arXiv:2305.01375v1 [cs.MS] 2 May 2023

Diddy：用于无限离散动力系统的 Python 工具箱

Ville Salo*和 Ilkka Törmä†

图尔库大学数学与统计学系 芬兰图尔库

{vosalo,iatorm}@utu.fi

2023年5月3日

摘要

我们介绍了 Diddy，这是一个用于分析无限离散动力系统的 Python 脚本集合。其主要 focus 是广义多维有限型移位（SFTs）。我们展示了如何使用 Diddy 轻松定义有限型移位和元胞自动机，并分析它们的基本性质。我们还演示了如何验证或重新发现编码理论和元胞自动机理论中的一些结果。

关键词：离散动力学；符号动力学；元胞自动机；算法；软件。

1 引言

本文介绍并展示了*Diddy*，这是一个用于定义和分析无限离散动力系统的新型 Python 库和领域专用语言（DSL）。其主要目的是促进对具体多维有限型移位（SFTs）和元胞自动机（CA）的研究，但作者还计划将其扩展到有限图、几何铺砌和替换系统等类别。Diddy 免费且开源，可从 [15] 获取。

Diddy 项目源于作者此前的研究，这些研究近期主要通过计算机辅助证明来探讨离散对象的组合与动力学性质。 [14, 13, 16] 这类研究将极大受益于一种灵活的语言，用于定义多维子移位和元胞自动机（而无需直接处理繁琐的禁用模式或局部规则列表）；还将受益于统一的 SAT 求解器及其他辅助程序接口，以及在需要时能够轻松切换专用语言与 Python。Diddy 既可作为独立的解释型语言使用，也可作为 Python 库使用；本文重点介绍前者。

*ORCID：0000-0002-2059-194X
†ORCID：0000-0001-5541-8517

Diddy 仍在开发中，且正处于快速迭代阶段。我们不保证未来版本与本文中的示例代码兼容。本文撰写时，Diddy 的主要功能如下：

- 能够使用一阶逻辑公式定义有限型子移位和元胞自动机，并通过组合其他元胞自动机构造它们。
- 用于检验有限型移位包含关系和元胞自动机相等性的测试。
- 计算有限型子移位的拓扑熵以及其构型的最小渐近密度的上界和下界。
- 用于显示有限型子移位图案的可视化工具，并能够自动将较小的模式扩展为更大的模式。

2 定义

设 H 是由有限集 $S \subset H$生成的群。若图 $G = (V, E)$ 是 H- 型的，则 H 通过图自同构在 G 上自由作用，且节点被划分为有限个 H-轨道。这意味着存在一个由有限集 $R \subset V$ 组成的代表性节点集，并且边集 E 完全由至少有一个端点位于 R中的边集决定。

下文我们将重点讨论 $G = \mathbb{Z}^d$ 的情形。对于 $\vec{n} \in \mathbb{Z}^d$， R 的平移 $\vec{n} \cdot R$ 称为一个 胞元，这些胞元构成 V 的一个划分。所有 Diddy 对象都定义在一个具有固定代表元集合、类似 $\mathbb{Z}^d$ 的图之上，这称为拓扑。对于下文的大多数定义，可以假设 $V = \mathbb{Z}^d$ 和 $E = \emptyset$，这对应于例如 [9] 中所用的符号动力学标准设定。类似 $\mathbb{Z}^d$ 的一般图便于对 Diddy 中的某些对象建模。

设 A 为一个有限集，称为字母表。图 $G = (V, E)$ 上的一个构型 是一个函数 $x : V \to A$ ，它为每个节点标记 A 中的一个元素。如果 G 是一个类似 $\mathbb{Z}^d$ 的图，那么 $\mathbb{Z}^d$ 通过平移作用于所有构型 A^V 构成的集合上：对于所有 $x \in A^V$ 、 $v \in V$ 和 $\vec{n} \in \mathbb{Z}^d$，$(\vec{n} \cdot x)_v = x_{\vec{n} \cdot v}$ 。我们称 A^V 为全 G移位。构型 $x \in A^V$ 若满足 $\vec{n} \cdot x = x$，则称为 $\vec{n}$周期的，或称为沿 $\vec{n}$周期的。如果构型沿着张成 $\mathbb{R}^d$ 的某些向量 $\vec{n}_1$、.....、$\vec{n}_d \in \mathbb{Z}^d$ 周期，则称其为完全周期的。

在 G 上， A 上的一个有限模式 由一个有限定义域 $D \subset V$ 和一个函数 $P : D \to A$给出。我们将其记为 $D = D(P)$。G 上每个由有限模式构成的集合 F 都定义一个-子移位 $X_F \subset A^V$，它由不存在 F 中任何元素的构型组成：

$$X_F = \{x \in A^V \mid \forall P \in F, \vec{n} \in \mathbb{Z}^d : (\vec{n} \cdot x)|_{D(P)} \neq P\}.$$

如果 F 是有限的，那么 X_F 就是有限型移位，简称 SFT。

设 $X \subset A^V$和 $Y \subset B^V$为 G-子移位。设 $R \subset V$为代表节点。块映射 是一个函数 $f : X \to Y$，由有限邻域$N \subset V$和一个局部规则 $F : A^N \to B^R$定义，其中 $f(x)_{\vec{n} \cdot r} = F((\vec{n} \cdot x)|_N)_r$对所有 $r \in R$和 $\vec{n} \in \mathbb{Z}^d$均成立。元胞自动机 是从一个完全的 G-移位到其自身的块映射。

设 $W : A \to \mathbb{R}$ 为一个函数，我们将其理解为赋予 A中各元素权重 的函数。构型 $x \in A^V$ 的上密度 定义为

$$W(x) = \limsup_{k \to \infty} \frac{\sum_{\vec{n} \in [-k,k]^d} \sum_{r \in R} W(x_{\vec{n} \cdot r})}{|R|(2k+1)^d}.$$

子移位 $X \subset A^V$的最小密度 $W(X)$ 为 $\inf_{x \in X} W(x)$。已知，子移位 X 的最小密度总能由某个构型 $x \in X$ 达到，并且在 $W(x)$ 中的上极限实际上是一个极限。

设 $\mathcal{V}$ 是一组变量。关于 $\mathcal{V}$ 的布尔公式可以是一个变量 $v \in \mathcal{V}$，也可以是以下形式之一：$\neg\phi$、$\phi \vee \psi$ 或 $\phi \wedge \psi$，其中 ϕ 和 ψ 是关于 $\mathcal{V}$ 的布尔公式。如果存在赋值 $\pi : \mathcal{V} \to \{真、假\}$，使得将变量替换为其 π 值后，ϕ 的计算结果为真，则公式 ϕ 是可满足的。若公式的形式为 $\bigwedge_{i=1}^{k} \bigvee_{j=1}^{n_i} a_{i,j}$，且其中每个 $a_{(i,j)}$ 都是一个变量或变量的否定，则该公式属于合取范式（CNF）。每个公式都等价于一个合取范式公式。布尔可满足性问题（SAT）是以下判定问题：给定一个合取范式布尔公式，判断它是否可满足。SAT 是 NP 完全问题，但现代 SAT 求解器在求解许多类型的实际 SAT 实例时效率极高。Diddy 通过 PySAT 库 [6] 使用 Glucose 4.1 求解器 [1] 。

一个（最大化）线性规划由一组变量 $\mathcal{V}$ 和一组约束组成，其形式为 $a_1 v_1 + \cdots + a_n v_n \bowtie b$，其中 $\bowtie$ 是 $\leq$、$=$ 或 $\geq$ 之一，每个 v_i 都是一个变量，而 $a_1, \ldots, a_n, b \in \mathbb{R}$ 都是常数。对变量 $v \in \mathcal{V}$ 求解线性规划，意味着要找到变量的一个赋值 $\pi : \mathcal{V} \to \mathbb{R}$，使得每个约束都成立，且 $\pi(v)$ 的值最大。显然，也可以改为最小化该值。已有高效算法可用于求解线性规划。Diddy 使用 Pulp 库的默认求解器 [12]。

3 表示

在 Diddy 中，拓扑 $G = (V, E)$ 通过固定代表元有限集 $R \subset V$以及有限集 E_0 来定义，其中包含三元组 $(r, s, \vec{n}) \in R^2 \times \mathbb{Z}^d$。然后，通过平移 R和 E_0，得到 G的节点 $V = \{(\vec{v}, r) \mid \vec{n} \in \mathbb{Z}^d, r \in R\}$以及边 $E = \{((\vec{m}, r), (\vec{m} + \vec{n}, s)) \mid \vec{m} \in \mathbb{Z}^d, (r, s, \vec{n}) \in E_0\}$。$\mathbb{Z}^d$ 的作用由 $\vec{n} \cdot (\vec{m}, r) = (\vec{m} + \vec{n}, r)$ 给出。例如，二维方格网可以用 $R = \{a\}$和 $E_0 = \{(a, a, (0, 1)), (a, a, (1, 0))\}$来指定。

为了使用 SAT 求解器分析符号动力系统对象，我们必须将相关问题编码为 SAT 实例。Diddy 可以用两种方式表示有限型子移位 $X \subset A^V$ over G in：一组具体的禁用模式，或一个表示该集合补集的布尔公式。首先，大小为 m 的字母表 $A = \{a_0, \ldots, a_{m-1}\}$ 由一组 $m - 1$ 个布尔变量 $v_1, \ldots v_{m-1}$ 表示，并附带公式$\phi_A(v_1, \ldots, v_{m-1}) = \bigwedge_{1 \leq i < j < m} \neg v_i \vee \neg v_j$，该公式规定这些变量中至多一个为真。其含义是，全为假的赋值表示 a_0，而将某个 v_i 赋值为真则表示 a_i。

假设 $D \subset V$ 是一个有限集，ψ 是变量集 $\mathcal{V}$ 上的一个布尔公式，该变量集包含 $v_{e,i}$（用于 $e \in D$ 和 $1 \leq i < m$），也可能包含一些辅助变量。它按如下方式定义一个有限型子移位 $X_\psi \subset A^V$。当且仅当对于每个 $\vec{n} \in \mathbb{Z}^d$，都存在一个赋值π：$\mathcal{V} \to \{真、假\}$，使得 ψ 的求值结果为真；并且对于每个 $e \in D$，将赋值 π 限制在 $v_{e,1}, \ldots, v_{e,m-1}$ 后，其表示 $(\vec{n} \cdot x)_e$，此时构型$x \in A^V$ 才属于 X。换言之，我们根据 x 在 $\vec{n}$ 处的局部模式来设定变量 $v_{e,i}$ 的取值，并要求部分求值后的 ψ 可满足。

每个有限型子移位都使用布尔公式表示，只有在需要时才使用禁用模式表示。原因是，将禁用模式列表 con-

vertinga list of forbidden patterns F 转换为合取范式公式 ϕ_F 很简单，而且 ϕ_F 的大小（以计算机内存计）与 F 的大小呈线性关系。然而，布尔公式可能需要数量呈指数级增长的禁用模式。

要表示块映射，我们需要指定邻域 $N \subset V$ 和局部规则 F： $A^N \to B^R$。只需描述对于每个节点 $r \in R$ 和符号 $b \in B$，满足 $F(P)_r = b$ 的模式集合 $P \in A^N$。这可以像我们处理有限型移位时那样，用布尔公式编码。因此，块映射由 $|R||B|$ 个布尔公式定义。

4 定义拓扑、有限型移位和块映射

回想一下， $\mathbb{Z}^d$类拓扑 $G = (V, E)$ 由指定代表元集合 $R \subset V$以及三元组集合 E_0 来定义，其中三元组的形式为 $(r, s, \vec{n}) \in R^2 \times \mathbb{Z}^d$。在 Diddy 中，三元组都有名称，可以通过这些名称引用。例如，以下代码定义了二维六边形网格（也可通过内置命令 %topology hex 使用）。代表元为 $R = \{0, 1\}$，共有六个三元组，名称分别为 lt、rt、up 和 dn。如果三元组来自不同的顶点，它们可以同名。我们还演示了字母表的定义；此处为二元字母表 $A = \{0, 1\}$。

%dim 2 %nodes 0 1 %topology lt (0,0,0) (-1,0,1); lt (0,0,1) (0,0,0); rt (0,0,0) (0,0,1); rt (0,0,1) (1,0,0); up (0,0,0) (0,1,1);dn (0,0,0) (0,-1,1) % alphabet 0 1

在 Diddy 中，有限型移位可以通过两种方式定义：列出禁用模式，或使用编译为中间电路表示、再转化为 SAT 实例的一阶公式 （一阶公式）。本文撰写时，有效一阶公式的语言包含以下元素（本文不对此作形式化定义）：

- 取值范围为 V 的节点、 $\mathbb{Z}^d$ 的单元格、 A 的符号或真值的变量。

- 沿边移动：如果 x 是一个节点变量，其值的形式为$n = (\vec{m}, r)$，而 m 是三元组 $(r, s, \vec{n})$ 的名称，那么 x.m 就是节点 $(\vec{m} + \vec{n}, s)$。如果 x 是胞元或节点，而 s 是节点 s 的名称，那么 x.s 则表示胞元 x 中的节点 $(\vec{m}, s)$。

- 相等与邻接关系：如果 x 和 y 是节点变量，x @ y 表示它们是同一个节点，而 x ~ y 表示它们在 G 中相邻。如果 x 是节点变量而 y 是符号变量（或字面符号），x = y 表示 x 具有符号 y。如果 y 是节点变量，则表示这些节点具有相同的符号。可以用 x !@ y、x !~ y 和 x != y 表示它们的否定。

- 逻辑连接词：Diddy 提供前缀运算符 !（否定），以及中缀运算符 &（合取）、 | （析取）、-> （蕴涵）和 <->（等价），其语义与通常定义相同。

- 受限量化：Ey[x2]定义一个新的节点变量 y，其存在量化范围为 $B_2(x)$，即以已有节点变量 x 为中心、相对于 G 的路径距离半径为 2 的球。方括号内可以有多个限制条件：例如，在 Ez[x2y1] 中，变量 z 的取值范围为并集 $B_2(x) \cup B_1(y)$。Ay[x2] 是对应的全称量词。EC 和 AC 量化的是单元格，而不是节点。
- 局部定义：令 func a b := a @ b | (a = 0 & b = 0) in 定义一个辅助的双参数公式 func，用于检查它的两个参数是否为同一个节点，或者是否都包含符号 0。在定义之后的代码中，可以通过 func x y 调用它。辅助公式可以有任意数量的参数，这些参数可以是节点变量或符号变量。

可以使用命令 %SFT name formula 定义有限型子移位。下面通过一个例子说明其语法。图$G = (V, E)$上的半径为r 识别码 [8] 是一个子集 $C \subset V$，满足

- 所有 $v \in V$都满足 $B_r(v) \cap C \neq \emptyset$，并且
- 所有 $v \neq w \in V$都满足 $B_r(v) \cap C \neq B_r(w) \cap C$。

其基本思想是，C 中的节点充当“传感器”：如果距离 r以内的任何节点发生事件，它们就会发出警报。事件发生时，我们希望至少能收到一个警报，并能够根据发出警报的传感器集合推断事件发生的位置。半径为 1 的识别码对应的有限型子移位可定义如下：

```
%有限型子移位 idcode Ao let cnbr u v := v=1 & u~v in
  (Ed[o1]cnbr o d) & (Ap[o2] p!@o -> Eq[o1p1] (cnbr o q & p!~q) |
  (cnbr p q & o!~q))
```

我们首先定义辅助公式 cnbr u v：当 v 属于 C（建模为具有符号 1）且与 u 相邻时，该公式为真。下一行中，我们检查全称量化的 o（表示任意节点）在 C 中是否有一个邻居 d。最后，我们检查距离 o 不超过 2 的所有节点 p 是否满足一个条件：如果 p 和 o 不同，那么它们中应有一个节点拥有邻居 q，且 q 属于 C，但不是另一个节点的邻居。请注意，此定义独立于拓扑 G。

目前，Diddy 尚不支持一般块映射的语法。定义全移位上的元胞自动机的语法为 %CA name preimages，其中 preimages 是一组命令列表，命令的形式为 node symbol formula。该公式描述了元胞自动机的局部规则 $F : A^N \to A^R$ 何时应将符号 symbol 写入节点 node，而该节点是 R 的一个元素。与描述有限型移位时类似，N 只是布尔公式所引用的节点集合。

我们再次省略更精确的描述，转而用一个例子来说明。考虑定义在双轨二进制字母表上的三个元胞自动机 L, R, F，即 $A = \{0,1\}^2$，其中 L 和 R 分别将上轨道向左和向右移位，而 F 则将上轨道加到下轨道上（模 2）。这三个元胞自动机均可逆，它们的复合构成一个与灯夫群 $\mathcal{L}$同构的群。它是花环积 $\mathbb{Z}_2 \wr \mathbb{Z}$，即群 $\bigoplus_{n\in\mathbb{Z}} \mathbb{Z}_2 \rtimes \mathbb{Z}$ 的半直积，其中 $\mathbb{Z}$ 通过移位作用于无限直积 $\bigoplus_{n\in\mathbb{Z}} \mathbb{Z}_2$。灯夫群的一个元素可以看作一条指令，用于移位一条双向无限的比特磁带，并翻转有限个磁带单元。

%alphabet0 1 %nodes 上 下 -- 两条轨道，上轨道和下轨道 %dim 1 %topology rt (0, 上) (1, 上); rt (0, 下) (1, 下); lt (0, 上) (-1, 上); lt (0, 下) (-1, 下) %CA R -- 上轨道上的部分右移位 上 1 ACo o.上.lt=1下 1 ACo o.下=1%CA L -- 上轨道上的部分左移位 上 1 ACo o.上.rt=1下 1 ACo o.下=1%CA F -- 将上轨道加到下轨道 上 1 ACo o.上=1下 1 ACo (o.下=1 | o.上=1) & (o. 下 =0 | o. 上 =0)

5 有限型移位的比较

给定 G 上的两个有限型子移位 X、$Y \subset A^V$，它们是否相等 $X = Y$？一般而言，对于 $d \geq 2$，这个问题是不可判定的；即使固定 $Y = \emptyset$，也是如此，正如 Berger 在 [2] 中所示。因此，我们无法实现用于检验有限型子移位相等性的全算法。然而，以下结论成立：

1. 如果 $X \subseteq Y$，那么可以通过计算验证。这是基于紧致性论证。设 F 和 F' 分别是 X 和 Y 的禁用模式集。如果 $X \subseteq Y$，那么必定存在一个有限集 $N \subset \mathbb{Z}^d$，使得对于所有 $P' \in F'$ 和 $\vec{n} \in N$，若 $x \in A^V$ 满足 $(\vec{n} \cdot x)|_{D(P')} \neq P'$，则 $x|_{D(P)} \neq P$ 对所有 $P \in F$ 都成立。2. 如果存在全周期构型 $x \in X \setminus Y$，那么只需枚举全周期构型，并检查它们是否包含 X 和/或 Y 的禁用模式，即可通过计算找到该构型。特别地，如果 X 具有稠密的全周期点（许多自然例子都是如此），那么 $X \subseteq Y$ 是可判定的。

由此得到一种用于检查包含关系 $X \subseteq Y$ 的部分算法。对于递增的 $k = 1$、2、...，检查条件 1 是否在 $N = [-k, k]^d$ 下成立；若成立，则返回 “ 是 ” ；否则，针对周期 $\vec{n}_i$ =(0, $\cdots$, 0, k, 0, $\cdots$, 0)检查条件 2，若发现这样的周期构型，则返回“否”。这实质上是 Wang 用于检查一组彩色方形瓦片是否能铺满无限平面 [18] 的半算法。命令 %equal 使用 SAT 求解器，通过公式表示有限型子移位来执行检查。当然，如果 $X \setminus Y$ 非空但不包含任何全周期构型，该算法就永远不会终止。

6 计算与比较块映射

给定两个 G-有限型移位 $X \subseteq A^V$和 $Y \subseteq B^V$以及块映射 $f, g\colon X \to Y$，判断 $f = g$ 是不可判定的，原因与有限型子移位之间的相等性检查相同：如果 f 和 g 的局部规则总是给出不同的输出，那么 $f = g$

当且仅当 $X = \emptyset$。但是，如果 $X = A^V$ 是满移位，这个问题就“仅仅”是 co-NP 完全，因为此时我们只需检查是否不存在这样的输入： f 的某个电路求值为真，而 g 的电路求值为假，或反之。Diddy 通过 %equal_CA 命令提供此功能。

此外，可以组合元胞自动机：既可以使用命令 %compose_CA name CA_list 显式组合，也可以枚举给定一组元胞自动机在指定长度以内的所有组合，并报告其中哪些组合相等，这可以通过 %calculate_CA_ball bound filename CA_list 实现。后一个命令会将结果写入日志文件。它的名称源于该命令本质上分析了由给定自动机生成的半群的凯莱图中一个球的形状和大小。

继续看灯笼人示例，考虑元胞自动机 $\alpha = FR^3FL^5FR^2$ 和 $\beta = L^2FR^5FL^3F$。在这两种组合中，上轨道都与下轨道相加，而下轨道分别移位 -2、0 和 3 步（顺序不定）。因此，它们应该相等。我们可以在 Diddy 中验证这一关系：

%compose_元胞自动机alpha F R R R F L L L L L F R
R %compose_元胞自动机beta L L F R R R R R F L L L
F %equal_元胞自动机 alpha beta

Diddy 还允许我们计算元胞自动机的阶，至少在阶较小时如此。考虑定义在 $\mathbb{Z}$ 上、字母表为 $\{0,1\}$ 、邻域为 $\{0,1,2,3,4,5,6\}$ 的元胞自动机 f，它将

$$f(x)_0 = 1 \iff x_{[0,6]} \in \{0111000, 1000100, 0111100, 1010110, 1111110, 0010001, 0101001, 1101001, 0000101, 0010101, 0011101, 0000011, 1110011, 0011011, 1011011, 0111011, 1111011, 1000111, 1100111, 0010111, 1110111, 0001111, 0111111\}.$$

第一作者在 AUTOMATA 2017 会议的特邀报告中提出，这个元胞自动机可能是幂零的，并证明其幂零度（使 f^n 将所有内容映射为 0 的最小幂 n）至少为 7。后来，作者通过一个临时构造的证明表明，其幂零度至多为 9。

使用 Diddy，我们可以直接计算其幂零度。这个元胞自动机可以相当直接地转换为 Diddy 代码（省略了若干行）：

%dim 1 %字母表 0 1 %节点 0 %拓扑 rt (0,0) (1,0); lt (0,0) (-1,0) %元胞自动机 f 0 1 Ao let x a b c d e f g :=o=a & o.rt=b & o.rt.rt=c & o.rt.rt.rt=d & o.rt.rt.rt.rt=e& o.rt.rt.rt.rt.rt=f & o.rt.rt.rt.rt.rt.rt=g in x 0 1 1 1 0 0 0 | x 1 0 0 0 1 0 0 | [...] | x 0 1 1 1 1 1 1

这里我们定义一个谓词 x，用于检查原点的邻域中是否存在某个特定的词，然后对上面集合中的元素取析取。现在我们也定义零元胞自动机，并计算这对元素生成的球：

%CA zero 0 1 0=1%calculate_元胞自动机_
ball10 outfile f zero

输出文件包含多行内容，但最相关的一行是 zero = f f f f f f f。它表明 f 的七次幂是零元胞自动机。如果我们已经知道（或正确猜出了）阶，也可以使用 %compose_CA seventh f f f f f f f 和 %equal_CA seventh zero 更快地计算出这一结果。

7 最小密度

在许多应用中，确定有限型子移位 $X \subset A^V$ 相对于某些字母表权重 W：$A \to \mathbb{R}$的最小密度 $W(X)$ 十分重要。Diddy 包含两种用于近似 $W(X)$ 的算法：一种用于求上界，另一种用于求下界。它们都不能保证求出真值，但在实践中往往能给出不错的结果。

7.1 通过周期构型求上界

我们先介绍求上界的算法。选取一组 $d-1$向量 $K = \{\vec{n}_2, \vec{n}_3, \ldots, \vec{n}_d\} \subset \mathbb{Z}^d$，使得 $\{\vec{e}_1, \vec{n}_2, \vec{n}_3, \ldots, \vec{n}_d\}$ 构成$\mathbb{R}^d$的一组基，其中 $\vec{e}_1 = (1, 0, \ldots, 0)$ 是第一个标准基向量。令 $X_K \subset X$ 为沿每个 n_i 周期的配置所组成的集合 $2 \le i \le d$。那么X_K也是一个有限型子移位，并且 $W(X_K) \ge W(X)$。此外，我们声称，$W(X_K)$ 可以用相当有效且可并行化的方式计算。事实上，我们已经在 [16]中宣布了一个特殊情形。要计算 $W(X)$ 的上界，只需选取合适的周期集合 K 并应用该算法。请注意，如果 X 恰好是非周期的，那么无论选择何种 K，该算法都无法得出结果；无论如何，其表现取决于周期构型对密度 $W(X)$ 的逼近程度。

在此算法中，我们用一个禁用模式集合 F 来表示 X。设 $D \subset \mathbb{Z}^d$ 为向量 $K \cup \{\vec{e}_1\}$ 的基本域，并记作 $U = \{a_2\vec{n}_2 + \cdots + a_d\vec{n}_d \mid a_2, \ldots, a_d \in \mathbb{Z}\}$。设 $B = \bigcup_{\vec{u} \in U} \vec{u} + D$ 为边界，并将 $B^+ = \{(\vec{b} + p\vec{e}_1) \cdot r \mid p \ge 0、\ r \in R\}$ 称为顶点集 V 的右侧，将 $B^- = \{(\vec{b} + p\vec{e}_1) \cdot r \mid p \le 0、\ r \in R\}$ 称为其左侧，其中 $R \subset V$ 是代表元集合。集合 X_K 本质上是一个一维有限型子移位，其字母表为 $A^{D \cdot R}$，动力学是按 $\vec{e}_1$ 平移。我们将其表示为标记有向图 $G' = (V', E')$，如下所示。一个顶点 $v \in V'$ 是一个平移后的有限图样集合 $Q = \vec{m} \cdot P$，其中 $P \in F$ 且 $\vec{m} \in \mathbb{Z}^d$ 在按 U 平移下保持不变：若 $Q \in v$ 且 $\vec{n} \in U$，则 $\vec{n} \cdot Q \in v$。我们还要求，对于每个 $Q \in v$，其定义域 $D(Q)$ 与 B^+ 和 B^- 均相交。请注意，虽然每个 $v \in V'$ 在严格意义上都是一个由有限集构成的无限集，但 U-不变性和交性质确保它可以用有限量的数据编码。特别地，V' 是一个有限集。

用 v_0 表示满足 $D(Q) \subset B^+$ 和 $D(Q) \cap B^- \ne \emptyset$ 的 $P \in F$ 与 $\vec{m} \in \mathbb{Z}^d$ 的模式集合 $Q = \vec{m} \cdot P$。对于 v_1、$v_2 \in V'$ 和 $S \in A^{D \cdot R}$，当且仅当满足以下条件时，存在一条从 v_1 到 v_2、标签为 S 的边：

$$v_2 = \{-\vec{e}_1 \cdot Q \mid Q \in v_1 \cup v_0, \nexists n \in D(Q) \cap D(S) : Q_n \ne S_n\}.$$

这些定义背后的直观理解如下。节点 $v_1 \in V'$表示 X_K 的一个部分定义构型，其中我们指定了 $V \setminus B^+$的内容。带有标签 S、指向 v_2 的边，表示将构型向“左”移位的作用

by $\vec{e}_1$，并在移位版本中指定无限带 $B^+ \cap B^-$ 的内容。由于该构型是 K-周期的，只需指定 $D \cdot R$的内容，而这恰好是S的定义域。因此，图 G' 中的双向无限游走确定了一个完整的 K-周期构型。由于 X 是由 F定义的有限型子移位，只需跟踪定义域与 $B^+ \cap B^-$相交的 $P \in F$的平移版本。图案会随整个构型一起向左移位。在被“处理”之前，它们不能离开 B^+；所谓“处理”，是指它们与 $B^+ \cap B^-$中的某个元素上的构型不一致，否则它们就会出现在构型中已完全指定的部分。

总而言之，我们计算出了一个有限有向图 $G' = (V', E')$，其边由 $A^{D \cdot R}$ 中的图案标记。如果将每个标签 S 替换为其密度$\sum_{v \in D(S)} W(v)/|D(S)| \in \mathbb{R}$，那么最小权重 $W(X_K)$ 等于 G' 上双无限游走的最小权重。后者又由一个简单环 [16, 引理 4.1], 达到，因此只需计算 G' 中简单环的最小密度。为此，我们使用 Karp 最小平均环算法 [7]，该算法很容易并行化。

在应用 Karp 算法之前，我们进行了额外的优化，以减小由 $\mathbb{R}$ 标记的图 G' 的规模。具体来说，我们反复执行以下操作，直到图不再发生变化：

1. 使用经过修改的 Hopcroft 算法对确定有限自动机最小化 [5], 简化图，从而得到一个可能更小的图；该图具有相同的有限游走（因而也包括双无限游走）的标签集合。[1]2. 对于每一对顶点 v、$v' \in V'$，从 v 到 v' 的所有边中，除权重最小的边外，删除其余边。

最小化不会改变图上双向无限游走的标签集合，第二个操作也不会改变此类游走的最小密度，因此就我们的目的而言，它们是安全的。

我们已经利用此功能，在无限六边形网格上找到了一个密度为 $53/126$ [16] 的新识别码，低于此前的纪录 $3/7$ [3]。要重现该识别码，请按照第 4 节中的方式定义六边形网格和识别码的有限型子移位，然后运行以下命令

```
%计算_禁止_图案 idcode 3 %最小_密度idcode
线程=3 (5,1)
```

这里，threads=3 是一个可选参数，用于指定线程数（有些命令接受可选参数，但 %minimum_density 目前是唯一使用并行计算的命令）。在笔记本电脑上，该计算大约需要 17 分钟。

7.2 通过放电求下界

接下来，我们将介绍计算 $W(X)$ 下界的算法。该算法不依赖于周期点的存在。只要有足够的

[1]请注意，从 [9, Definition 3.3.1]，的意义上说，我们的图并不是右解析的，这相当于使用确定有限自动机而不是非确定有限自动机。因此，就具有相同游走标签集合的所有图中顶点数绝对最少这一意义而言，该结果可能不是最小的。不过，该算法从不会增加顶点数，并且在实践中通常可以显著减少顶点数。

计算资源，它就能给出 $W(X)$ 的任意精度近似值。

其理论背景是图论中广泛使用的放电论证 [4]。我们将在当前语境下加以描述。其思路是利用确定性的局部规则重新分配节点的权重，并通过有限论证证明，所得的权重构型处处都至少为某个常数 α，从而推出 $W(X) \geq \alpha$。构型 $x \in A^V$的初始荷是函数 $C_x\colon \mathbb{Z}^d \to \mathbb{R}$，定义为 $C_x(\vec{n}) = \sum_{r\in R} W(x_{\vec{n}\cdot r})/|R|$。给定拓扑 $G = (V, E)$、代表集 $R \subset V$ 和字母表 A，一个放电规则是三元组 $(P, \vec{m}, c)$，其中 $P \in A^D$ 是有限模式，$\vec{m} \in \mathbb{Z}^d$且 $c \in \mathbb{R}$。有限集 $\mathcal{D}$ 中的放电规则定义了一个新的荷 $\mathcal{D}(C_x)$，其定义如下：

$$\mathcal{D}(C_x)(\vec{n}) = C_x(\vec{n}) + \left(\sum_{\substack{(P,\vec{m},c)\in\mathcal{D}\\((\vec{n}-\vec{m})\cdot x)|_{D(P)}=P}} c\right) - \left(\sum_{\substack{(P,\vec{m},c)\in\mathcal{D}\\(\vec{n}\cdot x)|_{D(P)}=P}} c\right). \qquad (1)$$

若 $X \subset A^V$ 是一个 G-子移位，且对所有 $x \in X$ 和 $\vec{n} \in \mathbb{Z}^d$，$\mathcal{D}(C_x)(\vec{n}) \geq \alpha$，那么 $W(X) \geq \alpha$。要证明 $\mathcal{D}(C_x)(\vec{n}) \geq \alpha$ 始终成立，只需考虑 X中出现的每个模式 $Q \in A^N$，其中 $N \subset V$是有限的，但足够大，能够包含 P的定义域以及所有 $(P, \vec{m}, c) \in \mathcal{D}$对应的 $-\vec{m}\cdot P$，并证明当 $x|_N = Q$时，它们都满足 $\mathcal{D}(C_x)(\vec{0}) \geq \alpha$。这是一个有限计算。

我们采用 Stolee 的思路 [17], ，并使用线性规划自动寻找一组合适的放电规则。首先，固定一个有限定义域 $D \subset V$ 和一个有限集向量 $T \subset \mathbb{Z}^d$。通过枚举某个更大有限定义域上的所有局部允许模式，生成一组图案 $\mathcal{P} \subset A^D$，其中包含所有在 X中出现的形状为 D 的模式。对于所有 $P \in \mathcal{P}$ 和 $\vec{m} \in T$，我们的三元组为（$P, \vec{m}, c(P, \vec{m})$），其中 $c(P, \vec{m}) \in \mathbb{R}$ 是线性规划中的无约束变量。我们还添加另一个变量 $\alpha \in \mathbb{R}$。

接下来，在定义域 $D' = \bigcup_{\vec{m}\in T\cup\{\vec{0}\}} -\vec{m}\cdot D$上生成另一组图案 $\mathcal{Q} \subset A^{D'}$，其中包含所有在 X中出现的 D'形模式。每个模式 $Q \in \mathcal{Q}$ 都会像（1）一样为线性规划定义一个约束：α 的值必须小于或等于 $C_Q(\vec{0})$之和，其中包括所有满足（$-\vec{m}\cdot Q$）$|_D = P$的 $c(P, \vec{m})$，以及所有满足 $Q|_D = P$的 $-c(P, \vec{m})$。任何满足这些约束的 α和 $c(P, \vec{m})$的取值，都能给出 $W(X)$的有效下界。因此，我们应在这些约束下最大化 α 的值。

线性规划的一个解，就是一种最优放电策略：根据图案 $\mathcal{P}$的出现情况，沿着向量 T 转移电荷；前提是有限型子移位 X 可能包含任意一种图案 $\mathcal{Q}$。增加定义域 D 中的节点数量、增加 T中的向量数量，或细化集合 $\mathcal{P}$ 或 $\mathcal{Q}$，通常都能得到更好的界，但计算成本也会相应增加。对应的 Diddy 命令是 %density_lower_bound radius domain vectors。例如，我们可以通过命令 %density_lower_bound idcode 0 (0,0,0) (0,0,1) (-1,0,1) (0,1,1) (0,-1,0) (1,0,0); (0,-1) (0,1) (1,0)，计算六边形网格上识别码密度的下界。大约 65 秒后，Diddy 得出了 0.4 这一界，与提出该问题的文章 [8] 中的结果相同。此后，这一结果已大幅改进，最近达到 23/55[17]。

8 可视化

Diddy 还支持交互式生成并显示定义在类 $\mathbb{Z}^2$图上的有限型移位的局部有效有限图案。命令 %tiler name 会打开一个新窗口，其中包含拓扑 G的有限网格状子图。用户可以缩放和平移视图、设置任意节点的值，并要求 Diddy 为其余节点赋值，使该图案在有限型移位 name 中局部有效（如果可能）。我们使用 Pygame 库 [10] 进行可视化。

9 未来方向

以下列出作者计划实现的部分功能。

- 支持不同有限型子移位之间的一般块映射。
- 子移位有限型比较算法的条件 2 可以扩展到例如最终周期、半线性或自动配置。
- 更好地支持块映射、图案和有限型移位之间的相互作用。例如，可以使用 SAT 求解器搜索不变子移位有限型、时间周期点或元胞自动机的吸引子。
- 可以扩展语法，让各种有限型子移位和元胞自动机更易于实现。这包括图中的距离计算（例如识别任意半径的码）、多层有限型子移位和元胞自动机（例如 Robinson 铺砖集 [11]），以及涉及计数或算术的定义（例如在 $\mathbb{Z}_p$ 上的线性元胞自动机，或康威生命游戏）。
- 许多在二维及更高维有限型子移位中不可判定的问题，在一维情形下可以借助有限自动机理论和线性代数有效求解。参见例如 [9, 第 3.4 节和第 4 节 3]。Diddy 应为这一特殊情形提供全算法。
- 二维有限型子移位的迹 是其构型中出现的无限列集合。它可以视为一个一维子移位，并可通过 sofic 子移位从上方逼近（有时可精确表示）。Diddy 应提供提取和分析多维有限型子移位的迹的功能。

致谢

Ilkka Törmä 获得芬兰科学院的资助，资助项目为 346566。

参考文献

[1]Gilles Audemard 和 Laurent Simon。Glucose 4.1，2016。网址：https://www.labri.fr/perso/lsimon/research/glucose/.[2] Robert Berger。《多米诺问题的不可判定性》。美国数学学会回忆录 *No.*，66，1966。72 页。

[3]Gérard D. Cohen、Iiro Honkala、Antoine Lobstein 和 Gilles Zémor。《六边形网格中顶点识别码的界》。*SIAM* 离散数学杂志，13(4):492–504，2000。doi:10.1137/S0895480199360990。

[4] Daniel W. Cranston 和 Douglas B. West。《通过图着色介绍消去法》。离散数学，340(4):766–793，2017。doi:10.1016/j.disc.2016.11.022。[5] John Hopcroft。一种用于最小化有限自动机状态的 n 对数 n 算法。载于机器与计算理论（*1971* 年海法理工学院国际研讨会论文集），第 189–196 页。Academic Press，纽约，1971。[6] Alexey Ignatiev、Antonio Morgado 和 Joao Marques-Silva。PySAT：用于使用布尔可满足性问题预言机进行原型设计的 Python 工具包。载于 布尔可满足性问题，第 428–437 页，2018。doi:10.1007/978-3-319-94144-8_26。[7]Richard M. Karp。《有向图中最小环路均值的刻画》。离散数学，23(3):309–311，1978。doi:https://doi.org/10.1016/0012-365X(78)90011-0。[8] Mark G. Karpovsky、Krishnendu Chakrabarty 和 Lev B. Levitin。《一种用于识别图中顶点的新型编码》。*IEEE* 信息论汇刊，44(2):599–611，1998。doi:10.1109/18.661507。[9]Douglas Lind 和 Brian Marcus。符号动力学与编码导论。剑桥大学出版社，剑桥，1995。doi:10.1017/CBO9780511626302。[10]Pygame 开发团队。Pygame 2.3.0，2023。网址：https://github.com/pygame。[11] Raphael M. Robinson。《平面铺砖的不可判定性与非周期性》。*Invent. Math.*，12:177–209，1971。[12]J.S. Roy、Stuart A. Mitchell、Christophe-Marie Duquesne 和 Franco Peschiera。PuLP 2.7.0，2022。网址：https://github.com/coin-or/pulp。[13] Ville Salo 和 Ilkka Törmä。《预言机能告诉我们生命的最终命运什么？》。载于第 *49* 届 *EATCS* 自动机、语言与编程国际会议，第 229 卷，*LIPIcs. Leibniz* 国际信息学会议论文集，论文编号 131，共 20 页。Schloss Dagstuhl。Leibniz 信息学中心，Wadern，2022。doi:10.4230/lipics.icalp.2022.131。[14] Ville Salo 和 Ilkka Törmä。《生命游戏中的伊甸园》。载于自动机与复杂性——为庆祝 *Eric Goles 70* 岁生日而作的论文集，第 42 卷，新兴复杂性计算，第 399–415 页。Springer，Cham， [2022]©2022。doi:10.1007/978-3-030-92551-2_22。[15]Ville Salo 和 Ilkka Törmä。Diddy，2023。网址：https://github.com/ilkka-torma/diddy。[16] Ville Salo 和 Ilkka Törmä。《自动在无限网格上寻找编码》，2023。arXiv:2303.00557。[17] Derrick Stolee。《针对网格密度问题的自动消去论证》，2014。arXiv:1409.5922。[18] Hao Wang。《通过模式识别证明定理 II》。*Bell System* 技术期刊，40:1–42，1961。