arXiv:2512.23143v1 [math.HO] 29 Dec 2025

x玩宝可梦，几乎对所有 x。

C. Evans Hedges

Abstract. 本文简要说明，对于任意有限状态游戏，析取数 x 最终都会赢得该博弈。这里采用的证明技巧广为人知，而这一结论可直接由自动机理论中的经典结果推出。这篇短文主要是一篇阐述性文章，旨在汇集这一证明，并以 π 玩宝可´梦的趣味背景作为动机。

1. Introduction

2021年10月，Twitch 频道 $WinningSequence$ 开始直播 π 玩宝可´梦：每个数字 $0-9$ 都映射到一个 GameBoy Advance 按钮，随后 π 以每秒一个数字的速度开始玩宝可´梦蓝宝石。截至本文撰写时，$WinningSequence$ 已经运行了略多于 π 年，按下了超过一亿个按钮；尽管从未离开起始区域，却已经拥有一只 90 级的蜥蜴王。这自然引出了一个问题：π 最终能赢得宝可´梦吗?

虽然我们怀疑 π 是一个正规数 [1], ，但我们不知道事实是否如此。遗憾的是，由于我们对 π 的性质知之甚少，目前无法回答这个问题。不过，我们能够证明，对于任意一个析取数 x [2] （也就是勒贝格几乎处处的数 [1]），x 最终都会赢得宝可梦´。该定理适用于任意有限状态博弈，其中任何时刻都可以采取的行动数量有限。它也适用于使用确定性伪随机数生成器的博弈，并且对于使用真随机数生成器的博弈，该结论以概率 1 成立。这个结果源自自动机理论中的一个著名结论 [3, 8] 。本文旨在以一种有趣的方式介绍如何运用图论和自动机理论来建模我们可能感兴趣的博弈。

2. Background

为了对这些有限状态博弈建模（由于内存限制，所有计算机游戏都必然属于此类），我们从一组可能的游戏状态 V 开始。如果游戏还使用一个由单个生成种子初始化的伪随机数生成器（pRNG），我们也可以将 V 该伪随机数生成器的当前状态纳入其中。因此，给定游戏的某个状态 $v \in V$，对于一次给定的按键操作 b，v 将确定性地转移到某个 $v' \in V$。于是，我们称从 v 到 v' 存在一条标记为 b的有向边。令 $\mathcal{E}$ 表示给定游戏的所有边的集合。于是，我们称游戏 G 就是图 $G = (V, \mathcal{E})$。从这个意义上说，游戏包含了有关游戏状态的信息，以及状态之间可能发生的转移。

请注意，这类博弈是确定性的，因为我们已将 pRNG 的当前状态嵌入 V 中。如果没有这样做，我们就必须纳入多个具有相同标签的转移，并将其与（p）RNG 的输出结合起来。不过，本文采用简化情形，即 V 包含 pRNG 的状态以及我们的游戏状态

Date：2025年6月16日。

因此，可以认为转移是确定性的。我们在此指出，经典的 Pok´emon 游戏就是这种情况，包括 Pok´emon Sapphire；该游戏使用周期为 2^{32}的线性同余生成器（LCG） [5] 。因此，状态空间 V 实际上是游戏内存状态与 LCG 的 2^{32} 个可能状态的笛卡尔积。

我们还称图 G 中的路径为边序列 $e_1e_2\dots e_k$，这些边依次经过一系列顶点 $v_0v_1\dots v_k$，其中边 e_i 是从 v_{i-1} 到 v_i 的一条边。对于上述给定的图 G ，如果一个词 w 满足以下条件，就称它是同步的 [3] ：对于任意两个起始顶点 v_1， $v_2 \in V$，沿着由 w 生成的边路径前进（即按照 w 中的标签为边标记），最终到达的顶点相同。也就是说， $v_1 \mapsto_w v$ 和 $v_2 \mapsto_w v$。

最后，如果一个顶点 $v \in V$ 满足：每条离开 v 的边 $e \in \mathcal{E}$ 都将 v 映射到 v，那么它就称为汇点状态。这意味着一旦到达 v，博弈之后的所有状态都将是 v，博弈实际上就陷入了停滞。在游戏领域，这通常称为“硬锁定”。在本文余下部分，我们假设 G 有一个唯一的汇点状态 $\tilde{v}$ ，且对于任意 $v \in V$ ，都存在一条从 v 到 $\tilde{v}$ 的路径。就博弈而言，我们希望 $\tilde{v}$ 表示你已经赢得了博弈；而要求任意 $v \in V$ 都有一条通往 $\tilde{v}$ 的路径，意味着无论博弈当前处于什么状态，仍然都有可能获胜。这意味着我们假设该博弈是“可化归至获胜”的——你无法把自己硬锁定在一个再也不可能获胜的状态中。

最后，我们来考虑如何生成按钮按压操作来进行博弈。对于给定的博弈 G，我们有有限个可能的操作 b 。我们将这些操作表示为字母表 $\mathcal{B} = \{0, \dots, b-1\}$中的符号。因此，我们可以取任意实数 $x \in \mathbb{R}$，并考察 x的 b 进制展开。本文的一个重要概念是析取数。

定义 2.1. 我们称 $x \in \mathbb{R}$ 在 b进制下是析取的，如果对于任意有限词 $w \in \{0, \dots, b-1\}^n$，词 w 都会出现在 x的 b 进制展开中。若 x 在每个进制 $b \geq 2$下都是析取的，则称其为析取数。

请注意，勒贝格几乎处处的实数都是正规数 [1], ，这是一个强得多的性质，它要求每个有限词都以适当的频率出现。此处我们不作形式化定义，因为这并非必要；我们只需指出，几乎每个数都是析取数 [2]。

3. Statement of Theorems

定理 3.1. 设 G 是如上所述的博弈（图）。那么存在一个同步词 w [3] ，可将所有顶点映射到 $\tilde{v}$，其长度的上界为 $|G|^2$ [7]。

注记 3.2. 二次上界 $|G|^2$ 是一个标准结果 [7], ，但值得注意的是它与著名的 *Černý* 猜想 [3] 之间的联系。该猜想指出，对于任何具有 n 个状态的同步自动机，都存在一个长度至多为 $(n-1)^2$ [8]的同步词。尽管这仍是自动机理论中最重要的未解问题之一，但对我们的目的而言，存在某个多项式上界就足够了。

证明。 首先，我们枚举状态 $V = \{\tilde{v}, v_1, \dots, v_N\}$，其中 $N = |V| - 1$。对于每个 $1 \leq i \leq N$，都存在一个获胜词 w_i ，将 v_i 映射到 $\tilde{v}$。我们通过归纳法构造同步词 W 。

设定 $W_1 = w_1$。应用 W_1后，我们观察到 v_1 被映射到 $\tilde{v}$。由于 $\tilde{v}$ 是汇点状态，任何后续输入都会使博弈保持在状态 $\tilde{v}$。现在考虑 v_2。应用 W_1后，状态 v_2 被映射到某个 $v_{i_2} \in V$。设 w_{i_2} 为满足以下条件的词：

将 v_{i_2} 映射到 $\tilde{v}$。然后我们设定 $W_2 = W_1 w_{i_2}$。根据构造， W_2 将 v_1 和 v_2 都映射到 $\tilde{v}$。对所有 N 个顶点继续这一过程，我们得到一个有限词 $W = W_N$，使得对于每个起始顶点 $v \in V$，应用 W 后游戏状态为 $\tilde{v}$。

最后，我们注意到，每个 w_i 的长度都可以选得不超过 $|V|$。因此， W 的长度上界为 $N|V| \leq |V|^2$。□

注 3.3。虽然定理 *3.1* 保证了获胜词的存在，但要找到最短的获胜词却困难得多。在计算复杂性领域，已知对于给定的自动机，寻找最短同步词是 ***NP*** 困难的[4, 6]。因此，尽管 x 最终会获胜，速通玩家（或特别缺乏耐心的实数）仍会发现，优化这一获胜序列在计算上是难以处理的。

定理 3.4。设 x 是析取的。那么， x 将在任何无法输掉的确定性游戏中获胜。

证明。这由定理 3.1 立即得出。令 W 为定理 3.1 所保证存在的同步词。由于 W 会出现在 x 中，并且我们知道，无论博弈当前处于什么状态， W 都会获胜，因此在看到 W之前， x 采取什么行动并不重要；完成 W 后，我们就必定能赢得这场博弈。□

观察 3.5. 如果一个博弈有 b 个按钮可按，那么就有 $b^{|G|^2}$ 个长度为$|G|^2$的序列。将这些序列全部串接起来后，我们就能在按下按钮 $|G|^2 b^{|G|^2}$次后赢得博弈。

对于现实情况来说，这个上界糟糕透顶，因为 Pok´emon Sapphire 的存档状态有 1MB，这意味着我们可以将 $|G|$ 限制在 8×10^6以内。由于它还有 8 个按钮，我们可以在

$$(8 \times 10^6)^2 \times 8^{(8\times 10^6)^2} = 2^6 \times 10^{12} \times 2^{3\times 2^6 \times 10^{12}} \approx 2^{10^{14}}$$

次按键操作内赢得 Pok ≈ 30 emon Sapphire。这是一个糟糕透顶的上界；这个数字有 ≈ 30 万亿位，而可观测宇宙中大约有 10^{80} 个原子，因此这个数字实际上比它大$10^{30 \text{ trillion}}$ 倍。与可观测宇宙中的普朗克体积数量相比，它也同样巨大。这个上界糟糕得令人震惊，尽管我们可以相对轻松地在这里写出 $2^{10^{14}}$，但就任何实际用途而言，它都是一个大得离谱的数字。

此外，我们在此指出，如果我们对博弈 G进行适当编码，使其能够独立支持随机数生成器，那么只要随机数生成器以某个正概率显示每个词，每个析取数 x 都将以概率 1 获胜 G （相对于随机数生成器）。

定理 3.6. 设 G 是一个带有随机数生成器的博弈。设该随机数生成器在所有词的空间上具有满支撑。若 x 是析取的，则 x 将以1的概率赢得博弈 G （相对于随机数生成器）。

证明。令 W 为定理 3.1 中的同步词，并令 R 为所需的随机数生成器输出序列，使得可以保证（W， R）获胜。由于随机数生成器对所有有限词都具有满支撑，因此概率 $P(R) > 0$；又由于 x 是析取的，我们知道 W 会无限次出现。因此，对于 W的每次出现， x 届时不会赢得博弈的概率为（$1 - P(R)$) < 1。由于 W无限次出现，我们知道在极限情况下， x 不会获胜的概率为 $\lim_{n\to\infty}(1 - P(R))^n = 0$。因此，以概率 1， x将赢得博弈。□

4. Concluding Remarks

我们已经证明，对于几乎所有实数 x， x 的以 b 为底的展开中都包含一串按钮操作，能够赢得任何确定性可胜性归约游戏。由于析取数包含每个有限字符串，它也必然包含每一种工具辅助速通（TAS）的输入序列，包括逐帧精准的最优通关（假设按钮以适当的速度按下）。

然而，正如观察3.1中的界所示，这一结果纯粹是存在性结论。对于像宝可梦蓝宝石这样复杂的博弈，所保证的获胜序列长度极其庞大，远远超过可观测宇宙中的普朗克体积数。虽然析取数 x 最终会获胜，但几乎可以肯定，在到达获胜数字之前，运行游戏的物理硬件早已在宇宙热寂中停止运作。因此，尽管 π 很可能包含一局获胜的宝可梦游戏流程，可悲的是，宇宙中很可能没有足够的时间让我们亲眼见证。

References

[1] ´埃米尔·博雷尔。《可数概率及其算术应用》。巴勒莫数学学会报告，27:247–271，1909。[2] Cristian S. Calude 和 Tudor Zamfirescu。析取序列：一种概率方法。离散数学，193(1–3):45–53，1998。[3] J´an ˇCern´y。《关于齐次有限自动机实验的注记》。Matematicko-fyzik´alny ˇcasopis，14(3):208–216，1964。[4] David Eppstein。单调自动机的重置序列。SIAM计算杂志，19 510，1990 3 500( ): – 。[5] Donald E. Knuth。《计算机程序设计艺术》第2卷：半数值算法。Addison-Wesley，马萨诸塞州雷丁，第3版，1997。[6] Jan Olschewski 和 Michael Ummels。有限自动机中寻找重置词的复杂性。载于《计算机科学数学基础》（MFCS），计算机科学讲义第6281卷，第568–579页。施普林格，2010。[7] Jean-´Eric Pin。自动机理论中产生的两个组合问题。离散数学年刊，17:535–548，1983。[8] Mikhail V Volkov。同步自动机与ˇcern´y猜想。《语言与自动机理论及应用》，第11–27页，2008。