在枚举公式中产生相同的加项。我们用较简单的示例 $\sum_{z \in I} \bot(a \wedge z) \cdot \top(a \vee z)$ 来说明这一点。在同构 φ 下，满足 $\varphi(a) = \tilde{a}$ （另请注意，顶元与底元保持不变），我们得到:

$$\sum_{\varphi(z) \in I} \bot(\varphi(a \wedge z)) \cdot \top(\varphi(a \vee z)) = \sum_{\varphi(z) \in I} \bot(\tilde{a} \wedge \varphi(z)) \cdot \top(\tilde{a} \vee \varphi(z)) = \sum_{\tilde{z} \in I} \bot(\tilde{a} \wedge \tilde{z}) \cdot \top(\tilde{a} \vee \tilde{z}).$$

由于 φ 是双射，且保持并、交和序，两个和（无论是否应用 φ）都相等。反同构会颠倒序，但这会被关于顶元与底元的对称性抵消。这些论证可以推广到定理 7 中的实际公式。□

我们在下一节中介绍的支持图形处理器的定理 9 实现，可在约 3 秒内计算 d(8)。

6 计算第九个戴德金数的算法

我们列出t 通过定理 9 计算 d($n+4$) 所需的所有步骤，并逐一处理。 ly.

1. 按照第 1 节末尾所述生成 $\mathbb{D}^n$。2. 计算 Int($\mathbb{D}^n$)/$_{\equiv}$ 的等价类，并为每个等价类保存一个代表元及其基数。3. 对于 Int($\mathbb{D}^n$)/$_{\equiv}$的每个等价类代表元 I，计算 ($I\times I\mid_{\leq}$)/$_{\sim}$的等价类。同样，为每个等价类保存一个代表元及其相应的基数。4. 根据定理 9 执行计算:

数据：Int(D n)/$_{\equiv}$ 和 $\forall[I]\in$ Int($_{\mathbb{D}}$n)/$_{\equiv}$: ($I\times I\mid_{\leq}$)/$_{\sim}$ 结果：对于 $[I]\in$ Int(D n)/$_{\equiv}$，d($n+4$) do 对于 $[(a,b)]\in$ ($I\times I\mid_{\leq}$)/$_{\sim}$ do - 生成矩阵 α_{ab} 和 β_{ab}；- 计算矩阵乘积 $\gamma_{ab}=\alpha_{ab}\cdot\beta_{ab}$；- 计算 γ_{ab}^2的迹；- 将迹乘以 $\omega(a,b)$ 和 #[(a,b)]；end - 将上述各值相加；- 将总和乘以 #[I]；end

6.1 步骤 1

生成 $\mathbb{D}^n$ 非常直接，无需特别说明。$\mathbb{D}^4$ 的元素可以用 16 位无符号整数表示，$\mathbb{D}^5$ 的元素可以用 32 位无符号整数表示，而$\mathbb{D}^6$ 的元素可以用 64 位无符号整数表示。存储$\mathbb{D}^6$ 的所有元素约需 63 MB。甚至$\mathbb{D}^7$ 的元素也能在几小时内生成，但存储它们需要约 38.63 TB 内存，因此$\mathbb{D}^7$ 不适合作为计算$\mathbb{D}^9$的基础。

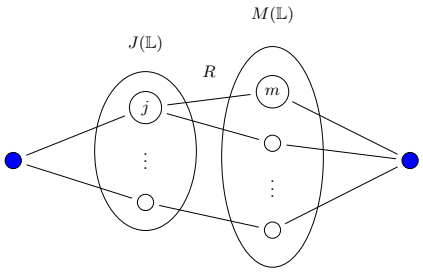
6.2 步骤 2

生成所有区间 Int($\mathbb{D}^n$)是一项简单的工作。下面我们将说明如何确定关于 $\equiv$的等价类，如例 2 所示。为此，我们使用形式概念分析 （见 [6]），该理论研究完备格与二元关系之间的关系。我们将介绍这套理论中有助于实现目标的一些内容。

给定一个完备格 $\mathbb{L}=(L,\leq)$，若一个元素不能表示为 $\mathbb{L}$ 中严格低于（高于）它的其他元素的真并（交），则称其为并不可约元（交不可约元）。令 J（$\mathbb{L}$）表示所有并不可约元，M（$\mathbb{L}$）表示所有交不可约元。$\mathbb{L}$ 可表示为二元关系$\mathbb{R}:=(J(\mathbb{L})\ M(\mathbb{L})\ R)$，其中 $R:=\leq|_{J(\mathbb{L})\times M(\mathbb{L})}$，这称为一个形式背景。可以从形式背景重建 $\mathbb{L}$ 的每个元素，以及它的各种性质和参数。例如，每个 $x\in L$ 都可以表示为集合 $J(x)$（所有等于或低于 x 的并不可约元）或 $M(x)$（所有等于或高于 x 的交不可约元）。由于从 $\mathbb{L}$ 到 $\mathbb{R}$ 的转换可将规模缩减至对数级，因此计算时通常使用 $\mathbb{R}$ 更为有利。

在 [6] 中证明，区间 $[x,y]\in$Int(L) 可以用 R 的一个子关系来描述，即 ($J(y)\ M(x)\ R\cap(J(y)\times M(x))$)。此外，当且仅当存在形式背景同构 $(\alpha,\beta):\ \mathbb{R}_1\to\mathbb{R}_2$ 时，才存在格同构 $\varphi:\ \mathbb{L}_1\to\mathbb{L}_2$。后者定义为一对双射$\alpha:\ J(\mathbb{R}_1)\to J(\mathbb{R}_2)$ 和 $\beta:\ M(\mathbb{R}_1)\to M(\mathbb{R}_2)$，满足 $(j,m)\in R_1\Longleftrightarrow(\alpha(j)\ \beta(m))\in R_2$。格反同构对应于从 $\mathbb{R}_1$ 到对偶形式背景 $\mathbb{R}_2^d:=(M(\mathbb{R}_2)\ J(\mathbb{R}_2)\ R_2^d)$ 的形式背景同构，其中 $R_2^d:=\geq\cap(M(\mathbb{R}_2)\times J(\mathbb{R}_2))$。

图 6. 将形式背景解释为二部图。



形式背景的（反）同构识别问题可以转化为着色二部图的同构问题。这是因为每个形式背景都表示一个二部图（见图 6）。如果我们只想确定形式背景同构，就给两个顶点集分别指定不同的颜色标签，并计算保持这些颜色标签不变的图同构。由于我们也想确定反同构，因此必须允许两个顶点集完全互换，但仍不允许将一个集合映射到另一个集合中。这可以通过添加两个颜色相同的额外顶点，并将它们分别与其所代表的二部图顶点集中的每个顶点相连来实现。

为了实际确定同构图，我们使用软件“nauty”（见 [8]），它可以计算给定彩色图的规范字符串表示。这样就将图同构问题转化为字符串比较，因为当且仅当两个图的规范字符串表示相等时，它们才同构。

数据： Int($\mathbb{D}^n$) **结果：** Int($\mathbb{D}^n/\equiv$ **对于** $[x,y] \in$ Int($\mathbb{D}^n$) **执行** - 计算一个表示 $[x,y]$ 的形式背景；- 将该背景转换为图 6 所示的彩色二部图；- 使用“nauty”计算规范字符串；- 统计每个字符串的出现次数；**结束**

算法 1： 计算 Int($\mathbb{D}^n$)/$\equiv$ 的步骤。

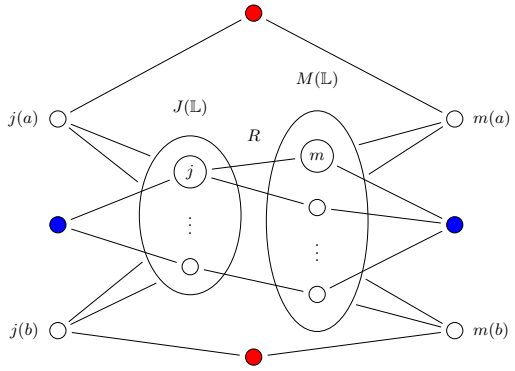
对于 $\mathbb{D}^2$，我们计算得到 6 个等价类；对于 $\mathbb{D}^3$，得到 18 个；对于 $\mathbb{D}^4$，得到 134 个；对于 $\mathbb{D}^5$，得到 9919 个等价类。得益于 nauty 的高效性，最后一次计算在一台 2.4 GHz 台式电脑上仅需几秒钟即可完成。请注意，根据定理 2，$\mathbb{D}^5$有 7828354 个区间。因此，我们将其缩减至仅 0.00127%，这对 d$(5+4)$ 的计算运行时间有显著影响。

6.3 步骤 3

考虑区间 $I \in$ Int($\mathbb{D}^n$)，我们可以生成所有数对 $(a,b) \in I \times I\,|_{\leq}$。为了确定对于给定的数对 (a,b) 和 $(\tilde{a},\tilde{b})$，是否存在满足 $\varphi(a) = \tilde{a}$ 且 $\varphi(b) = \tilde{b}$ 的同构/反同构 $\varphi\colon I \to I$，我们采用了步骤 2 中方法的改进版本，如图 7 所示。

首先，必须生成用于解码 I 上（反）同构问题的二部图（如步骤 2 所示）。接下来，我们通过添加 4 个额外顶点来施加进一步限制，这些顶点分别与关于 a 和 b 的所有并不可约元和交不可约元相连。图 7 中的红色顶点分别连接与 a 和 b 相关的顶点。此配置确保：若我们根据 nauty 的规范字符串表示计算等价类，则 I 上的（反）同构会保持数对 $(a,\ b)$ 不变，并且还可以交换 a 和 b。例 3 展示了 $([0,15]\times[0,15])\ |_{\leq}$ 的等价类。

图 7. 用于解码关于 $\sim$ 的（反）同构问题的图。



例 3。关于 $\sim$ 的 $([0,15]\times[0,15])\ |_{\leq}$ 的等价类。{(0, 0), (15, 15)}，{(1, 1), (7, 7)}，{(3, 3), (5, 5)}，{(0, 1), (7, 15)}，{(3, 5)}，{(0, 3), (0, 5), (3, 15), (5, 15)}，{(1, 3), (1, 5), (3, 7), (5, 7)}，{(1, 7)}，{(0, 7), (1, 15)}，{(0, 15)}。

如果我们对每个 $[I]\in \mathrm{Int}(\mathbb{D}^{\mathbb{II}})/_{\equiv}$ 都执行计算，就需要处理一定数量的数对 (a,b)，并将它们约简为等价类。我们得到的约简结果为 $\mathbb{D}_2$： $56\to 33$，$\mathbb{D}_3$： $1127\to 446$，$\mathbb{D}_4$： $274409\to 80741$ 和 $\mathbb{D}_5$： $8646896880\to 4257682565$。在 AWS "c6i.32xlarge" 集群上使用 128 个核心计算 $\mathbb{D}_5$ 耗时约一小时。有人可能会认为这不值得投入，因为 $\mathbb{D}_5$ 的约简率仅略高于两倍。但可以观察到，较大的区间通常具有更高的约简率。这是因为与较小的区间相比，较大的区间为对称性留下了“更多空间”。例如，$\mathbb{D}_5$ 中最大的区间是 $[0,4294967295]$。对于这个区间，我们的约简结果为 $57471561\to 140736$。这一约简对运行时间影响显著，因为所涉及矩阵的维数为 7581×7581。

6.4 步骤 4

对于给定的 $I\in\mathrm{Int}(\mathbb{D}^{\mathbb{II}})$，我们在 CPU 主机上预先计算 $\bot(\cdot)$ 和 $\top(\cdot)$ 的所有值，并将其传输到 GPU 设备上。随后在该设备上生成矩阵 α 和 β，并进行相乘。通过 CUDA 的“cublasDgemmStridedBatched”内核并行处理一批矩阵。由于数值从整数值转换为双精度，我们必须确保所有值都不大于 2^{53}，以避免精度损失。实际上，我们使用较小的界限 $2^{51},$，因为这有利于另一个估计。

α 和 β 各元素的上界分别为 $\max_\alpha=\bot(a)\cdot\top(b)$ 和 $\max_\beta=\bot(b)\cdot\top(a)$。由此可得 $\max_\gamma=\max_\alpha\cdot\max_\beta\cdot\#I$。我们针对 $\mathbb{D}_5$，计算所有数对 (a,b) 的 $\max_\gamma$。在 118084 种情况下，该估计超过 2^{51}。对于这些情况，我们使用 C++ Eigen 库以 64 位整数精度执行精确的矩阵乘法，并确认 $\max_\gamma$ 小于 2^{51}。

矩阵相乘后，使用手写 CUDA 内核计算 γ^2 的迹。为此，使用 128 位无符号整数值。由于 γ 中的最大元素小于 2^{51} 和 $7581\leq 2^{13}$，我们可以通过 $(2^{51}\cdot 2^{51}\cdot 2^{13})\cdot 2^{13}=2^{128}$ 估算 γ^2 的迹的最大值。这保证了使用 128 位无符号整数精度足以计算该迹。最后，所有迹值的求和以及与等价类基数的乘法，均在主机系统上使用 1024 位无符号整数完成。

7 结论

我们在第 4 节和第 5 节概述了计算戴德金数的公式。其中一些公式可以用矩阵乘法来解释，从而能够在 GPU 设备上快速进行数值运算。第 6 节介绍了计算第九个戴德金数的算法步骤。为此，我们使用形式概念分析来检测格区间中的对称性。这些对称性可以借助图同构软件 nauty 高效地计算。

最后需要说明的是，我们在英伟达 A100 图形处理器上运行了该算法。经过 5311 个 GPU 小时和 4257682565 次矩阵乘法后，我们得到了第九个戴德金数的如下数值：

$$2863865776682984111284691516675984988123 66.$$

参考文献

1. Berman, J., Köhler, P.：有限分配格的基数。吉森数学研讨会报告，第 103–124 页（1976年） 2. Church, R.：若干自由分配结构的数值分析。《杜克数学杂志》，6(3):732–734（1940年） 3. Church, R.：按秩枚举具有 7 个生成元的自由分配格。《美国数学会通报》，(11):724（1965年） 4. Dedekind, R.：《论通过最大公因数对数进行分解》。《全集》，第 2 卷，第 732–734 页（1897年） 5. Fidytek, R., Mostowski, A.W., Somla, R., Szepietowski, A.：计算单调布尔函数的算法。《信息处理快报》**79**(5)，203–209（2001年）。
https://doi.org/https://doi.org/10.1016/S0020-0190(00)00230-1,
http://www.sciencedirect.com/science/article/pii/S0020019000002301 6. Ganter, B., Wille, R.：形式概念分析：数学基础。施普林格纽约出版社（1997年） 7. Hoedt, P.J.：使用 Java 中的 MPI 并行化以求得第九个戴德金数（2015年1月），http://oeis.org/A000372 8. McKay, B.D., Piperno, A.：实用图同构，II。CoRR **abs/1301.1493** （2013年），http://arxiv.org/abs/1301.1493 9. Ward, M.：关于自由分配格阶数的注记。《美国数学会通报》，(52):423（1946年） 10. Wiedemann, D.：第八个德德金数的计算。《序》**8**(1)，5–6（1991年3月）。https://doi.org/10.1007/BF00385808, https://doi.org/10.1007/BF00385808 11. Yusun, T.J.：戴德金数及相关数列（2008年）