arXiv:2601.07444v1 [cs.LO] 12 Jan 2026

亲和数理论的形式化

ZHIPENG CHEN、HAOLUN TANG 和 JINGYI ZHAN

电子与信息学院，上海电机大学，中国上海 201306*e-mail address*：chenzhipeng@sdju.edu.cn

Abstract. 本文介绍了在 Lean 4 证明助手中对亲和数理论进行形式化的方法。若两个正整数 m 和 n 的真因子之和分别等于 n 和m，则称它们为一个亲和数对，其中 m 的真因子之和等于 n ， n 的真因子之和等于m。我们的形式化引入了真因数和函数 $s(n) = \sigma(n) - n$，定义了亲和数对和亲和数的概念，并通过计算验证了历史上著名的亲和数对。此外，我们形式化了基本结构定理，包括对称性、非平凡性，以及与盈数/亏数的联系。一项关键贡献是使用指标移位和 zify 策略，对经典的 Th⁻abit 公式（9世纪）给出了完整的形式化证明。此外，我们还为 Th⁻abit 规则和欧拉广义规则（1747年）提供了完整的形式化证明；这两个规则是生成亲和数对的基本定理。一项重大成果是首次完整形式化 Borho-Hoffmann 生成方法（1986年），该方法包含 540 行代码和 33 个定理，并利用自动代数策略（zify 和 ring）验证复杂的多项式恒等式。我们还形式化了若干扩展内容，包括社交数（真因数和循环）、订婚数（准亲和数对）、奇偶性约束定理，以及互质数对的计算搜索界（$> 10^{65}$）。我们验证了最小的长度5社交循环（Poulet 循环），并通过计算验证了具体实例。该形式化包含 2076 行 Lean 代码，组织为 Mathlib 候选模块和论文专用模块，其中包括 139 个定理，以及用于因数和乘法性与互质性推理的全部必要基础设施。

1. Introduction

亲和数是数论中最古老、最迷人的研究对象之一。若两个正整数 m 和 n 满足： m 的真因数之和等于 n，且同时 n 的真因数之和等于 m，则称它们构成一对亲和数。这个概念可以追溯到古希腊；早在公元前5世纪，毕达哥拉斯学派就已知晓最小的亲和数对（220，284）。古人赋予这些数字神秘的意义，将它们视为完美友谊的象征——正如两个朋友一样，各自的价值恰好等于对方的价值。

亲和数的研究已有两千多年历史。9世纪，阿拉伯数学家 Th⁻abit 发现了一个可系统生成亲和数对的公式；几个世纪后，费马和笛卡尔 [1, 2] 重新发现了这一成果。1636年，费马运用该公式找到了亲和数对 (17296，18416)。两年后，笛卡尔发现了 (9363584，9437056)。1747年，欧拉进行了系统研究，并发现了60个新的亲和数对 [2]。然而，最令人惊讶的发现出现在1866年：一位名叫 Niccol`o Paganini 的16岁意大利男孩找到了亲和数对 (1184，1210)，它

Key words and phrases: 形式化验证、Lean 4、亲和数、因子和、数论

提交至《计算机科学逻辑方法》的预印本

© Z.Chen、H. Tang、和J.Zhan
知识共享

是按数值大小排列的第二小亲和数对，却被所有数学家忽略了两千多年。

迄今已知的亲和数对超过 12 亿组 [3], ，但许多基本问题仍未解决。Erd˝os 证明，亲和数的计数函数 $A(x)$ 增长速度低于线性 [4], ，Pomerance 给出了更精细的界 [5, 6]。亲和数对是否有无穷多组？是否存在两个成员都是奇数的亲和数对？是否存在互素亲和数对？尽管计算搜索已达到 10^{65} [7], ，这些问题仍悬而未决。这种具体实例层出不穷、基本问题却依然未解的局面，使亲和数理论成为形式化研究的理想课题。

使用证明助手对数学定理进行形式化验证，已成为数学研究的重要组成部分。Lean 定理证明器，尤其是其第四版，凭借强大的类型系统和丰富的数学库 Mathlib [8, 9]，已在数学界得到广泛应用。近期的成果包括对正则素数情形下费马大定理的形式化 [10, 11], ，展现了 Lean 处理深层数论结果的能力。与此同时，Koutsoukou-Argyraki 在 Isabelle/HOL 中对亲和数理论进行了形式化 [12], ，并给出了经典生成规则的形式化表述。这些工作不仅验证了数学结论的正确性，也为后续研究提供了可靠的形式化基础。

本文全面形式化了 Lean 4 中的亲和数理论。我们的主要贡献包括：(1) 对三种经典生成方法给出了完整的形式化证明——Th⁻abit 公式（9世纪）、欧拉广义规则（1747年）以及 Borho-Hoffmann 生成方法（1986年）；据我们所知，后者是这一结果首个完整的机器验证证明；(2) 创新地运用 `zify` 和 `ring` 策略，自动处理复杂的多项式恒等式；(3) 形式化扩展内容，包括社交数（真因数和循环）、订婚数（准亲和数对），以及对主要开放问题的形式化表述；(4) 通过计算验证历史上著名的亲和数对，以及 Poulet 的长度5亲和循环。完整的形式化内容共包含 2076 行代码、139 条定理，且没有任何 `sorry` 占位符，是迄今为止对亲和数理论最全面的形式化处理。

2. 数学背景

我们首先奠定亲和数理论的数学基础。设 n 为正整数，并用 $\sigma(n)$ 表示 n的所有正因子的和，即 $\sigma(n) = \sum_{d|n} d$。例如， $\sigma(12) = 1+2+3+4+6+12 = 28$。

定义 2.1（真因子和）。 正整数 n 的**真因子和** 定义为 n 的所有真因数之和（即除 n 本身以外的因子），记作 $s(n)$。等价地，

$$s(n) = \sigma(n) - n = \sum_{\substack{d|n \\ d<n}} d. \tag{2.1}$$

对于 $n = 1$，由于 1 没有真因数，因此我们有 $s(1) = 0$。对于素数 p，由于 p 唯一的真因数是 1，因此我们有 $s(p) = 1$。作为亲和数研究中的经典示例， $220 = 2^2 \cdot 5 \cdot 11$ 的真因数为 $\{1, 2, 4, 5, 10, 11, 20, 22, 44, 55, 110\}$，其和为 $1+2+4+5+10+11+20+22+44+55+110 = 284$。同样， $284 = 2^2 \cdot 71$ 的真因数为 $\{1, 2, 4, 71, 142\}$，其和为 $1+2+4+71+142 = 220$。

定义 2.2（亲和数对）。 两个不同的正整数 m 和 n 称为一个亲和数对，记作 (m, n) 是一对亲和数，当且仅当

$$s(m) = n \quad \text{and} \quad s(n) = m. \tag{2.2}$$

换言之， 利用因数和函数，这可以表示为 $\sigma(m) = \sigma(n) = m + n$。

定义中的要求 $m \neq n$ 排除了完全数。回顾一下，完全数是满足 $s(n) = n$ 的正整数 n；如果允许 $m = n$，那么每个完全数都会构成一个“与自身成对的亲和数对”，这有违“亲和”的本意。

定义 2.3（亲和数）。 正整数 n 若属于某个亲和数对，则称为亲和数。

因此，220、284、1184、1210 等都是亲和数。利用因子和的等价刻画，验证一对数是否为亲和数对很简单：只需分别计算两个数的因子和，并检查两者是否分别等于这两个数的和。

亲和数与另外两个经典概念——盈数和亏数——有着天然的联系。若数 n 满足 $s(n) > n$，则称其为盈数；若满足 $s(n) < n$，则称其为亏数；若满足 $s(n) = n$，则称其为完全数。对于亲和数对 (m, n)，其中 $m < n$，由于 $s(m) = n > m$，我们知道 m 是盈数；而由于 $s(n) = m < n$，我们知道 n 是亏数。因此，在任意亲和数对中，较小的成员是盈数，较大的成员是亏数。

2.1. 亲和数对的经典生成方法。纵观历史，数学家们发展了系统生成亲和数对的方法。本文介绍三个经典生成定理，它们在发现新的亲和数对方面已被证明最为卓有成效。

萨比特于9世纪发现了一种系统生成亲和数对的方法，这是已知最早的亲和数生成公式。

定理 2.4 (Th⁻abit 规则)。 设 $n \geq 1$ 是一个正整数，并定义

$$p = 3 \cdot 2^{n-1} - 1, \tag{2.3}$$

$$q = 3 \cdot 2^{n} - 1, \tag{2.4}$$

$$r = 9 \cdot 2^{2n-1} - 1. \tag{2.5}$$

若 p、 q 和 r 都是素数，则

$$(2^n \cdot p \cdot q, \quad 2^n \cdot r) \tag{2.6}$$

是一个亲和数对。

该公式的条件相当严格：三种特定形式的数必须同时为素数。对于 $n = 2$，有 $p = 5$、$q = 11$、 $r = 71$，它们确实都是素数，因此得到亲和数对（$2^2 \cdot 5 \cdot 11$，$2^2 \cdot 71$）=（220，284）。对于 $n = 3$，$r = 9 \cdot 2^5 - 1 = 287 = 7 \times 41$ 不是素数，因此该公式不适用。对于 $n = 4$，$p = 23$、$q = 47$、 $r = 1151$ 都是素数，因此得到费马亲和数对（17296，18416）。

有趣的是，并非所有亲和数对都能由 Th⁻abit 公式生成。帕加尼尼的发现（1184,1210）就是一个例子——这对数不符合 Th⁻abit 公式所要求的形式。事实上，绝大多数已知亲和数对都无法通过 Th⁻abit 型公式生成。

定理 m 2.5 (欧拉广义规则**).** 令 $1 < m < n$ 为正整数，并定义 e

$$a = 2^{n-m} + 1, \tag{2.7}$$

$$p = 2^m \cdot a - 1, \tag{2.8}$$

$$q = 2^n \cdot a - 1, \tag{2.9}$$

$$r = 2^{n+m} \cdot a^2 - 1. \tag{2.10}$$

如果 p、 q 和 r 都是素数，那么

$$(2^n \cdot p \cdot q, \quad 2^n \cdot r) \tag{2.11}$$

就是一对亲和数。

当 $n - m = 1$ 时，我们有 $a = 3$，该规则便化简为 Th¯abit 公式。目前已知只有两组参数对适用：$(m,\ n)$ =(1，8) 和 (29，40)。

定理 2.6 (Borho-Hoffmann 繁殖法**)。** 设 $(a \cdot u, a)$ 是一对亲和数（称为“繁殖者”数对）。设 $t = \sigma(u)$ 和 n 为正整数，并定义

$$p_1 = t^n(u + 1) - 1, \tag{2.12}$$

$$p_2 = t^n(u + 1)(t - u) - 1, \tag{2.13}$$

$$M = a \cdot u \cdot t^n \cdot p_1, \tag{2.14}$$

$$N = a \cdot t^n \cdot p_2. \tag{2.15}$$

如果 t、 p_1 和 p_2 都是素数，且满足所需的互素条件（即 最大公因数$(a \cdot u, t) = 1$、最大公因数$(a \cdot u, p_1) = 1$ 和 最大公因数$(a, p_2) = 1$），那么 (M, N) 就是一对亲和数。

这种方法在历史上意义重大，因为它可以从已知的亲和数对大规模生成新的亲和数对。

2.2. **相关概念。** 亲和数理论自然推广到一些相关概念，这些概念放宽或扩展了定义条件。

定义 2.7（订婚数）。 两个不同的正整数 m 和 n 称为一对准亲和数对 （也称准亲和数），如果

$$s(m) = n + 1 \quad \text{and} \quad s(n) = m + 1. \tag{2.16}$$

订婚数是亲和数对的一种放宽形式，其中每个真因子和都恰好比另一个数大 1。最小的订婚数对是 (48,75)：有s(48) $= 76 = 75 + 1$ 和 s(75) $= 49 = 48 + 1$。

定义 2.8（社交数）。 由互不相同的正整数 n_1、 n_2、.....、 n_k（其中 $k \geq 2$）组成的序列称为长度为 k 的亲和循环，当且仅当

$$s(n_i) = n_{i+1} \quad \text{for } i = 1, 2, \ldots, k-1, \quad \text{and} \quad s(n_k) = n_1. \tag{2.17}$$

亲和循环中的数称为社交数。

社交数将亲和数对推广为任意长度的循环。亲和数对对应于长度为2的亲和循环。已知最小的长度5亲和循环由 Poulet 于 1918 年发现： $[12496, 14288, 15472, 14536, 14264]$。长度大于2的亲和循环极为罕见。

2.3. 开放问题。 尽管已有两千多年的研究历史，关于亲和数的许多基本问题仍未解决。

关于亲和数对的存在性，目前尚不清楚是否存在无穷多个亲和数对。尽管计算搜索已发现超过 12 亿对，但这并不能证明其数量是无穷的。

关于亲和数的奇偶性，所有已知亲和数对中的两个数都是偶数。是否存在两个数都是奇数的亲和数对，既未得到证明，也未被否定。

关于亲和数的互素性，所有已知亲和数对（m, n）都满足最大公因数$(m,\ n) > 1$。互素亲和数对是否存在也是一个开放问题。

3. Formalization in Lean 4

我们的形式化使用 Mathlib 库在 Lean 4 中实现。本节介绍关键定义与定理，并重点说明形式化过程中的技术考量。

3.1. 核心定义。 形式化从定义真因数和函数开始。Mathlib 已提供函数 `Nat.properDivisors`，它返回 n的所有真因数构成的有限集。在此基础上，我们定义真因数和：

```
定义 properDivisorSum (n : Nat) : Nat := Finset.sum n .


记号 "s" => properDivisorSum
```

清单 1：真因数和函数的定义

我们引入记号 `s` ，在数学中对应于 s 。接下来，我们定义亲和数对：

```
定义 IsAmicablePair（m n : 自然数）: 命题:= m ≠ 0 ∧ n


```

代码清单 2：亲和数对谓词

该定义包含三个守卫条件： $m \neq 0$、 $n \neq 0$ 和 $m \neq n$。前两个条件排除了退化情形，因为在 Mathlib 中，`Nat.properDivisors 0` 会返回空集。第三个条件排除了完全数。

亲和数的定义直接使用了存在量词：

```
定义 IsAmicable (n : 自然数 ) : 命题:= ∃ m : 自然


```

代码清单 3：亲和数谓词

3.2. **基本性质的证明**。 我们首先确立真因数和函数的基本性质。对于 $s(0)$ 和 $s(1)$，证明只需直接化简：

```
定理 真因数和_零：s 0 = 0：=由 simp [真因数和]
定理 真因数和_一：s 1 = 0：=由 simp [真因数和_]
```

列表 4：边界情形

亲和数对对称性的证明只需展开定义并交换条件的顺序：

```
定理 IsAmicablePair . symm { m n : 自然数 } (h : IsAmicablePair m n) : IsA
```

列表 5：对称性定理

要证明素数不可能亲和，需要利用素数的真因子和为 1 这一事实：

```
定理 真因数和_pr ime { p : 自然数 } ( hp : 自然数 . Prime p ) : s p = 1 : = 由
```

列表 6：素数的真因子和

由此可知，素数不亲和：如果 p 是素数且亲和，那么存在 m 使得 $s(p) = m$ 并且 $s(m) = p$。由于 $s(p) = 1$，我们有 $m = 1$，但$s(1) = 0 \neq p$，这就产生了矛盾。

```
定理 not_IsAmicable_素数 { p : 自然数 } ( hp : 自然数 . Prime p ) : not (
Is
hı
si
```

列表 7：素数不亲和

3.3. **特定亲和数对的验证**。 对于特定的亲和数对，我们利用 Lean 的计算能力进行验证。本机_decide 策略会将判定过程编译为本机代码以供执行，从而高效处理数值较大的计算：

```
定理 真因数和_220 :s 220 = 284 := 由 原生_判定 定理 真因数和_284 :s
```

清单 8：验证 220 的真因子和

有了这两个引理，验证亲和数对 (220,284) 就顺理成章了：

```
定理 isAmicablePair_220_284 : IsAmicablePair 220 284 := 证明
  构造
  · 判定        -- 220 ≠ 0
  构造器
  。判定 -- 284 ≠ 0
  构造器
  。确定 -- 220 ≠ 284
  构造器
  。 精确 真因数和_220   -- s (220) = 284。 精确 真因数和
```

列表 9：验证第一个亲和数对

同样，我们验证了 Paganini 的亲和数对 (1184,1210)、欧拉的数对 (2620, 2924) 和 (5020, 5564)，以及费马亲和数对 (17296,18416)。

3.4. **与因数和函数的关系**。 我们确立了真因子和与因数和之间的关系：

```
定理 pro perDiv isorSu m_eq_s um_div isors_ sub ( n : 自然数 ) : s n   =( 有限
集
pr
p
```

列表 10：真因子和与因数和之间的关系

利用这一关系，我们可以给出亲和数对的等价刻画：$(m,\ n)$ 是亲和数对，当且仅当 $\sigma(m) = \sigma(n) = m + n$。

3.5. **与盈数和亏数的关系**。 Mathlib 已经定义了盈数（盈数）和亏数（亏数）。我们证明了，在一对亲和数中，较小的数是盈数，较大的数是亏数：

```
定理 IsAmicablePair . lt_iff {m n : Nat } (h : IsAmicablePair m n) : m   < n ↔ 盈数 m   ∧
亏
_s
```

列表 11：亲和数与盈数/亏数之间的关系 例如，我们验证了 220 是盈数，而 284 是亏数。

3.6. **Th⁻abit 公式的形式化**。这项工作的主要贡献之一，是对 Th⁻abit ibn Qurra 公式进行了完整的形式化证明。主要挑战在于处理 Lean 中的自然数减法，其中 $a - b = 0$ when $a < b$。我们使用一种指标移位技术来解决这一问题：不使用参数 n 及类似 $n - 1$ 的表达式，而是使用 k，其中 $n = k + 1$，从而完全消除指数中的减法。

我们使用移位指标定义关键组成部分：

```
1 def p_thabit (k : 自然数 ) : 自然数 :=  3*  2^k - 1 2 def q_thabit (k : 自然数 ) : 自然数 :=  3
*
k
数
```

列表 12：萨比特公式的重新参数化定义

证明策略包含三个主要步骤：

步骤 1：确立关键的代数恒等式。证明依赖一个关键恒等式：$(p + 1)(q + 1) = r + 1$。这个恒等式将三个素数参数联系起来，并使因子和计算成为可能。我们使用基本的环算术来证明这一点：

```
1 定理 萨比特 _关键_ 恒等式 _带有_ 子项（k：自然数）：2（p_萨比特 k  +  1）*（q_萨比特
k


_
7
萨
```

清单 13：核心代数恒等式

步骤 2：利用乘法性计算 $\sigma(m)$。由于 $m = 2^{k+1} \cdot p \cdot q$，其中各因子两两互素（因为 2^{k+1} 是偶数，而 p 和 q 是奇素数），因此我们可以应用 σ 的乘法性：

$$\sigma(m) = \sigma(2^{k+1}) \cdot \sigma(p) \cdot \sigma(q) = (2^{k+2} - 1) \cdot (p + 1) \cdot (q + 1). \tag{3.1}$$

这里我们使用 $\sigma(2^{k+1}) = 2^{k+2} - 1$（等比级数求和）以及 $\sigma(p) = p + 1$，其中 p 为素数。同理，对于 $n = 2^{k+1} \cdot r$：

$$\sigma(n) = \sigma(2^{k+1}) \cdot \sigma(r) = (2^{k+2} - 1) \cdot (r + 1). \tag{3.2}$$

步骤 3：验证 $\sigma(m) = \sigma(n) = m + n$。利用关键恒等式 $(p + 1)(q + 1) = r + 1$，我们得到：

$$\sigma(m) = (2^{k+2} - 1) \cdot (p + 1) \cdot (q + 1) = (2^{k+2} - 1) \cdot (r + 1) = \sigma(n). \tag{3.3}$$

要验证 $\sigma(m) = m + n$，我们需要证明 $(2^{k+2} - 1) \cdot (p + 1) \cdot (q + 1) = 2^{k+1} \cdot p \cdot q + 2^{k+1} \cdot r$。这个代数恒等式可通过 zify 策略将自然数算术转换为整数算术来验证；在整数算术中，ring 策略可以正确处理减法：

```
1 定理 sigma_等于_m_加_n ( k：自然数 ) ( hk：  1 ≤k)(hp：(p_萨比特 k ) . 素数 ) ( hq：(
2
```

```
对 ( m_萨比特 k). 因子中的 d 求和，d =m_萨比特 k +n_萨比特 k: =通过    ...zify [
hj
```

列表 14：使用 zify 进行代数验证

主定理利用因数和函数的乘法性结合了这些结果：

```
定理 萨比特_规则_一般化 ( k : 自然数 ) ( hk : 1 ≤k)(hp:(p_萨比特 k ) . 素



  构造
  。 精确 sigma_等于_m_加_n k hk hp hq hr 。 rw [← sigma_


```

列表 15：Th⁻abit 一般规则

该定理指出，对于任意 $k \geq 1$，如果 $p = 3 \cdot 2^k - 1$、$q = 3 \cdot 2^{k+1} - 1$ 和 $r = 9 \cdot 2^{2k+1} - 1$ 都是素数，那么 $(2^{k+1} \cdot p \cdot q,\ 2^{k+1} \cdot r)$ 是一个亲和数对。这对应于 $n = k + 1 \geq 2$ 的经典公式。

3.7. **扩展：社交数及相关概念**。为了更全面地介绍亲和数理论，我们在单独的模块 AmicableLib/Extensions.lean 中形式化了若干相关概念。

3.7.1. 订婚数。订婚数（也称为准亲和数对）是对亲和数对的一种放宽：

```
定义 IsBetrothedPair (m n : 自然数 ) : 命题: =
  m ≠ 0 ∧ n ≠ 0 ∧ m ≠ n ∧ s m = n + 1 ∧ s n = m + 1
```

列表 16：订婚数

最小的准亲和数对是 (48,75)：我们有 $s(48) = 76 = 75 + 1$ 且 $s(75) = 49 = 48 + 1$。我们通过计算验证了这一点，第二小的准亲和数对 (140,195) 也是如此。

3.7.2. 社交数。 社交数将亲和数对推广到长度大于 2 的循环。如果对列表中的每个元素应用 s，得到的都是循环中的下一个元素，那么该列表就构成一个亲和循环：

```
定义 IsSociableCycle（L：List Nat）：命题：=L . 长度 ≥ 2 ∧（∀ x 属



```

```
有 hlen： 0 <L. 长度：=证明线性算术
```

列表 17：亲和循环

已知最小的长度5亲和循环由 Poulet 于 1918 年发现：$[12496, 14288, 15472, 14536, 14264]$。我们通过计算验证了这个循环。亲和数对可以视为长度为 2 的亲和循环。

3.7.3. 欧拉规则。 欧拉规则通过引入一个额外参数，将 Th⁻abit 公式推广。对于 $1 < m < n$，定义：

$$a = 2^{n-m} + 1$$
$$p = 2^m \cdot a - 1$$
$$q = 2^n \cdot a - 1$$
$$r = 2^{n+m} \cdot a^2 - 1$$

如果 p、q 和 r 都是素数，那么 $(2^n \cdot p \cdot q,\ 2^n \cdot r)$ 是一个亲和数对。当 $n - m = 1$ 时，我们得到 $a = 3$，这就归结为 Th⁻abit 公式。目前已知仅有两组参数满足条件：$(m,\ n) = (1,\ 8)$ 和 $(29,\ 40)$。

我们在 Lean 中使用以下定义进行形式化：

```
1 def euler_a ( n m : 自然数 ) : 自然数 := 2^( n - m) + 12 3 def p_euler ( n m : 自然数 ) : 自然数
5
n
然
m
Is
1
1
1
1
```

列表 18：欧拉规则的定义

我们在 Lean 4 中给出了欧拉广义规则的完整形式化证明。该证明采用的证明策略与 Th⁻abit 公式相似，并作了适当的推广：

步骤 1：确立关键的代数恒等式。 与 Th⁻abit 情形一样，证明的关键在于恒等式 $(p + 1)(q + 1) = r + 1$。由 $a = 2^{n-m} + 1$，可得：

$$(p + 1)(q + 1) = (2^m \cdot a)(2^n \cdot a) = 2^{m+n} \cdot a^2 = r + 1.$$

这是对 Th⁻abit 恒等式的推广（其中 $a = 3$ when $n - m = 1$）。

步骤 2：利用乘法性计算因子和。 令 $M = 2^n \cdot p \cdot q$ 和 $N = 2^n \cdot r$。由于这些因子两两互素（已通过 27 条支持性引理验证

包括 了奇性、互素性和互异性），我们应用乘法性 ：

$$\sigma(M) = \sigma(2^n) \cdot \sigma(p) \cdot \sigma(q) = (2^{n+1} - 1) \cdot (p + 1) \cdot (q + 1),$$
$$\sigma(N) = \sigma(2^n) \cdot \sigma(r) = (2^{n+1} - 1) \cdot (r + 1).$$

步骤 3：验证 $\sigma(M) = \sigma(N) = M + N$。使用关键恒等式：

$$\sigma(M) = (2^{n+1} - 1) \cdot (p + 1) \cdot (q + 1) = (2^{n+1} - 1) \cdot (r + 1) = \sigma(N). \tag{3.4}$$

最终的代数恒等式 $\sigma(M) = M + N$ 使用与 Th⁻abit 证明中相同的 zify 和 ring 策略进行验证。此外，我们通过计算验证了 $(m, n) = (1, 8)$ 会产生亲和数对 (2172649216, 2181168896)，这既展示了一般性的数学证明，也给出了一个具体实例。

3.7.4. 奇偶性约束。我们对关于亲和数对奇偶结构的深刻定理进行形式化（不加证明）。若 (m, n) 是一个亲和数对，其中 m 为奇数且 n 为偶数，则：

- m 必须是完全平方数
- n 必须具有 $2^k \cdot a^2$ 的形式，其中 $k \geq 1$ 且 a 为奇数。我

们将这些约束形式化为 Lean 定义：

```
定义 奇数成员平方定理：=∀ m n : 自然数，IsAmicablePair m n  →


定义 偶数成员形式定理：=∀ m n : 自然数，IsAmicablePair m n  →


```

列表 19：奇偶性约束定理

这些定理通过 2-adic 赋值论证证明。目前尚未发现奇偶相异的亲和数对，因此这些约束仍停留在理论层面。我们的形式化工作中，历史上已验证的五个亲和数对均为偶数。

3.7.5. 计算搜索界。我们将通过穷尽计算搜索得到的经验观察形式化：在某个界限以下尚未发现互素亲和数对。根据 Pedersen（2003）等人的研究，如果存在互素亲和数对，则其成员之积必须大于 10^{65}：

```
定义 互素 Amicable 搜索界限：自然数：= 10^65

定义 界限以下无互素 Amicable：命题：=∀m n：自然数，m * n  <互素


```

列表 20：计算搜索界限

这是根据计算得出的经验事实陈述，并非数学定理。若不实际进行穷举搜索，就无法在 Lean 中证明它。

3.7.6. *Borho-Hoffmann* 繁殖法。 Borho 和 Hoffmann（1986年） [13] 提出了一种强大的“繁殖”方法，可根据现有的形如（$a \cdot u, a$）的“繁殖对”生成新的亲和数对。给定这样一个繁殖对，且 $t = \sigma(u)$ 为素数时，若满足某些素性条件，该方法便能构造出新的亲和数对（M, N）。

我们首次对这一方法进行了完整的形式化，其中包含 540 行经过验证的 Lean 4 代码、4 个定义和 33 个定理：

```
def t_borho ( u : 自然数 ) : 自然数 :=u  +(sigmau-u)

def M_borho_bh (a u n : 自然数 ) : 自然数 :=令 t:  =


def N_borho_bh (a u n : 自然数 ) : 自然数 :=令 t:  =  t_
```

列表 21：Borho-Hoffmann 构造性定义

该假设要求满足繁殖对性质、三个素性条件以及必要的互素假设：

```
定义 BorhoHoffmannHypothesis ( a u n : 自然数 ) : Prop :=令 t:=  t_borho  u令
```

列表 22：Borho-Hoffmann 假设

我们的主要结果表明，在这些条件下，(M， N) 构成一个亲和数对：

```
定理 borho_hoffmann_rule ( a u n : 自然数 ) ( ha :  0  <a) (hu :  0  <u) (hn :
```

清单 23：Borho-Hoffmann 繁殖定理

证明采用由四个主要步骤组成的结构化方法：

步骤 1：验证繁殖对的性质。给定一个繁殖对 ($a \cdot u$， a)，我们首先验证它确实是一个亲和数对。这意味着 $\sigma(a \cdot u) = a \cdot u + a$ 和 $\sigma(a) = a + a \cdot u$，化简后为 $\sigma(u) = u + a/u \cdot a$。我们将 $t = \sigma(u)$ 定义为 u 的因子和，根据假设，它必须是素数。

步骤 2：建立正性和互素性引理。我们证明 33 个辅助引理，以确立以下性质：

- 正性： $0 < M$， $0 < N$， $0 < p_1$， $0 < p_2$， $0 < t$
- 奇性： p_1、 p_2 和 t 均为奇数（因为它们是大于 2 的素数）
- 互素性： M 和 N 中的因子两两互素，因此可以使用乘法性
- 互异性： $M \neq N$ 且两者均不等于 0 或 1

步骤 3：计算 $\sigma(M)$ 和 $\sigma(N)$，利用乘法性。 令 $M = a \cdot u \cdot t^n \cdot p_1$和$N = a \cdot t^n \cdot p_2$，其中 $p_1 = t^n(u+1) - 1$ 且 $p_2 = t^n(u+1)(t-u) - 1$。利用互素条件以及 σ的乘法性：

$$\sigma(M) = \sigma(a \cdot u) \cdot \sigma(t^n) \cdot \sigma(p_1),$$
$$\sigma(N) = \sigma(a) \cdot \sigma(t^n) \cdot \sigma(p_2).$$

由于 $(a \cdot u,\ a)$ 是一个繁殖对，我们有 $\sigma(a \cdot u) = a \cdot u + a$ 且 $\sigma(a) = a + a \cdot u$。对于素数$t$ 和 p_i，我们使用 $\sigma(t^n) = (t^{n+1} - 1)/(t-1)$ 以及 $\sigma(p_i) = p_i + 1$。

步骤 4：验证 $\sigma(M) = \sigma(N) = M+N$通过自动化策略。 最具挑战性的部分是证明代数恒等式 $\sigma(M) = M + N$和 $\sigma(N) = M + N$。这些恒等式涉及包含六个变量 $(a,\ u,\ n,\ t,\ p_1,\ p_2)$ 和嵌套幂运算的复杂多项式表达式。我们使用 zify 策略将自然数算术转换为整数（在整数中减法的行为符合预期），然后使用 ring 策略自动进行多项式验证，从而克服了这一挑战。整个代数验证可简化为两行策略脚本，展现了自动化处理繁琐计算的强大能力。

这代表了 Borho-Hoffmann 方法的首个机器验证证明 .

繁殖方法彻底改变了计算搜索。在这些方法出现之前，人们只知道不到 1000 个亲和数对。借助生成技术，截至 2020年，已发现超过 12 亿个亲和数对。这些方法将搜索方式从“从头寻找新数对”转变为“根据已知数对生成新数对”，大幅加快了发现速度。

3.8. 开放问题的形式化陈述。 最后，我们将三个著名的开放问题形式化为 Lean 定义：

```
-- No odd-odd amicable pairs exist₂ def
Od
no
de
自
No
m
```

列表 24：开放问题的形式化陈述

这些定义为未来可能开展的证明工作提供了明确的目标陈述 .

4. Discussion

4.1. 计算验证与一般证明。 这种形式化采用两种互补的方法：使用原生_decide 对特定数值进行计算验证，以及针对全称命题进行一般数学证明。

计算验证对于检查特定数字的性质非常有效——验证 $s(220) = 284$ 归结为枚举 220 的所有因子并将其求和，这是一项

计算机非常擅长的任务。我们使用这种方法来验证五对历史亲和数对。

一般证明需要更多数学基础设施。对于 Th⁻abit 公式，我们需要：(1) σ的乘法性，即当 gcd(m, n) = 1时，$\sigma(mn) = \sigma(m) \cdot \sigma(n)$；(2) 关键引理 $\sigma(p) = p + 1$ 和 $\sigma(2^n) = 2^{n+1} - 1$；以及 (3) 互素性结果，证明公式中的 2^n 与 p、q和 r 等奇数互素。

主要的技术挑战是处理自然数减法：在 Lean 中，如果结果本应为负数，减法会截断为零。我们通过指标平移技巧（使用 k 而不是 $n - 1$）和 zify 策略解决了这个问题。zify 策略会将自然数等式转换为整数等式，在整数等式中，标准 ring 策略能够正确发挥作用。这种组合使我们得以完成 Th⁻abit 公式的一般性证明。

4.2. 形式化挑战与策略。 我们的形式化工作遇到了若干典型挑战，需要谨慎选择方法：

自然数算术。Lean 的自然数减法在结果为负数时会截断为零（$5 - 7 = 0$ 在 $\mathbb{N}$中）。这一看似无害的设计选择带来了重大的证明义务。对于 Th⁻abit 公式，当 n 可能为零时，表达式 $3 \cdot 2^{n-1} - 1$ 会产生问题。我们通过指标移位解决了这一问题：不再证明关于 $n \geq 1$ 的公式，而是改写为使用 k，并证明关于 $k \geq 1$ 的公式，从而消除了指数中的减法。随后，zify 策略将算术推理提升到整数范围，使代数变形能够顺利进行。

互素性管理。要证明诸如 $\sigma(2^n \cdot p \cdot q) = \sigma(2^n) \cdot \sigma(p) \cdot \sigma(q)$ 这样的因子和恒等式，必须证明各因子两两互素。我们制定了一套系统方法：先证明一般引理（例如，2 的幂与奇数互素），再将其应用于特定因子。对于 Borho-Hoffmann，我们在假设中明确声明互素条件，而不是试图推导这些条件，以牺牲一般性换取证明的可行性。

代数恒等式验证。Borho-Hoffmann 证明最终归结为验证$\sigma(M) = M + N$，这是一个涉及六个变量和嵌套幂的多项式恒等式。手动验证需要写满好几页繁琐的代数推导。我们改为将等式两边乘以 $(t - 1)$ 以消去除法，然后应用 zify 将其转换为整数算术问题，并使用 ring 自动完成多项式规范化。最终，证明简化为两行策略脚本，充分展现了自动化的强大威力。

计算验证的权衡。对于 (220，284) 这类具体的亲和数对，我们有两种选择：(a) 利用乘法性证明 $\sigma(220) = 504$，这需要用到关于因数分解的引理；或者 (b) 使用 `原生_decide` 直接计算因子和。对于具体示例，我们选择 (b)；对于一般定理，则选择 (a)，以兼顾证明的清晰性和计算效率。

这些经验揭示了形式化初等数论的一般原则：利用自动化处理代数，使用指标移位避免减法，在复杂情形中明确声明互素性，并将计算用于验证，而不是用于证明一般原理。

4.3. 历史发现的形式意义。 通过计算验证五组历史上著名的亲和数对，我们的形式化工作与数学史建立了有趣的联系。毕达哥拉斯学派通过试算和

计算发现了（220，284），而非依据任何系统理论。两千多年后，计算机仅用几毫秒便验证了这一发现。帕加尼尼发现（1184，1210）这组数对，提醒我们，即使在计算机时代之前，细致的手工计算也能发现被忽视的数学对象。费马发现的（17296，18416）则展示了 Th⁻abit 公式的威力 —— 一旦掌握正确的公式，发现新的亲和数对便只需检验素性。

4.4. **形式化的完备性**。本形式化全面涵盖了亲和数理论的基础内容，包括三种经典生成方法的完整证明：图拉的公式（9世纪）、欧拉广义规则（1747年）以及 Borho-Hoffmann 繁殖法（1986年）。这些定理共同代表了亲和数对系统生成方法发展史上的重要里程碑。Borho-Hoffmann 形式化尤其具有重要意义，因为它展示了现代自动推理策略（zify 和 ring）如何处理复杂的代数恒等式，而这些恒等式若由人工验证会十分繁琐。

可能的拓展方向包括：验证更多亲和数对，例如欧拉发现的 60 对中的一些；探究亲和数更深层的性质，例如对“所有已知亲和数对的最大公因数都大于 1”这一观察进行形式化研究；探索 te Riele 的其他繁殖变体 [14]；以及研究其与 Mathlib 中完全数理论的联系。

5. Related Work

5.1. **其他证明助手中的亲和数形式化**。最直接相关的工作是 Koutsoukou-Argyraki 在 I s- abelle/HOL 中对亲和数的形式化 [12], ，发表于 2020年8月 的《形式化证明档案库》。这项工作形式化了欧拉西格玛函数，定义了亲和数对，并给出了若干经典生成规则的形式化陈述，包括 Th⁻abit ibn Qurra 规则、欧拉规则、te Riele 规则，以及带有繁殖者的 Borho 规则。作者还展示了使用 Isabelle 的 simp 和 eval 策略进行素性检查和约数集合枚举的高效计算验证方法。

我们的工作在几个关键方面与 Koutsoukou-Argyraki 的工作不同：

- **证明的完整性**：尽管 Koutsoukou-Argyraki 给出了经典规则的形式化陈述，我们的工作还包含 完整的形式化证明 ，证明了 Th⁻abit 规则、欧拉规则和 Borho-Hoffmann 繁殖法，且没有使用任何 `sorry` 占位符。据我们所知，我们对 Borho-Hoffmann 方法（1986年）的形式化，是对这一结果进行的首个完整机器验证证明。
- **自动化策略**：我们展示了 zify 和 ring 策略在 Borho-Hoffmann 证明中的创新应用，以处理复杂的多项式恒等式，并说明现代自动化如何简化繁复的代数验证。
- **扩展**：我们形式化了其他概念，包括社交数（真因数和循环），并给出了经验证的 Poulet 的五循环示例、订婚数（准亲和数对），以及对主要开放问题的形式化表述。
- **范围**：我们的形式化共计 2076 行，包含 139 个定理，全面涵盖了核心理论及其扩展。

这两种形式化分别验证了机械化这门经典数论的可行性，而 Lean 4 和 Isabelle/HOL 中互补的方法也增强了我们对结果的信心。

5.2. **Lean 中的数论**。 Lean 定理证明器在形式化高等数论方面取得了显著进展。Mathlib 包含约数函数的基础设施，包括 `Nat.divisors`、`Nat.properDivisors` 和用于完全数的 `Nat.Perfect` ，以及关于偶完全数的欧几里得定理证明。这些基础组件为我们研究亲和数奠定了基础。

近期的重要成果包括对正则素数情形下费马大定理的形式化 [10, 11], ，其中涉及代数数论重要部分的形式化，包括库默尔引理和希尔伯特第90—94定理。这项工作基于 Mathlib 丰富的代数结构，展现了 Lean 处理深奥数学定理的能力。我们的亲和数理论形式化虽然更为基础，但通过涵盖经典数论的另一个分支及其独特挑战（积性推理、约数枚举和计算验证），与之形成了补充。

5.3. **其他证明助手中的数论**。 Harrison 在 HOL Light 中对素数定理解析证明的形式化，是形式化数论领域的一项里程碑式成就。通过构建涵盖柯西积分公式等内容的复分析庞大库，Harrison 展示了现代证明助手能够处理复杂的解析论证。由于 HOL Light 是一个极简系统，他的工作需要从第一性原理着手构建大量基础设施。

在 Isabelle/HOL 中，Eberl 的《解析数论九章》 [16] 对狄利克雷级数、黎曼ζ函数和加性组合学中的结果进行了形式化。这项工作展现了 Isabelle 的形式化证明档案库如何作为存放重要数学成果的仓库。为解析数论开发的基础设施与 Koutsoukou-Argyraki 关于亲和数的工作相辅相成，两者都属于同一生态系统。

尽管这些工作关注的领域不同（解析数论与初等数论），但它们与我们的形式化工作有着共同主题：需要谨慎处理算术运算、计算验证对具体实例的价值，以及构建可重复使用的辅助引理库的重要性。

6. Conclusion

本文在 Lean 4 中对亲和数理论及相关概念进行了形式化。我们定义了真因数和函数、亲和数对和亲和数，验证了历史亲和数对，证明了结构定理，并对三种经典亲和数对生成方法给出了完整的形式化证明：Th⁻abit ibn Qurra 公式（9世纪）、欧拉广义规则（1747年）以及 Borho-Hoffmann 繁殖法（1986年）。所有证明均采用严谨的数学技术，包括指标移位、用于处理自然数算术的 zify 策略，以及用于自动化多项式验证的 ring 策略。除了核心理论之外，我们还对相关扩展进行了形式化，包括社交数（将亲和数对推广为循环）、订婚数（准亲和数对）以及奇偶性约束定理。我们通过计算验证了长度5的 Poulet 社交循环，以及分别由欧拉规则和 Th⁻abit 规则得到的特定亲和数对。完整的形式化内容共计2076

行，共包含139条定理且不含任何sorry 占位符，代表了迄今为止最全面的亲和数理论形式化处理。

未来工作有多个可能方向。探索 te Riele 工作中更多的繁殖变体，可以拓展理论覆盖范围。尝试对奇偶性约束定理进行形式化证明，将推动该领域的发展。对开放问题进行形式化研究，即使只能取得部分成果，也可能带来新的见解。将这项工作贡献给 Mathlib，可使其融入更广泛的数论形式化工作。

代码可用性。完整的 Lean 形式化代码可在 https://github.com/chainstart/amicable-numbers 获取。代码分为两类：

- AmicableLib/：适合贡献给 Mathlib 的核心定义与定理，包括基础亲和数理论及其扩展（社交数、欧拉规则、Borho-Hoffmann 生成方法等）
- Amicable/：论文专属内容，包括已验证示例和完整的 Th¯abit 公式证明

该形式化包含 2076 行代码、139 条定理，且没有任何 sorry 占位符。所有主要定理都有完整的形式化证明。代码仓库包含 lean-toolchain 和 lakefile.toml，以确保构建可复现。该形式化基于 Lean 4 和 Mathlib。

致谢。 本研究得到上海市自然科学基金（项目编号：17ZR1411100）资助。

References

[1] Walter Borho. 关于萨比特·伊本·库拉亲和数公式的研究。 *Mathematics of Computation*, 26 118 571 578, 1972 ( ): – . [2] Leonard Eugene Dickson. *History of the Theory of Numbers*, 第 1 卷。卡内基华盛顿研究所，1919 年。[3] Mariano Garcia. 新的亲和数对。 *Mathematics of Computation*, 72(241):491–503, 2003. [4] Paul Erd˝os. 关于亲和数的研究。 *Publicationes Mathematicae Debrecen*, 4:108–111, 1955. [5] Carl Pomerance. 关于亲和数分布的研究。 *Journal für die reine und angewandte Mathematik* 293 294 217 222, 1977 , / : – . [6] Carl Pomerance. 关于亲和数分布的研究（二）。 *Journal für die reine und angewandte Mathematik* 325 183 188, 1981 , : – . [7] Paul Pollack. 关于互素亲和数对的研究。 *Colloquium Mathematicum*, 138(1):81–93, 2015. [8] Leonardo de Moura, Sebastian Ullrich 等。Lean 4 定理证明器与编程语言。载于*Proceedings of CADE-28*，第625–635页，2021年。[9] Mathlib 社区。Lean 数学库。载于 *Proceedings of CPP 2020* 367 381, ，第 – 页。[10] 2020 Alex J. Best、Christopher Birkbeck、Riccardo Brasca 和 Eric Rodriguez Boidi。正则素数的费马大定理（短文）。载于*(ITP 2023)* 268 *LIPIcs* 36 1 36 8, 2023 *14th International Conference on Interactive Theorem Proving* ，卷号 ，页码: – : 。[11] Riccardo Brasca 、 Antoine Chambert-Loir 、 Eric Rodriguez Boidi 等。Lean 中正则素数的费马大定理的完整形式化。 *Annales de la Faculté des Sciences de Toulouse*, 2024年。arXiv:2410.01466。[12] Angeliki Koutsoukou-Argyraki. 亲和数。 *Archive of Formal Proofs*, 2020年8月。https: //www.isa-afp.org/entries/Amicable_Numbers.html. [13] Walter Borho 和 Helmut Hoffmann。大量生成亲和数。 *Mathematics of Computation* 46 173 281 293, 1986 , ( ): – . [14] Herman J. J. te Riele. 如何从已知亲和数对生成新的亲和数对。 *Mathematics of Computation* 42 165 219 223, 1984 , ( ): – . [15] J ohn Harrison. 素数定理解析证明的形式化。 *Journal of Automated Reasoning* 43 3 243 261, 2009 , ( ): – . [16] Manuel Eberl. Isabelle/HOL 中的解析数论九章。载于 *10th International Conference on Interactive Theorem Proving (ITP 2019)* 141 *LIPIcs* 16 1 16 19, 2019 ，卷号 ，页码: – : 。[17] Paul Poulet. 问题 4865。 *L'Intermédiaire des Mathématiciens*, 25:100–101, 1918 年。[18] Henri Cohen. 关于亲和数和社交数的研究。 *Mathematics of Computation*, 24(110):423–429, 1970年。[19] David Moews 和 Paul C. Moews. 搜寻 10^{10}以下的真因数和循环。 *Mathematics of Computation*, 57 196 849 855, 1991 ( ): – . [20] Paolo Starni. 关于奇偶性不同的亲和数的研究。 *arXiv:math/0501402*, 2005年。

AppendixA. Code Statistics

表 1 提供了 Lean 形式化的详细统计数据。

表 1：源文件统计

File	行数	定义	定理	用途
AmicableLib/Amicable.lean	263	3	18	Mathlib 候选
AmicableLib/Extensions.lean	1321	30	90	扩展
(Borho-Hoffmann)	*(540)*	*(4)*	*(33)*	生成方法
Amicable/Basic.lean	25	0	0	重新导出
Amicable/Examples.lean	63	1	5	已验证数对
Amicable/Thabit.lean	404	10	26	Th¯abit 与欧拉证明
总计	**2076**	**44**	**139**	

Appendix B. List of Definitions and Theorems表2列出了所有定义，表3则完整列出了主要定理。

表2：形式化中的定义

Name	描述
真因数和	真因数和函数 $s(n)$
IsAmicablePair	亲和数对谓词
IsAmicable	亲和数谓词
萨比特	萨比特数 $t_n = 3 \cdot 2^n - 1$
p 萨比特，q 萨比特，r 萨比特	重新参数化的萨比特分量
m 萨比特，n 萨比特	萨比特数对成员
萨比特规则陈述	萨比特公式的陈述
OddOddAmicableConj	奇奇亲和数对猜想
InfinitelyManyAmicableConj	无穷多个亲和数对猜想
NoCoprimeAmicableConj	不存在互素亲和数对猜想
已知亲和数对	已知亲和数对列表

表3：主要定理列表

定理名称	描述
基本性质	
真因数和为零	$s(0) = 0$
真因数和为一	$s(1) = 0$
真因数和为素数	对于素数 p, $s(p) = 1$
IsAmicablePair.symm	亲和数对的对称性
IsAmicablePair.one_lt left/right	亲和数对中的数都大于 1
非亲和素数	素数不亲和
不是 IsAmicable one	1 不是亲和数
已验证的亲和数对	
IsAmicablePair 220 284	(220，284) 是一对亲和数
IsAmicablePair 1184 1210	(1184,1210) 是一个亲和数对
isAmicablePair 2620 2924	(2620,2924) 是一个亲和数对
isAmicablePair 5020 5564	(5020,5564) 是一个亲和数对
isAmicablePair 17296 18416	(17296，18416) 是一对亲和数对
与其他概念的联系	
真因数和等于因数和减去	$s(n) = \sigma(n) - n$
IsAmicablePair 当且仅当因数和	通过 σ 对亲和数对的刻画
IsAmicablePair.lt 当且仅当	与盈数／亏数的关系
互素数的因数和乘法	$\sigma(mn) = \sigma(m)\sigma(n)$ 对于互素的 m, n
两个幂的因数和	$\sigma(2^n) = 2^{n+1} - 1$
真因数和二的幂	$s(2^n) = 2^n - 1$
萨比特公式（一般证明）塔比特规则一般形式	适用于所有 $k \geq 1$ 的一般萨比特公式
带下标的萨比特关键恒等式	$(p+1)(q+1) = r+1$
σ m 萨比特	$\sigma(m) = (2^{k+2} - 1)(p+1)(q+1)$
σ n 萨比特	$\sigma(n) = (2^{k+2} - 1)(r+1)$

定理名称	描述
σ(m) ≡ σ(n)	$\sigma(m) = \sigma(n)$
σ(m) ≡ m + n	$\sigma(m) = m + n$
互素的 p、q；塔比特	最大公因数$(p, q) = 1$对于素数 p、q
塔比特规则对 n2 成立	$n = 2$的计算验证
塔比特规则对 n4 成立	$n = 4$的计算验证

本作品采用知识共享署名许可协议授权。如需查看该许可协议副本，请访问 https://creativecommons.org/licenses/by/4.0/，或致信知识共享，第二街171号，300室，旧金山，加利福尼亚州 94105，美国；或艾森纳赫大街2号，10777 柏林，德国