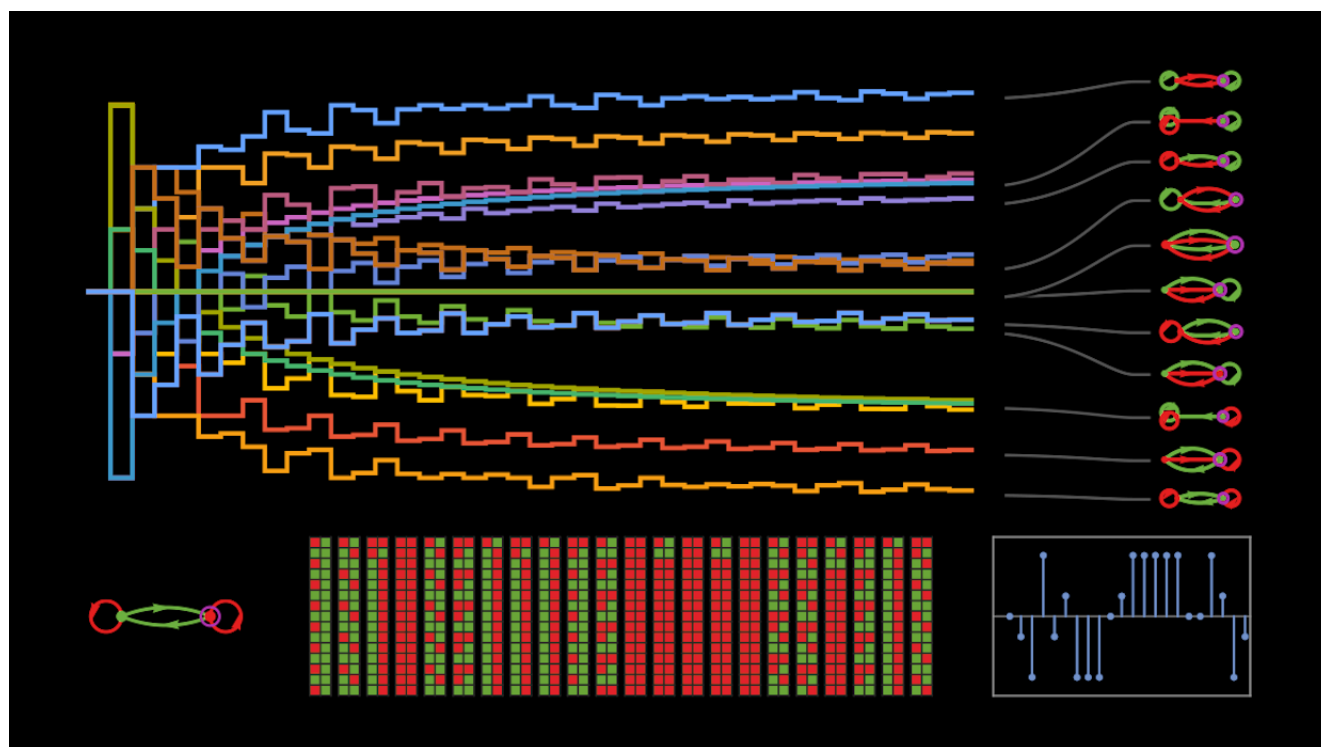


程序之间的博弈：竞争的规则学

June 4, 2026

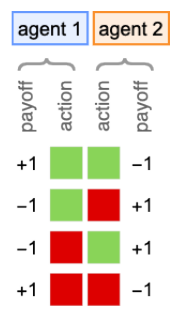


基本设置

无论是生物学、经济学、政治学，还是其他众多领域，我们常常会遇到可以用两个不断相互竞争的智能体来建模的情况。可以想象，在每一步中，每个智能体都能从一组特定行动中选取一个，然后——按照经典的博弈论方式——每个智能体（或“玩家”）会根据自己和对手采取的行动，获得某个固定的“收益”。但智能体如何决定该采取什么行动呢？我们设想每个智能体都有一个固定的决策程序（或“策略”）。并且，我们设想这些决策的输入是智能体及其对手过去一系列行动的序列。

在过去将近一个世纪里，人们针对特定策略的选择进行了大量研究。但我长久以来一直好奇的是，如果系统地考虑所有可能的策略会发生什么。而如果把策略视为程序，这就变成了一个我们可以立即应用规则研究方法的问题。这正是我接下来要做的。

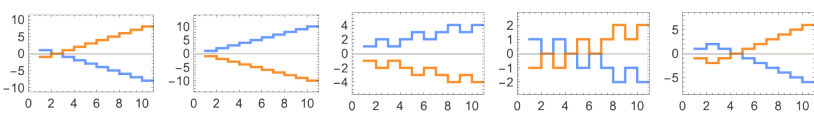
为了更具体地说明设定，我们假设在每一步中，每个参与者采取两种可能行动之一，由 ■ 和 ■ 表示。目前我们采用经典“匹配与否”（“匹配硬币”）博弈的收益——其中玩家 1 在匹配时获得更大收益，而玩家 2 在不匹配时获得更大收益：



那么当参与者反复进行这个博弈时会发生什么？嗯，这取决于他们的策略。以下是几个不同策略选择的例子：

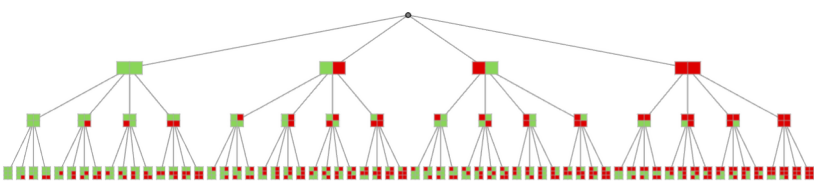


绘制每种情况下两个参与者（由 ■ 和 ■ 表示）的累积收益，我们得到：

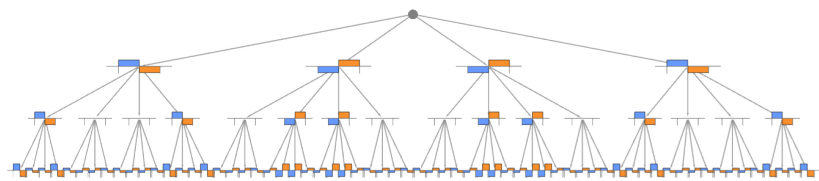


通常，我们会将经过一定步骤后累计收益数值最大（即在这些图中最终处于顶部）的“获胜智能体”视为赢家。基于这样的标准，我们能够对不同程序进行排名，并以此为基础探索竞争规则学。

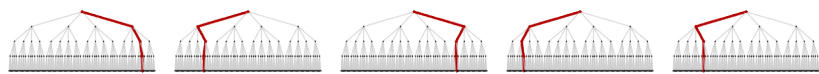
利用我们采用的基本设置，可以通过多路图表示所有可能的行动序列：



对于任何给定的行为序列，在我们的匹配或不匹配游戏中，每个智能体都有一个累积收益：



如果每个代理都采用特定策略，那么这将在多路图中定义一条特定的路径。对于上述示例中使用的策略，路径为：



拥有一个必胜策略需要什么？接下来，我们将考虑基于几种不同类型程序的策略。但我们始终可以追问的一个基本问题是：最终成为获胜策略的那些策略，究竟是基于更复杂的程序，还是更简单的程序——或者说，它们展现的行为是更复杂，还是更简单。

换句话说，如果你想要获胜，通常应该努力构建复杂的事物，还是更应期待找到一种“简单的技巧”来“破解游戏”，从而——至少通常情况下——让你获胜？实际上，我们是在问：竞争究竟是倾向于导致复杂性，还是简单性？

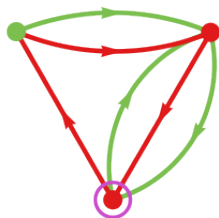
我最近研究了生物进化和机器学习的极小模型，在这些模型中，程序通过适应性演化来最大化某种外部设定的适应度函数。我发现，即便所使用的适应度函数非常简单，最大化该函数的程序行为通常也相当复杂。换句话说，适应性演化往往会让一个简单的固定目标以复杂的方式实现。

那么，如果我们不设定固定、外部强加的目标，而仅仅是在广义上要战胜其他智能体，会怎样呢？这种可能无休止的竞争，是否会促使我们产生更复杂的行为（或更复杂的程序）？这正是我们通过研究竞争规则学所要探讨的问题。

有限状态机策略

有限状态机可以被视为定义了极其简单的程序（可能用于模拟生物学中的通路、经济学中的决策过程等）。为了开始研究竞争规则学，我们将考察由有限状态机定义的策略。

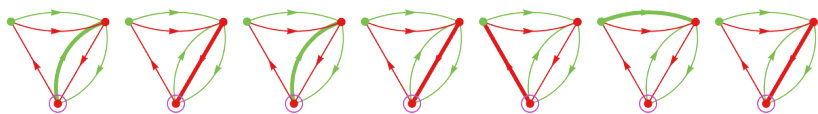
一个典型的有限状态机示例（此处为 3 个状态）如下：



我们将使用这个有限状态机来定义某个智能体的策略。为了理解其运作方式，假设该智能体的对手所采取的行动序列为：

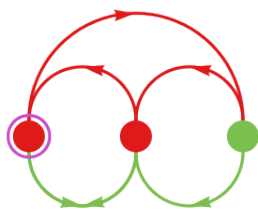
■, ■, ■, ■, ■, ■, ■

思路是利用这一系列动作在有限状态机图中定义一条路径，然后根据所到达状态的颜色确定下一步动作。我们从带有输入箭头（箭头指向）的顶点开始，随后依次沿着颜色与对手下一步移动相匹配的边前进：

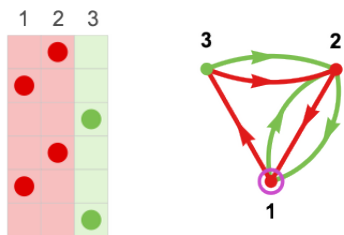


在此过程结束时，我们将到达图中的某个顶点（即有限状态机中的某个状态）。以这里展示的具体情况为例，我们最终抵达的状态是 ■。接着，我们取策略的输出——即智能体应采取的下一步行动——为 ■。

有时将有限状态机的状态排列在一条线上会很方便：

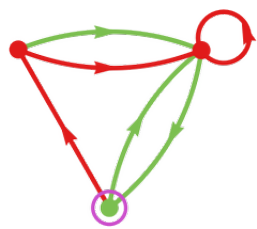


然后我们可以通过展示连续达到的状态来总结在特定输入下的路径：

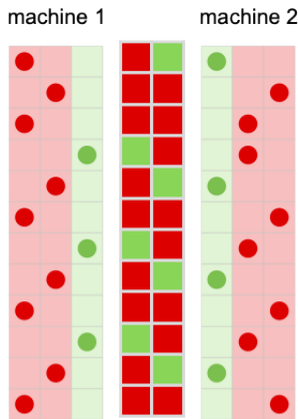


那么当两个有限状态机相互竞争时会发生什么？基本思想是，一个机器的连续输出成为另一个机器

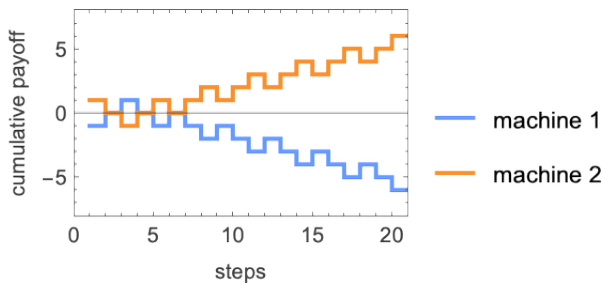
的连续输入，反之亦然。如果我们的第二个机器是



那么我们可以通过以下方式表示这两个机器的行为：



如果我们使用的收益是针对匹配或不匹配游戏的，那么这些机器的累计值就是



因此最终智能体 2 可以被视为获胜者。

需要注意的是，在我们使用的设定中，一切都是确定性的：每一步，每个智能体都会根据其策略，从过去的移动历史中确定性计算出的行动。这与博弈论中通常研究的情况不同，在博弈论中，每一步实际上被视为独立，但不同行动可能存在概率（“混合策略”），最终会对“不同的骰子结果”进行平均。

可能的有限状态机空间

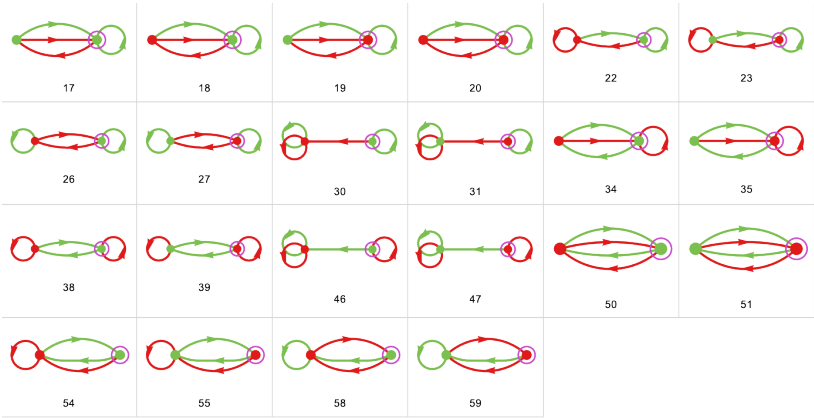
对于具有 s 个状态的有限状态机，可能的图的数量是 $(2s^2)^s$ 。但其中一些图对应行为相同的机器

——因此不同机器的数量更小：

states	total	canonical	distinct
1	2	2	2
2	64	47	22
3	5832	1728	956
4	1 048 576	83 968	51 924
5	312 500 000	5 141 600	341 5990

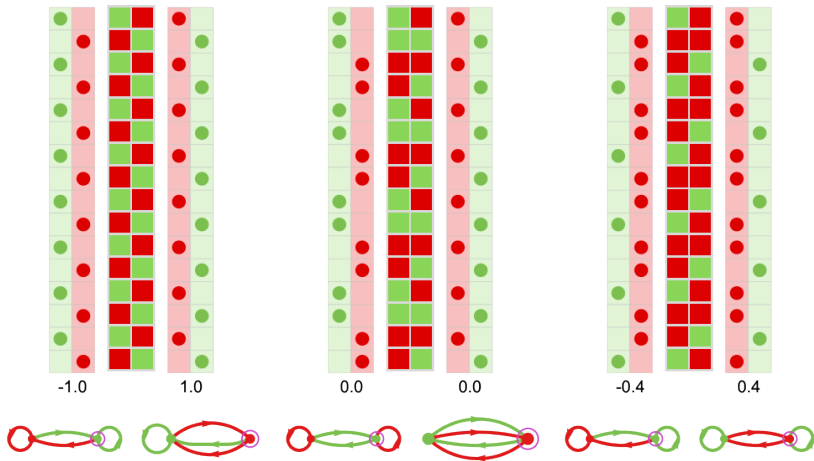
二态机器

在二态的情况下，有 22 个不同的机器



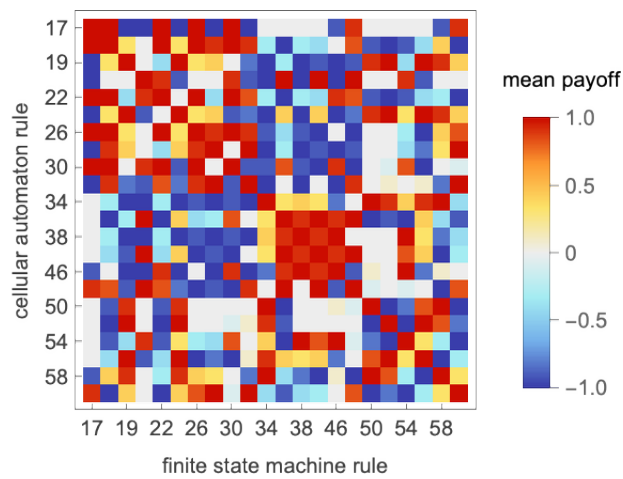
我们通过编号来标识每台机器。

那么，当这些机器成对竞争时会发生什么？以下是一些例子，每个例子中我们都在识别平均收益（这里是指进行 10 轮“匹配或不匹配”游戏后的平均收益）：

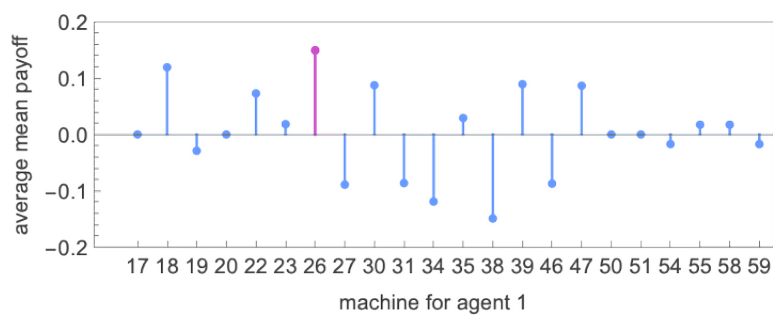


(在所有有限状态机之间的竞争中，移动序列最终必然变得周期性——周期最多等于每台机器状态数的乘积。)

如果 22 个不同的 2 状态机中的每一个与其他每一个进行竞争，会发生什么？我们可以通过展示每对机器的平均（长期）收益来总结结果（这里的收益是每台机器“作为智能体 1”时的收益；在“匹配与否”游戏中，如果“作为智能体 2”，则收益取反）：



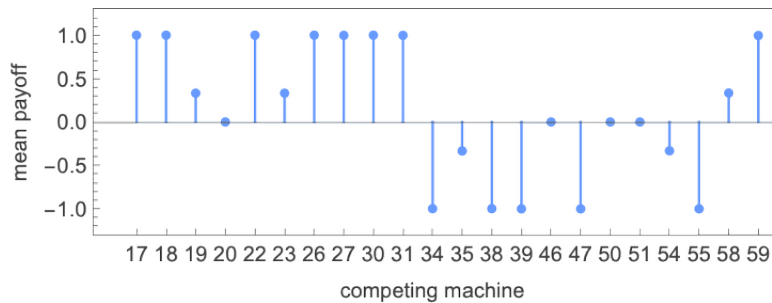
那么，哪台机器是“总冠军”呢？一种评估方法是看某台机器与所有其他（不同）机器竞争时获得的平均收益的均值：



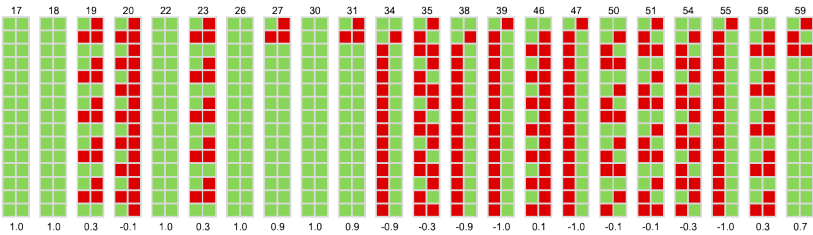
按此度量，优胜者是机器 26：



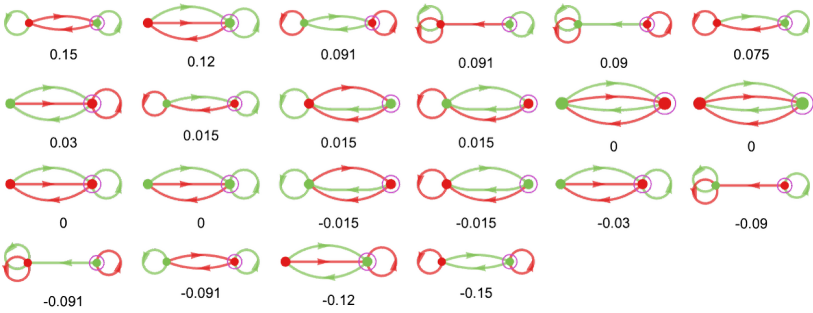
将此机器与所有（不同的）2 状态机器对弈，我们得到以下平均收益：



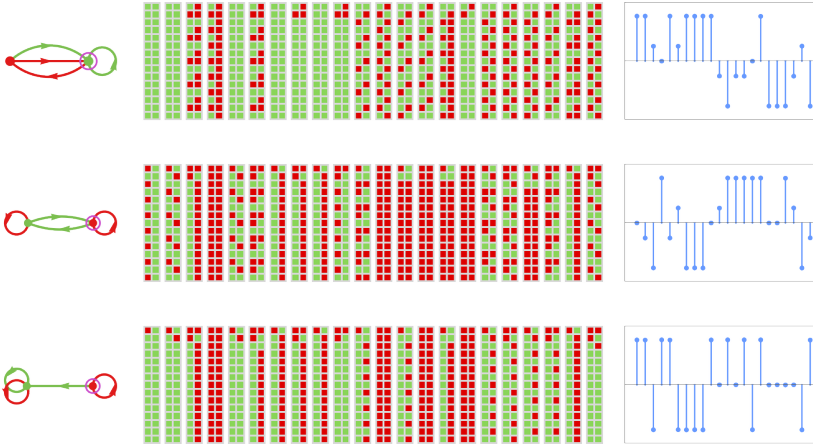
每种情况的实际行为（本身不依赖于收益，仅取决于所涉及的机器）如下：



获胜机器的“亚军”有哪些？ 以下是所有不同的机器，按平均收益排序：

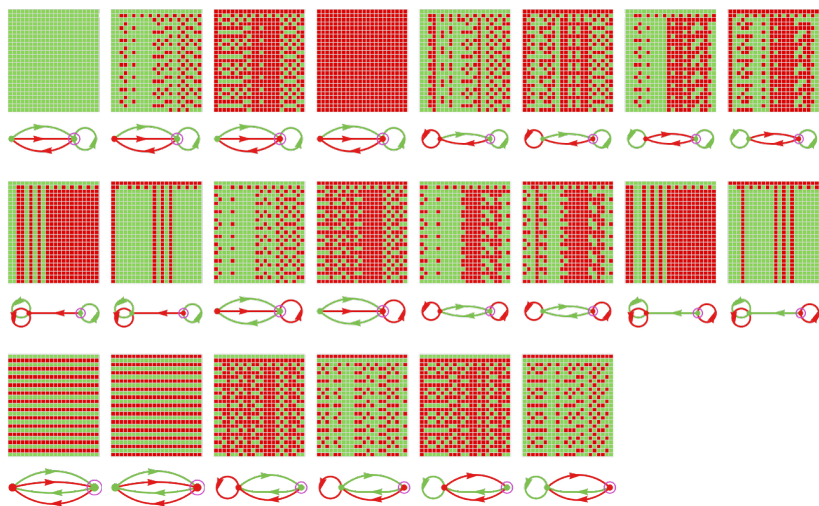


如果我们将前三名亚军与所有机器对弈， 结果如下：



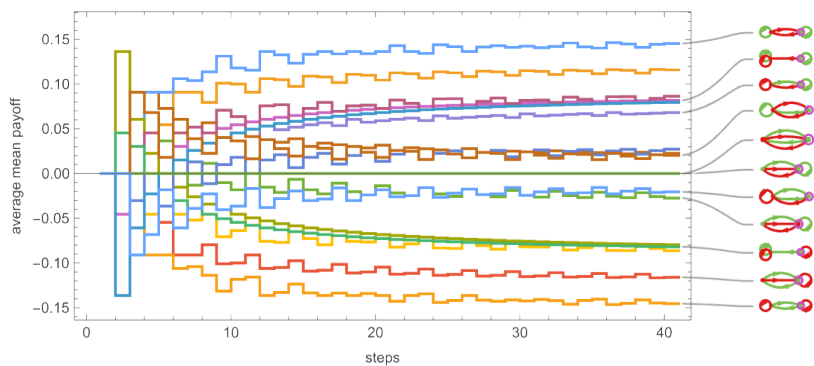
我们可以通过展示机器在与所有其他机器对弈时的行为历史（或者实际上，通过将上面这类图片的第一列组合起来）来总结机器行为。以下是所有机器（进行 15 步）的结果，按平均得分从高到低

排列：

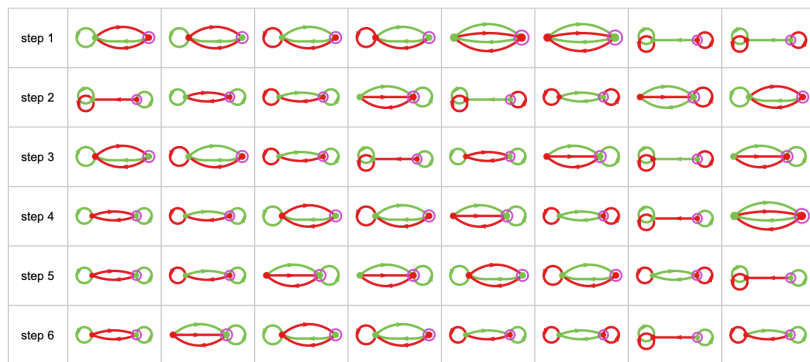


(再次强调，这些图片完全由所涉及的机器决定；"匹配或不匹配"游戏中的收益仅决定了它们的排序。)

我们这里所讨论的内容有一个脚注，涉及让机器进行多少步竞争的问题。对于所有有限状态机，其行为最终必然会变得周期性——对于 2 状态机，最大周期为 4 步，最大瞬态长度为 3 步。但实际的平均收益会根据所考虑的步数而变化：



值得注意的是，至少在最初几步中，排名会发生变化：

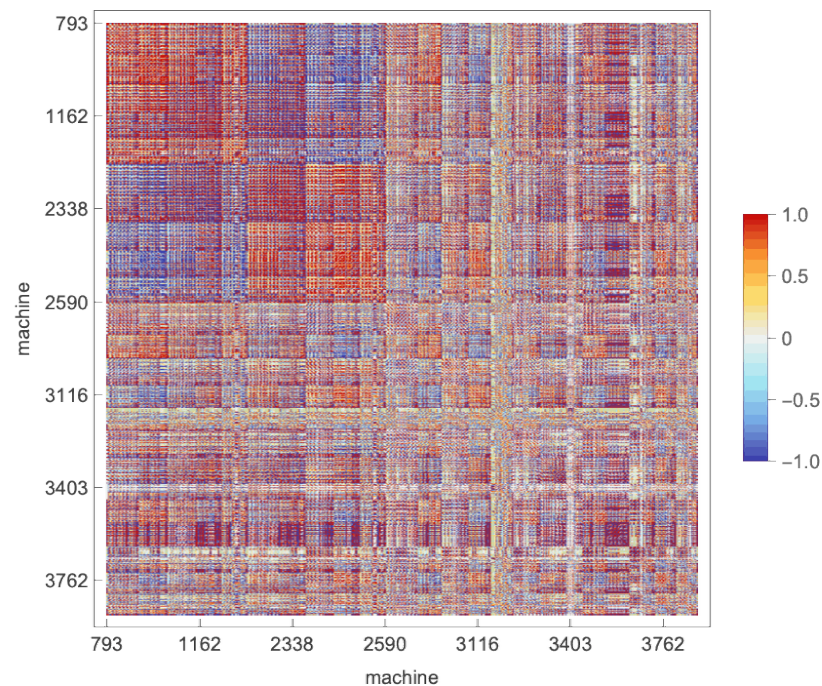


但在这种情况下，不需要太多步数就能清晰看到最终赢家（稍后我们会看到需要更长时间的例子）。

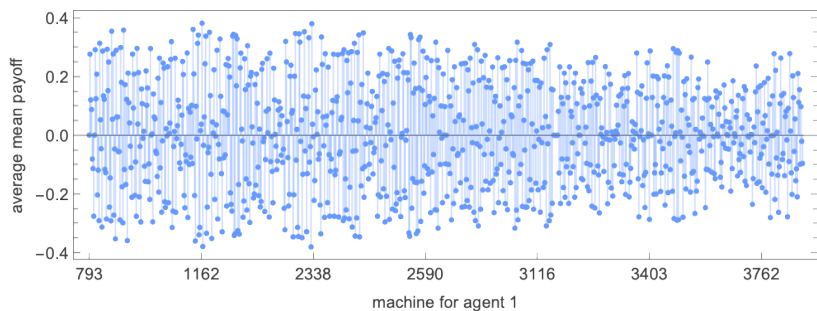
（还有其他细微之处。其中之一是，我们通过让每台机器与所有其他不同的机器对弈来计算平均收益。原则上，我们也可以包含其他等效机器——这会对平均值的权重有微弱影响。但由于我们真正关心的是策略而非机器本身，我们使用的方案似乎更合适。）

三状态机器

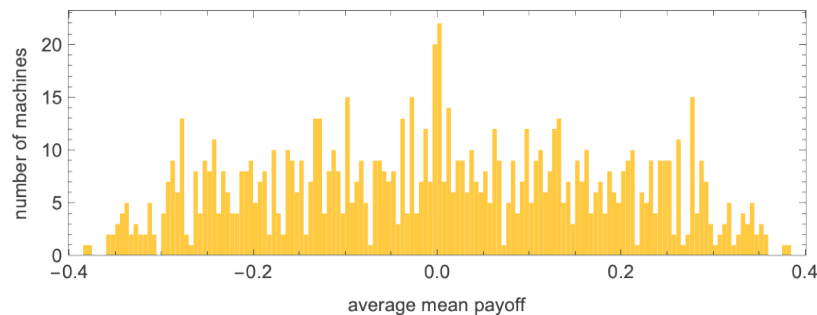
对于 $s=3$ 个状态的 956 台不同的机器，对应的“竞争数组”（经过 1000 步后）为：



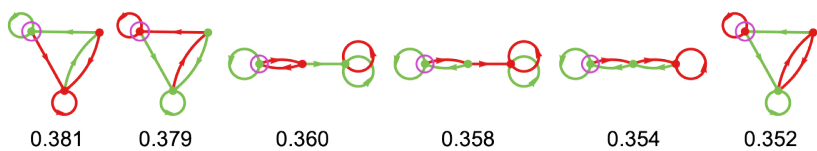
每台机器的平均收益（即“竞争数组”中每行的平均值）为



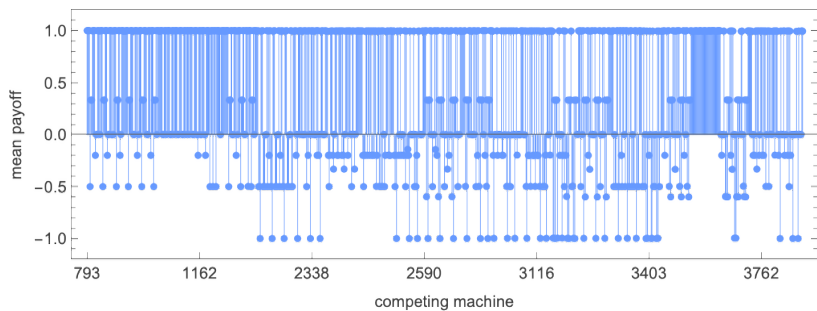
而这些平均收益的分布为：



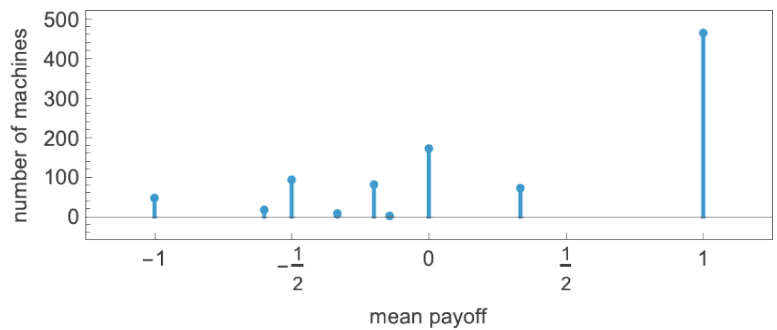
以下是几个最擅长进行“匹配或不匹配”游戏的机器：



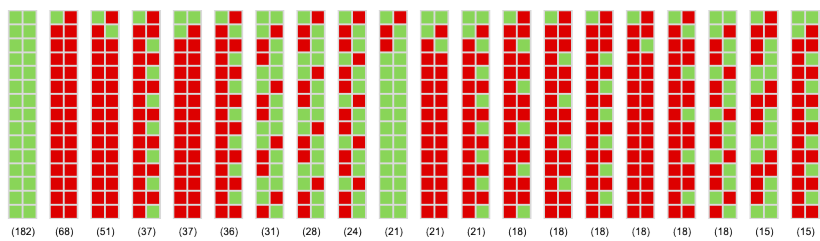
将顶级机器（ $s = 3$ 机器 1164）与所有（不同的）三态机器对战时，我们得到以下平均收益：



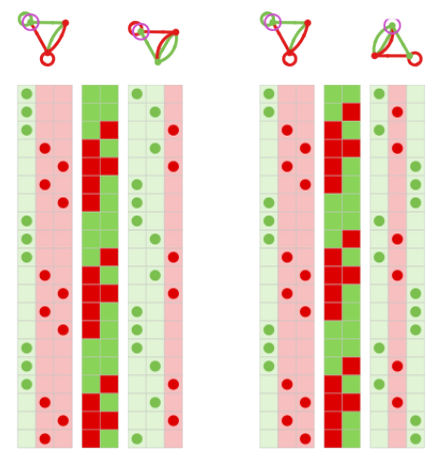
可能的极限平均收益分布如下：



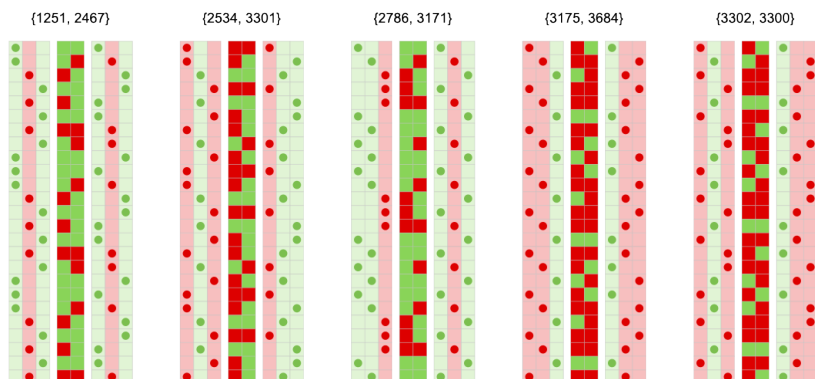
最常见的行为模式如下：



两台三状态机器竞争的最大可能周期为 9。机器 1164 从未完全达到这一数值；其最大周期为 7，出现在与机器 2546 和 2755（两者均给出极限平均收益-1）竞争时：



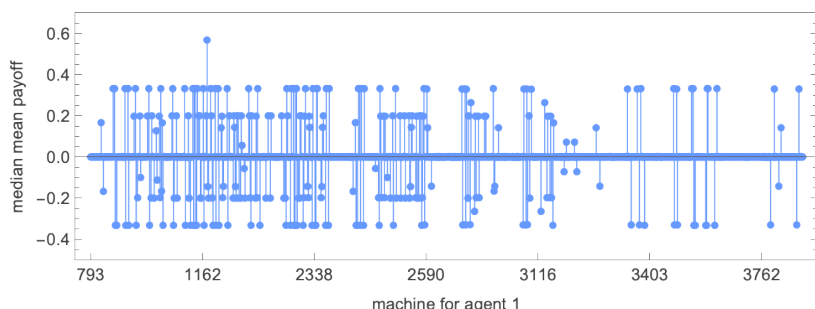
如果考察所有可能的三状态机器配对，结果有 792 对呈现出周期为 9 的行为，例如：



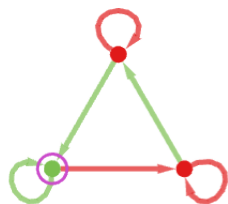
(这些均无瞬态；三状态机器的最大瞬态被证实为 8。)

一个旁注：我们所说的“平均”是什么意思？

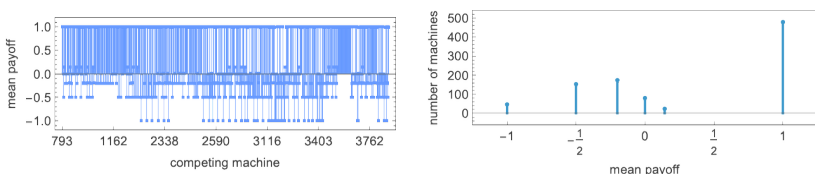
我们已经讨论过，当一台机器与所有其他（不同的）机器竞争时，其表现“平均”如何。但“平均”到底指什么？到目前为止，我们所说的“平均”是指与每台其他机器竞争所得收益的均值（而这里的收益本身是连续步骤中的均值）。但如果我们用中位数代替均值呢？以下是每台机器与其他所有机器运行 1000 步后的中位数收益：



这里脱颖而出的获胜机器是 1172 号：



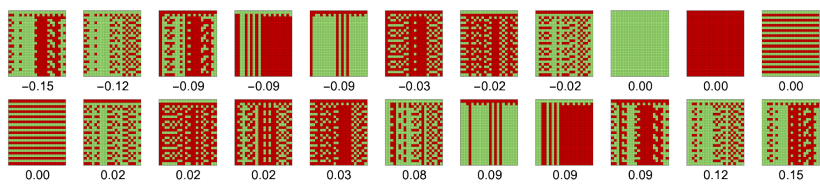
在这种情况下，均值收益及其分布是：



中位数“异常高”，因为该机器恰好有 $1/2$ 的平均收益为+1。（相应的均值因平均收益分布中的“左尾”而被拉低。）

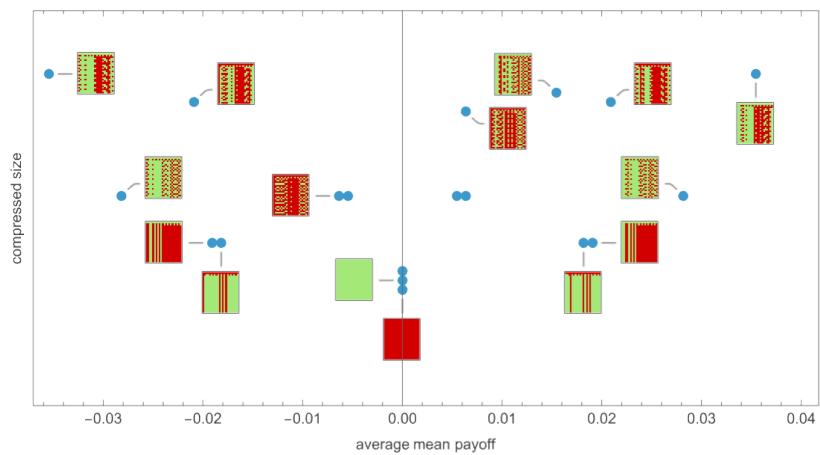
胜利的复杂性

让我们（基本如上所述）观察每个不同的 2 状态有限状态机在与所有其他 2 状态机竞争时的实际行为，按平均收益均值从小到大排序：

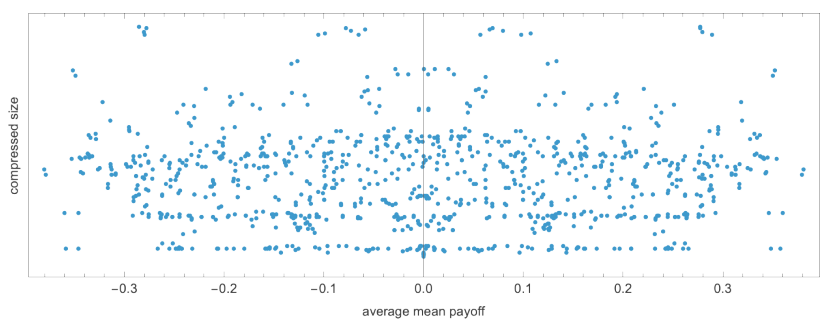


平均收益均值为 0 的情况在行为上看起来很简单。但对于其他平均收益均值，某台机器与所有其他机器竞争时的行为似乎更为复杂。

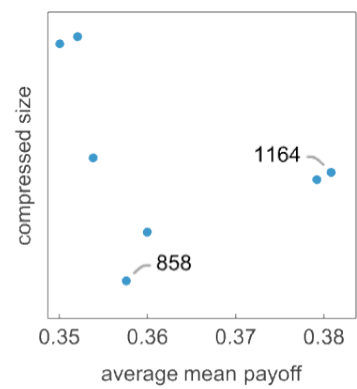
通过观察上述行为数组的压缩大小（由 [Compress](#) 获得），我们可以感受到这种复杂性：



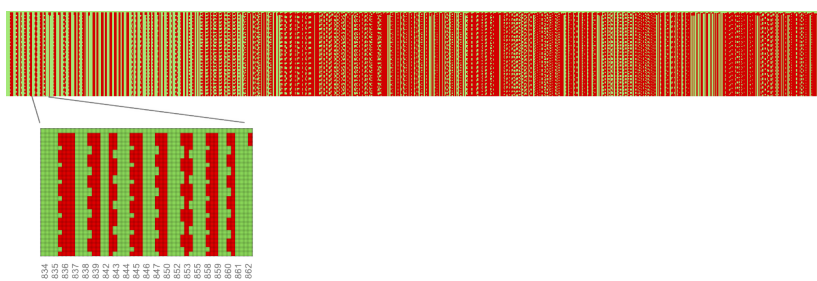
以下是 956 种不同的三状态机器的对应结果——显示平均收益与行为复杂度的估计之间没有强相关性：



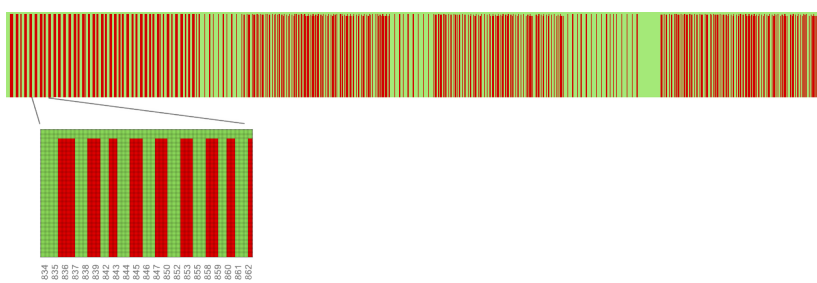
事实上，在平均收益最高的机器中，行为复杂度仍然存在相当大的多样性



而所指机器的“行为轨迹”如下：



以及



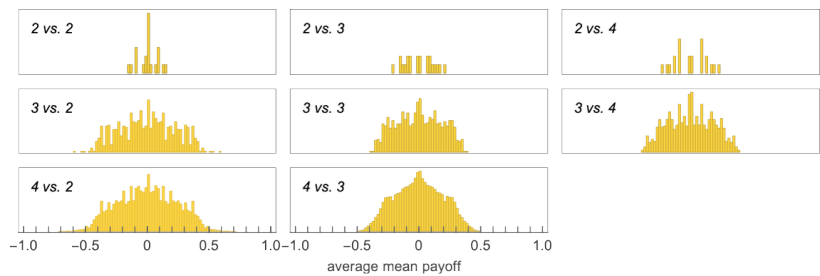
换句话说，至少在这个案例中，我们无法断言获胜的机器要么行为特别复杂，要么特别简单。似乎决定机器能否获胜的，是具体的结构细节，而非整体特征。

不同规模机器之间的竞争

状态数更多的有限状态机是否必然比状态数更少的系统表现更优（即获得更高的收益）？在与所有其他 2 状态机竞争时，任何 2 状态机所能达到的最佳平均收益约为 0.151。但若考虑 3 状态机与 2 状态机竞争（进行 1000 轮），其最佳平均收益则提升至 0.593。

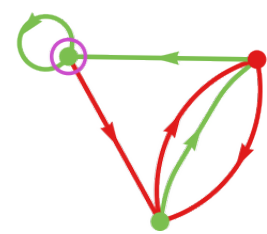
winner states	opponent states	winner machine	average payoff
1	1	0	0.0
1	2	0	0.0
1	3	1	0.0
1	4	1	0.0
2	1	22	0.999
2	2	26	0.151
2	3	22	0.208
2	4	22	0.235
3	1	804	0.999
3	2	1234	0.593
3	3	1164	0.381
3	4	1164	0.389
4	1	74 520	0.999
4	2	109 828	0.712
4	3	104 438	0.496

观察可能平均收益的分布，我们发现三状态机的平均收益分布比二状态机更广——这一事实至少部分是由于三状态机可能的数量远多于二状态机所致：

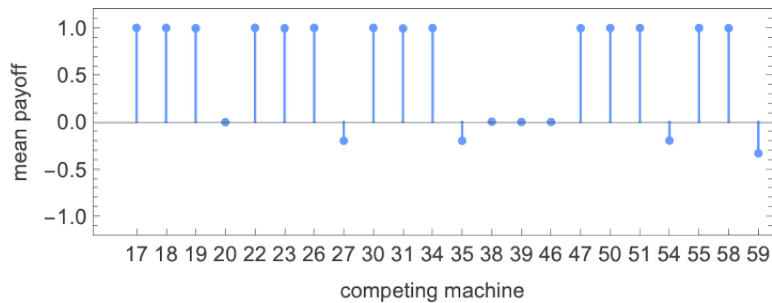


但值得注意的一点是，分布最广的是三状态机器与二状态机器之间的竞争：实际上，三状态机器凭借其更丰富的策略集，似乎能更有效地"智取"二状态机器。

在所有与二状态机器对决的整体表现中，最出色的三状态机器是 1234 号：



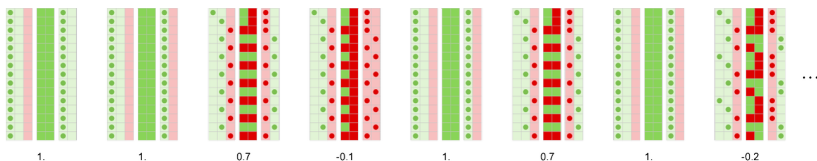
它并非总能稳操胜券（平均收益为+1），但在大多数情况下都能获胜：



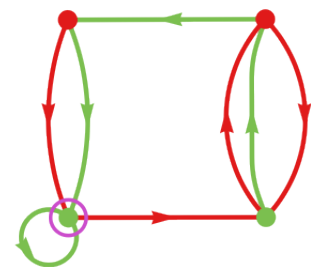
它是如何做到的呢？基本上，面对许多不同的二状态机器，这个特定的三状态机器都能模仿它们的行为：



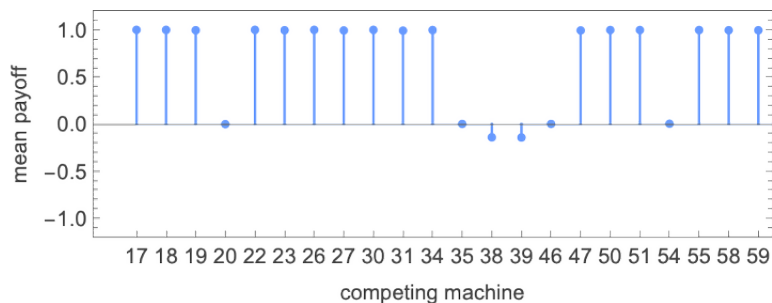
在某种意义上，3 状态机的某些方面与许多 2 状态机“共鸣”：



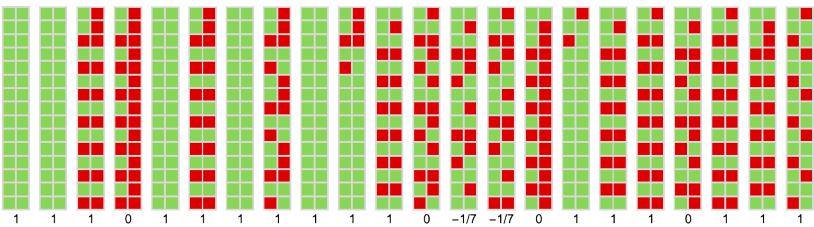
那么 4 状态机呢？在与 2 状态机的整体对战中表现最好的 4 状态机是机器 109828：



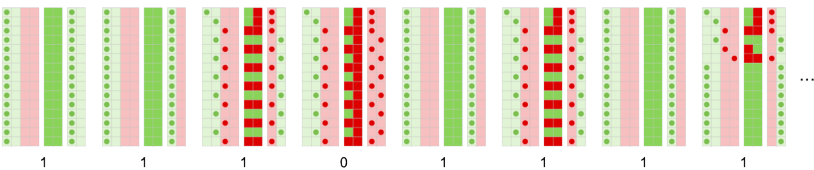
在 22 个 2 状态机中，它仅在 6 种情况下获得低于+1 的收益：



以下是全部 22 种情况的行为：



我们再次可以将 4 状态机器视为成功地“覆盖”了大多数 2 状态行为：

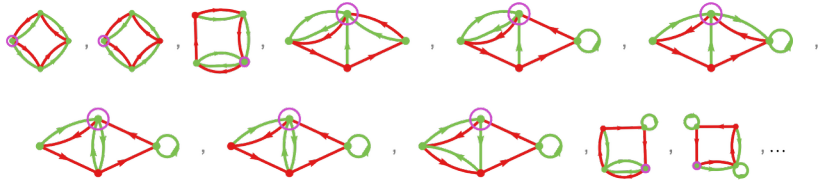


有限状态机的自适应演化

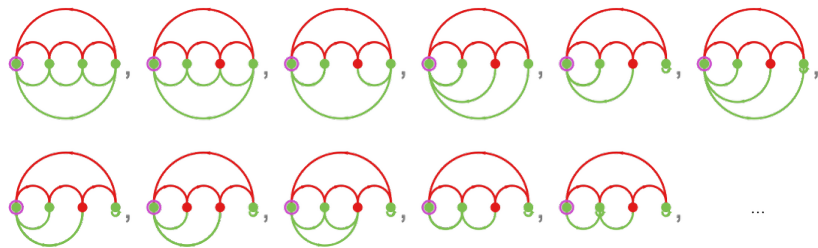
在许多存在竞争的实际场景中，竞争的参与者有办法进行演化。那么，我们能否使用有限状态机来构建一个最小模型呢？

在我们迄今为止的工作中，我们一直在考察所有可能有限状态机构成的空间。但通过自适应演化得到的机器序列又如何呢？是否存在一种方法，可以通过自适应演化让机器在竞争中表现越来越好？

执行此操作的第一步是观察如何对有限状态机进行连续突变。一个简单的方法是，任何给定的突变可以影响机器图中的随机顶点或随机边。对于顶点，突变只是翻转其颜色。对于边，要么翻转颜色，要么将边“重新路由”到不同的顶点（约束条件是这样做不会断开图的连接）。随机应用一系列此类突变，例如得到：

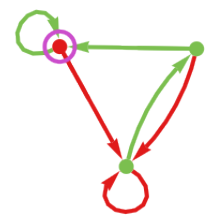


或者，使用不同的图渲染：

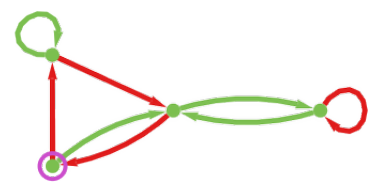


(注意，我们是在找到机器时对其任意形式进行突变；我们不关心机器之间的等价性或机器的规范化。)

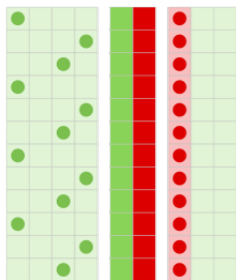
假设我们有一个对手机器——例如 3 状态机器 1165——它通常会导致失败，即限制收益为-1（例如，在与其它 3 状态机器竞争时大约有一半时间如此）：



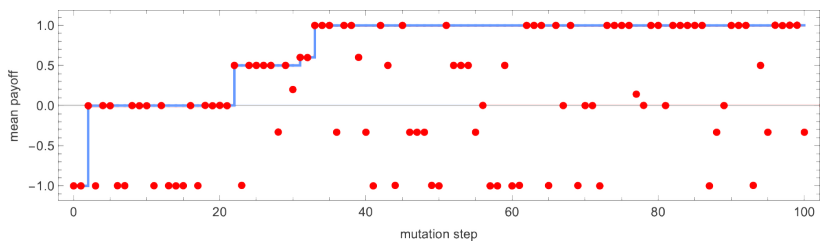
现在我们可以询问：是否能够自适应地进化出一个能够战胜这个对手机器？为了让自适应进化过程有一定的“操作空间”，我们将使用一个 4 状态机器。我们可以从一个随机的此类机器开始，例如



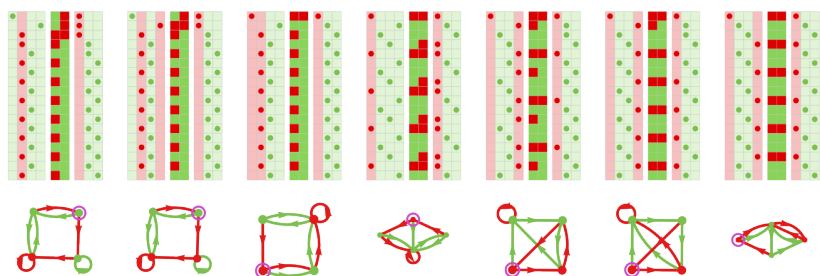
该机器在与机器 1165 对战时“落败”（收益始终为-1）：



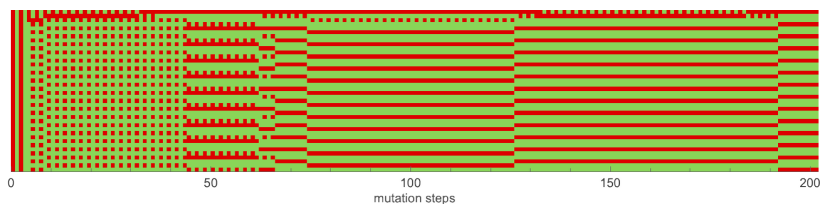
为了进行自适应进化，我们现在对该机器进行连续随机突变，“接受”那些不降低平均收益的突变，否则拒绝。结果是一条典型的“适应度曲线”，其中大多数突变（用红点表示）并未带来收益改善——但也存在一些导致“突破”的突变，使收益增加（有时增幅很小），最终收益达到最大值+1：



各种“突破”逐步收敛于一个收益为+1 的“完美解”：



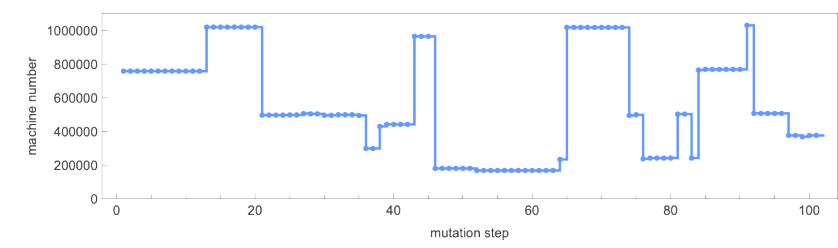
将自适应进化过程中的连续结果串联起来，我们可以看到最终收敛到完美解的过程，即两个智能体的行为始终匹配：



在不同的随机突变下，“适应度曲线”在细节上会有所不同，但总体形式保持一致。对于不同的特定对手也是如此。

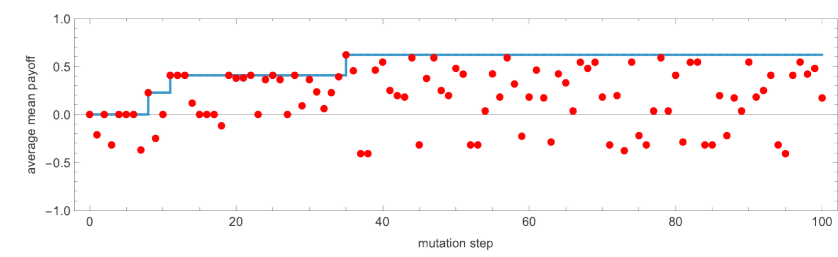
顺便提一下，利用我们为有限状态机编号的方法，我们可以绘制出自适应进化过程如何在规则空间

中“移动机器”的图表：

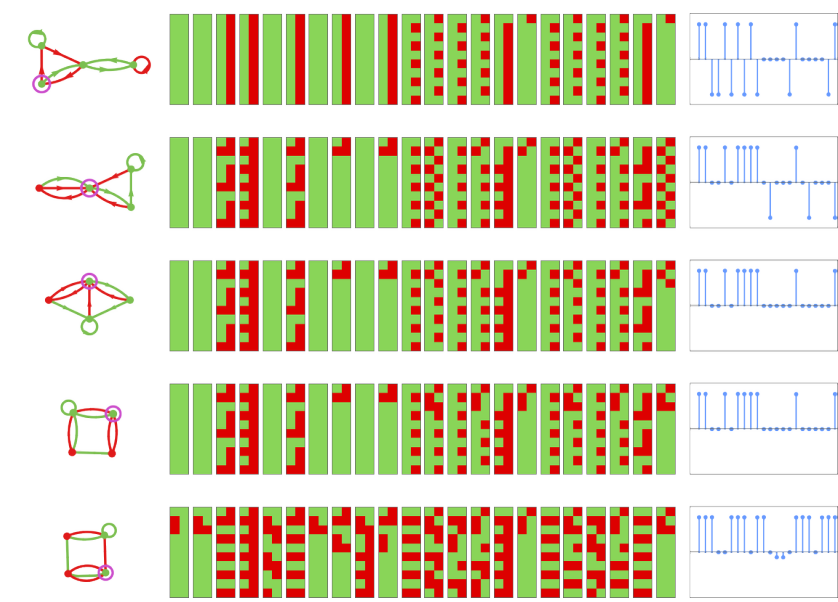


但如果我们像之前那样，询问针对给定规模的所有可能有限状态机对手的平均收益，会发生什么？例如，4 状态机器对抗所有可能的 2 状态机器时能表现得多好？

从与之前相同的随机 4 状态机开始，典型的适应度曲线为：

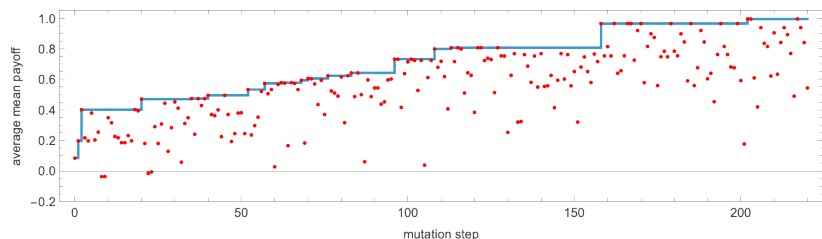


此处适应度增加，但从未达到+1。连续“突破性”机器与所有 2 状态机对弈的行为如下：

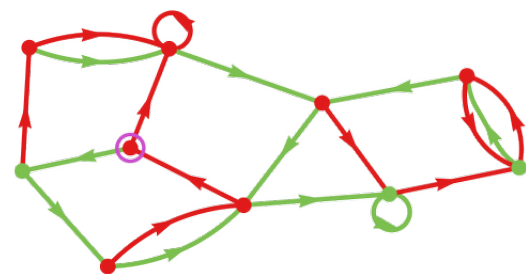


我们可以看到，即使我们得到了最佳机器，它仍然会输给某些 2 状态机，最终平均收益约为 0.62。

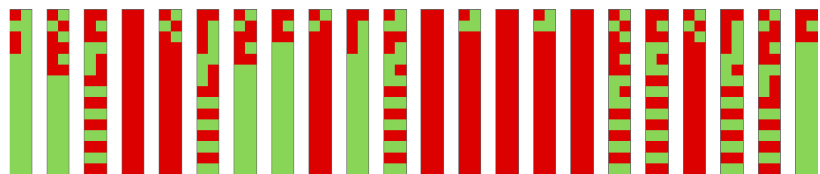
那么，如果我们考虑具有更多状态的机器会怎样？例如，对于 10 个状态，可能通过自适应进化获得一个对每个 2 状态机都能达到极限收益+1 的机器：



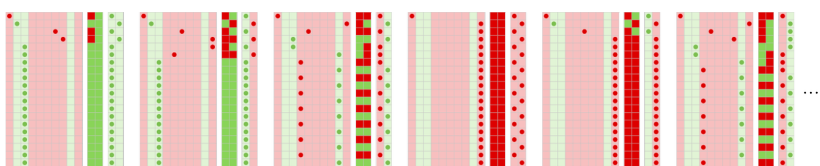
在此情况下获得的最终机器



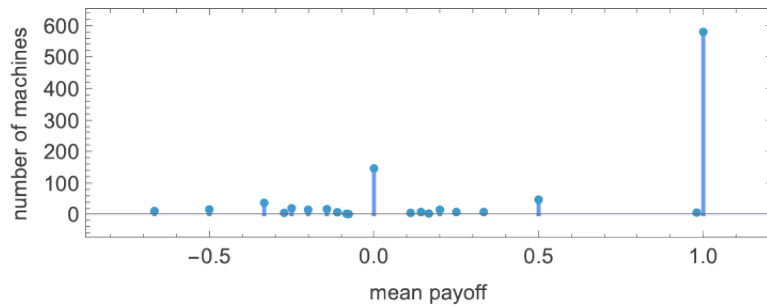
可以被视为一种（2 状态）“万能赢家”——它最终战胜了所有 2 状态机器：



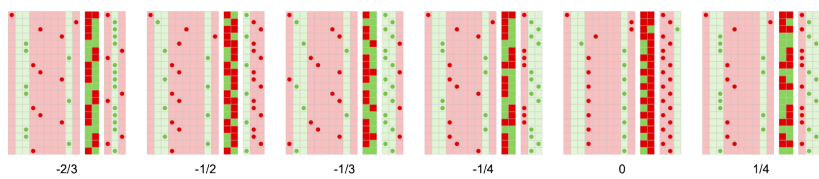
它是如何做到的呢？从某种意义上说，这台机器足够庞大，因此对不同对手可以拥有不同的“专用部件”。如果我们观察机器的运行方式，确实会发现，面对不同的对手，机器会进入其完整状态空间中的不同子集：



即使我们考虑所有 956 种三状态机器作为对手，我们的机器依然表现出色。虽然并非在所有情况下都能获胜，但它仍实现了平均+0.603 的均值收益：



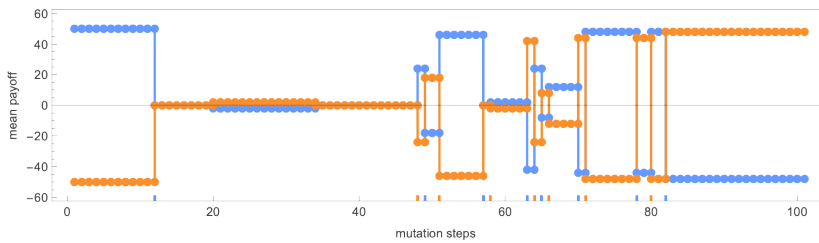
一些机器未能获胜的例子——实际上是因为它未能包含作为子机器来处理特定对手的部件——包括：



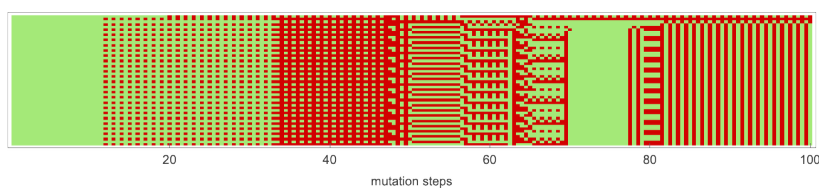
到目前为止，我们考虑的是一台机器与单个固定对手或一组固定对手竞争时的自适应进化。但如果机器与其对手都在进行自适应进化呢？

例如，假设在交替的适应性进化步骤中，我们对一个机器及其对手进行突变。如果该机器的（平均）收益没有下降，我们就保留该突变；否则，我们拒绝它。

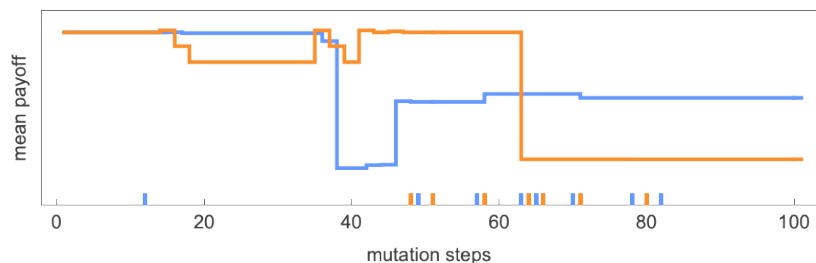
在此设置下，下面是两个（最初相同的）4 状态机器的平均收益演化：



存在一个机器获胜的时期，以及其对手获胜的时期——这可以从机器的实际连续行为中看出：



通过适应性进化发现的实际机器在规则空间中移动——很快丢失了它们最初形态的记忆：



如果机器中的状态数量发生变化或不相同，情况不会有太大改变——尽管对于状态较多的机器，胜者通常交替较少，大概是因为状态更多时，每个个体突变对行为的影响往往较小。

囚徒困境呢？

到目前为止，我们所做的一切都基于一个特别简单的游戏——匹配或不匹配（“匹配硬币”）。那么其他游戏的情况如何呢？特别是著名的“囚徒困境”游戏？以下是该游戏的收益

agent 1		agent 2	
payoff	action	action	payoff
-1	合作	合作	-1
-3	合作	背叛	0
0	背叛	合作	-3
-2	背叛	背叛	-2

在该游戏通常的叙述中，人们将 ■ 解释为“背叛”，将 ■ 解释为“合作”。

正如上文所述，我们可以设想基于有限状态机来定义囚徒困境游戏的策略。以下是几个两状态机器之间迭代游戏的例子——其收益由囚徒困境游戏决定：

agent 1		agent 2	
payoff	action	action	payoff
-3	合作	合作	0
-2	合作	背叛	-2
-1	合作	合作	-1
-3	合作	背叛	0
-2	背叛	合作	-2
-1	背叛	合作	-1
-3	背叛	合作	0
-2	背叛	背叛	-2
-1	背叛	背叛	-1
-3	背叛	背叛	0

agent 1		agent 2	
payoff	action	action	payoff
-3	合作	合作	0
0	合作	背叛	-3
-3	合作	合作	0
-3	合作	背叛	-3
0	背叛	合作	0
-3	背叛	合作	-3
0	背叛	背叛	0
-3	背叛	背叛	-3
0	背叛	合作	0
-3	背叛	合作	-3

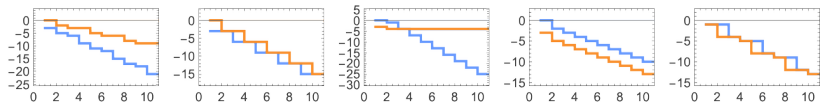
agent 1		agent 2	
payoff	action	action	payoff
0	合作	合作	-3
-1	合作	背叛	-1
-3	合作	合作	0
-3	合作	背叛	0
-3	背叛	合作	0
-3	背叛	合作	0
-3	背叛	背叛	0
-3	背叛	背叛	0
-3	背叛	合作	0
-3	背叛	合作	0

agent 1		agent 2	
payoff	action	action	payoff
0	合作	合作	-3
-2	合作	背叛	-2
-1	合作	合作	-1
-1	合作	背叛	-1
-1	背叛	合作	-1
-1	背叛	合作	-1
-1	背叛	背叛	-1
-1	背叛	背叛	-1
-1	背叛	合作	-1
-1	背叛	合作	-1

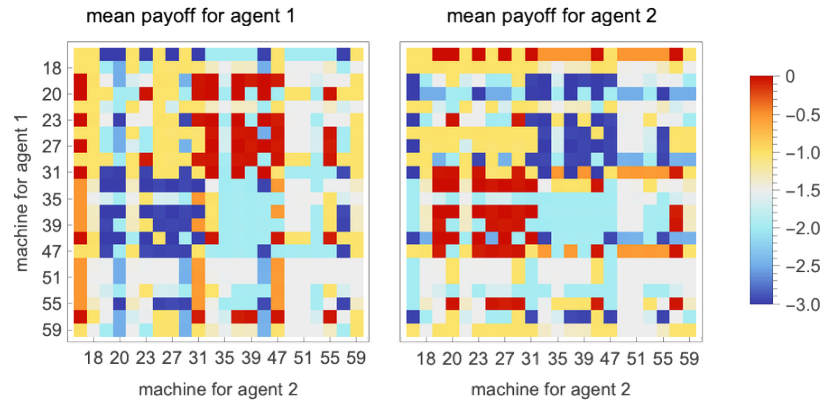
agent 1		agent 2	
payoff	action	action	payoff
-1	合作	合作	-1
0	合作	背叛	-3
-3	合作	合作	0
-1	合作	背叛	-1
0	背叛	合作	-3
-3	背叛	合作	0
-1	背叛	背叛	-1
0	背叛	合作	-3
-3	背叛	合作	0
-1	背叛	背叛	-1

在匹配与否的案例中，只需观察代理人在特定步骤中的行为是否匹配，就能直观地判断特定收益是 ± 1 还是 0。而在此处，这种视觉判断并不那么明显。

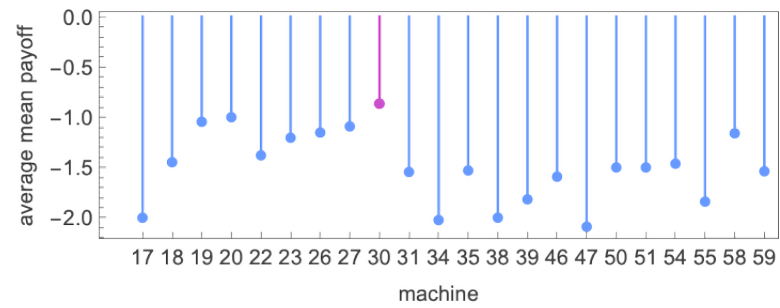
但利用囚徒困境游戏的收益，我们可以计算这些示例的累积收益（与零和博弈的匹配与否不同，这里两个代理人的收益在每一步中之和并不为零）：



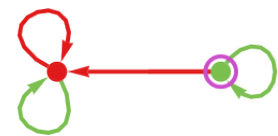
正如我们之前所做的那样，现在我们可以考虑代理人之间的竞争，其策略基于所有可能的两状态有限状态机（对于匹配与否，游戏的零和性质使得收益数组对称；而在这里，对称性仅来源于互换代理人 1 和代理人 2 的角色时收益保持不变这一事实）：



有了这样的设置，我们就可以问哪台机器是“总体赢家”——比如，在与所有其他（不同的）2 状态机器的对战中，它的平均收益最大：

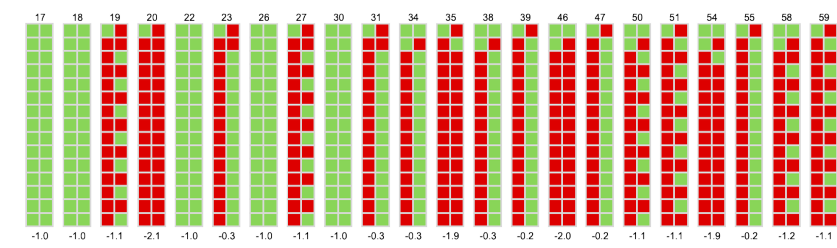


答案原来是机器 30：

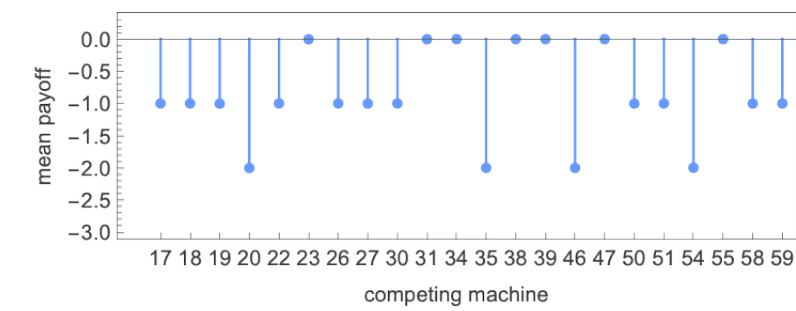


在囚徒困境的文献中，这通常被称为“冷酷触发”，因为它产生了一种策略：开始时选择 ■，然后重复这一行为，直到对手首次给出 ■ ——此后永远选择 ■。

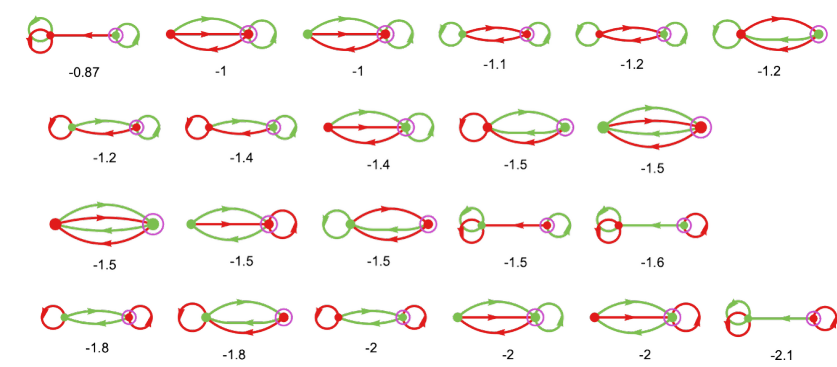
让这台机器与所有其他 2 状态机器对战后，我们得到以下行为：



对应于以下平均收益：



观察所有两状态机器的平均收益，这些机器的排名如下：

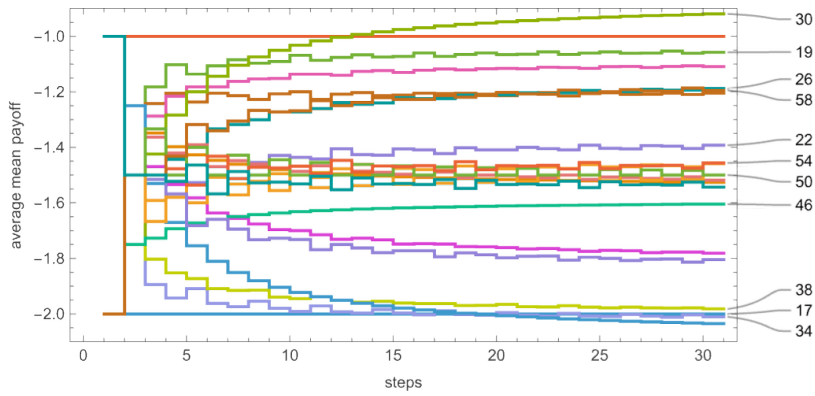


值得注意的是，机器 22（对应于著名的“以牙还牙”策略）

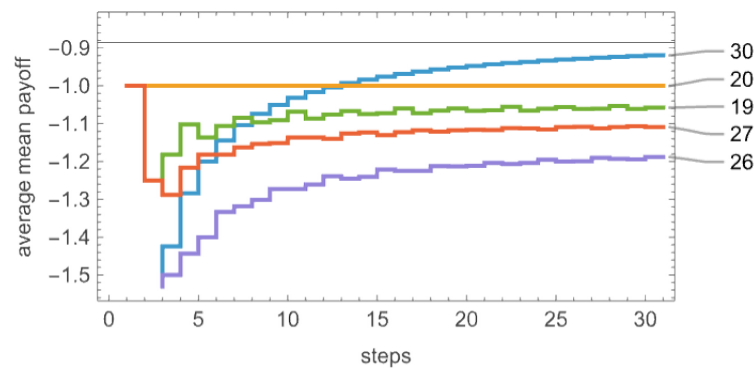


在这个排名中相当靠后，尽管它常被认定为人类建议策略集合中最成功的。

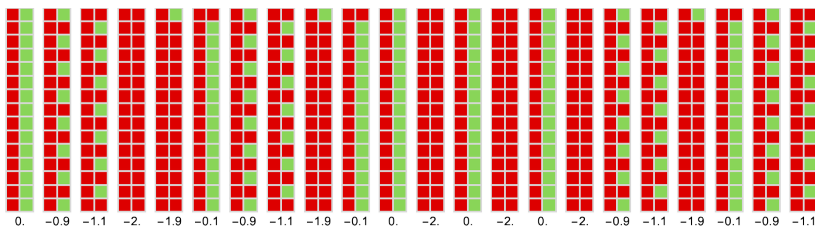
我们刚才给出的排名是基于多次囚徒困境博弈后获得的平均收益。但如果只进行少数几次迭代，排名可能会有所不同：



放大开头，我们可以看到机器 30 在 13 步后才开始获胜：

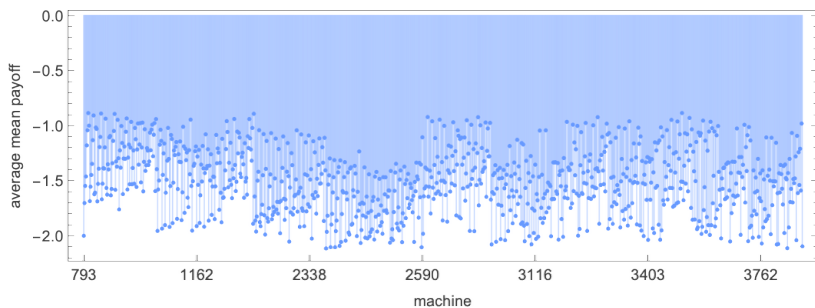


机器 20 给出的恒定平均收益为-1，来自

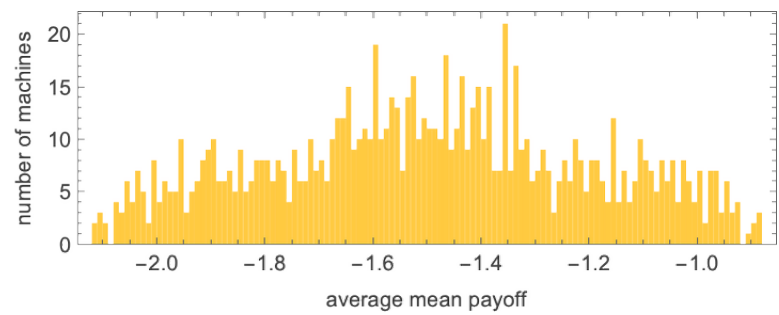


而机器 30 产生的平均收益由 $-\frac{37}{22t} - \frac{19}{22}$ 给出，极限值为 $-\frac{19}{22} \approx -0.86$ 。

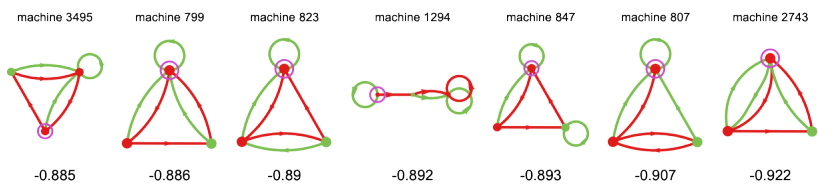
那么 3 状态机器呢？这给出了这些机器各自的平均囚徒困境收益：



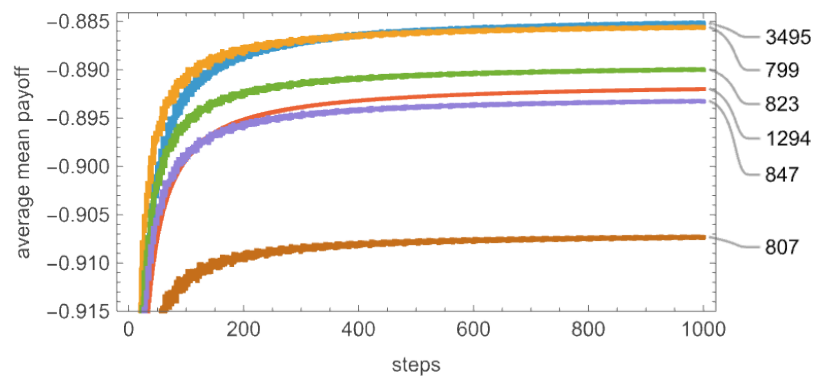
这些平均收益的分布为：



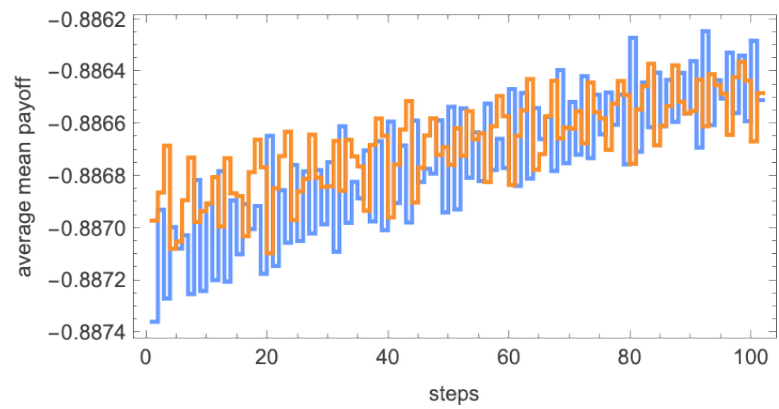
最终平均收益最高的机器是：



但这种排序仅在超过 500 步后才显现出来



平均收益的交叉情况异常复杂：

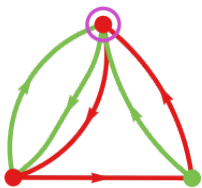


(机器之间竞赛行为的最终周期性行为中，许多不同周期的组合反映在看似相当随机的平均收益均

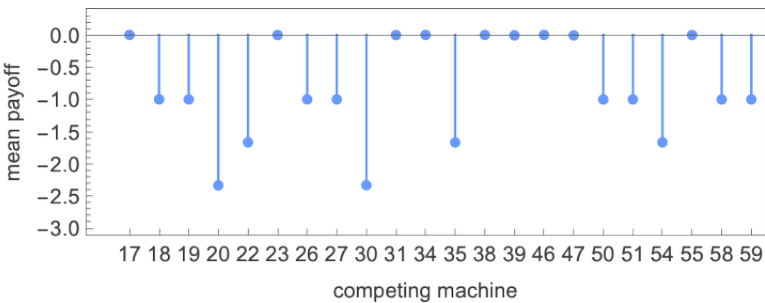
值变化上。)

那么，在囚徒困境博弈中，3 状态机器与 2 状态机器相比表现如何？将 2 状态机器相互对抗时，机器 30 获得的最高平均收益均值约为-0.866。与此同时，当 3 状态机器相互对抗时，获得的最高平均收益均值略小，为-0.885。那么 2 状态机器与 3 状态机器对抗呢？它们表现不佳。机器 30 表现最好——但现在它给出的平均收益均值不是-0.866，而是约为-0.97。

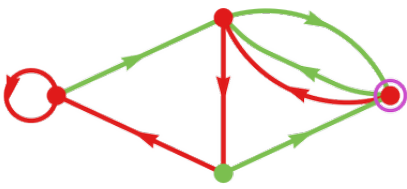
然而，当 3 状态机器与 2 状态机器对抗时，最佳平均收益均值更大——约为-0.80，由机器 2743 实现。



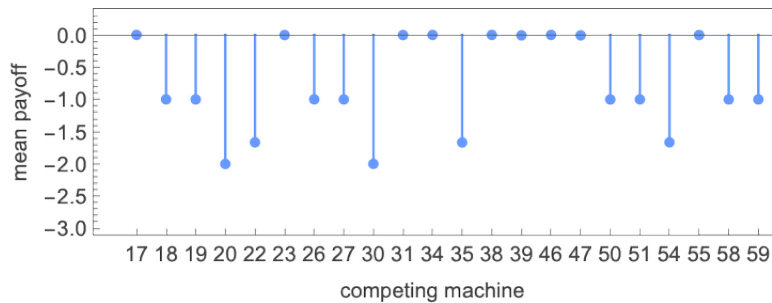
将其与每种可能的 2 状态机器对抗时获得的平均收益如下：



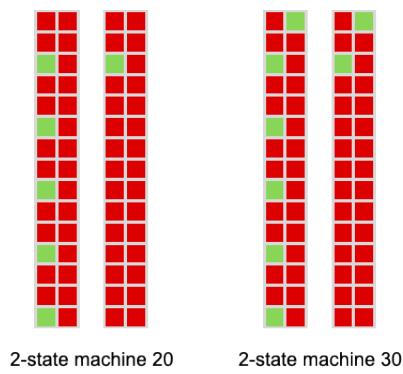
那四状态机呢？将所有这些机器与二状态机对局，最终胜者是机器 336766，平均收益为-0.77：



此情况下每个二状态机的平均收益



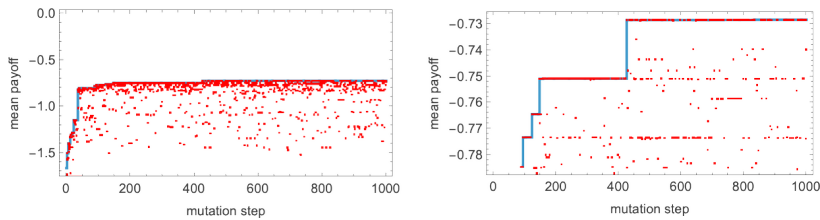
与获胜的三状态机非常相似，唯一不同的行为发生在对手是二状态机 20 和 30 时：



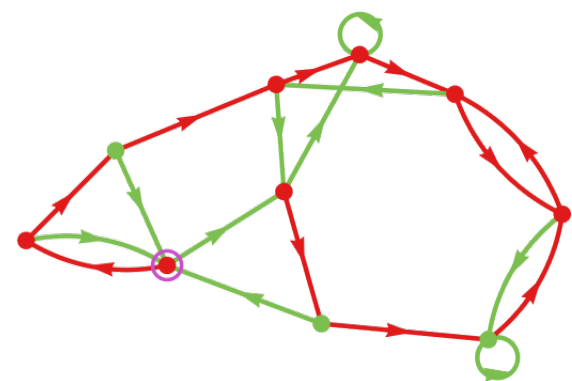
总结这些结果，我们在囚徒困境中找到的获胜的小状态数机器是：

winner states	opponent states	winner machine	average payoff
1	1	0	-1.000
1	2	0	-1.000
1	3	0	-1.000
1	4	0	-1.000
2	1	20	-1.000
2	2	30	-0.865
2	3	30	-0.972
2	4	20	-1.000
3	1	800	-1.000
3	2	2743	-0.804
3	3	3495	-0.885
3	4	1294	-0.911
4	1	74 512	-1.000
4	2	336 766	-0.773
4	3	456 142	-0.847

但是，那些拥有更多状态的机器呢——我们或许可以通过自适应进化找到它们？这里有一个 10 状态的自适应进化例子，与所有 2 状态机器进行竞争：

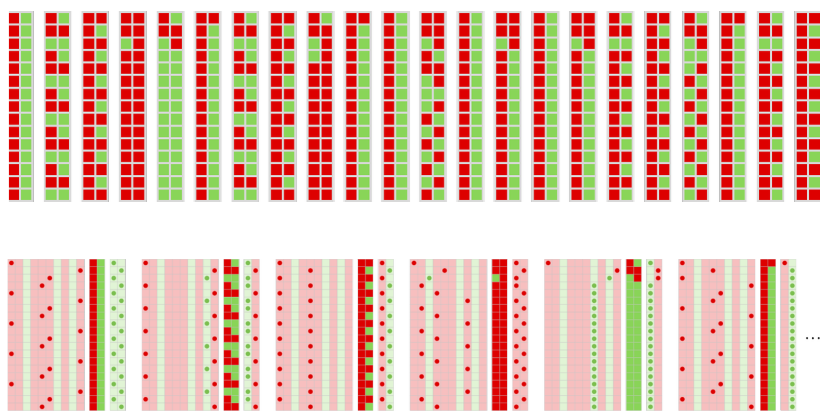


经过 1000 步这样的自适应进化，我们得到了这个 10 状态机器



with average mean payoff -0.73 .

The behavior of this machine competing with all 2-state machines is:



The Space of All Possible Games

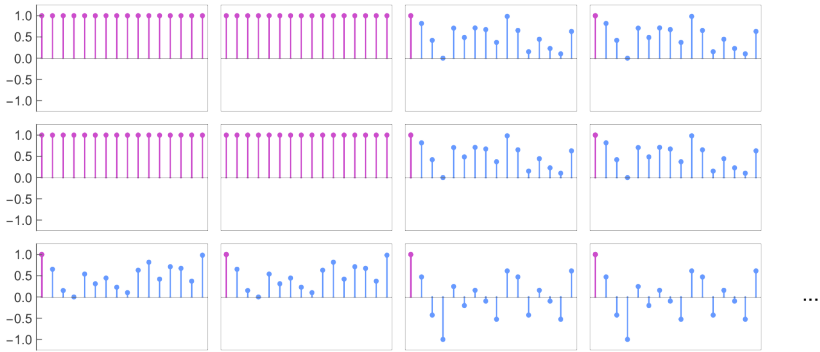
We’ve now looked at two specific examples of games—match-or-not and prisoner’s dilemma—and we’ve seen very similar phenomena in both cases. But what about other games?

如果我们允许收益为-1 和+1（如匹配或不匹配），那么总共有 256 种可能的博弈：



其中，16 种是零和博弈（如匹配或不匹配）——即两个参与者的收益总和始终为零，另外 16 种是对称博弈（如囚徒困境）——即两个参与者的收益始终相同。

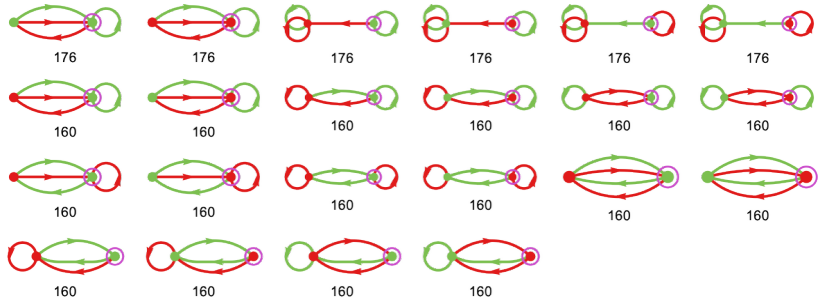
对于这 256 种可能的博弈中的每一种，我们可以计算每个可能的 2 状态有限状态机在与所有 2 状态机竞争时的平均收益：



这 256 种博弈的获胜平均收益始终为-1、0 或+1：

maximum payoff	-1	0	+1
number of games	16	128	112

在大多数情况下，许多机器都能获得最大收益；在所有游戏中，每个机器成为赢家的次数如下：



当我们审视更多游戏时——例如那些收益为 -1、0、+1 的游戏——情况又如何呢？此类游戏共有 6561 种。结果非常相似，仅有细微差别：

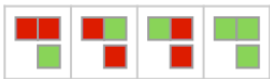
maximum payoff	-1	0	+1
number of games	81	4131	2349

细胞自动机策略

我们迄今为止所做的一切都基于使用有限状态机作为策略的来源。现在，我们将转向另一个策略来源：细胞自动机。

我们采用的方式是，智能体的行为由运行细胞自动机规则决定。基本思路是，每一步，细胞自动机的初始条件由对手迄今为止所采取的行动序列给出。然后，我们的智能体的下一个行为由运行细胞自动机（运行步数与对手当前已采取的动作数量相同）所得到的细胞值决定。

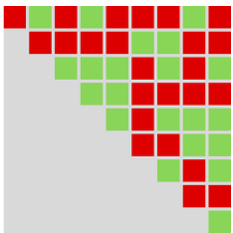
具体来说，假设我们的细胞自动机规则为：



再假设对手迄今为止所采取的行动为：



然后，想法是使用这些作为初始条件运行细胞自动机



并提取最终细胞值以确定下一步行动。

例如，如果我们的两个竞争的细胞自动机有规则



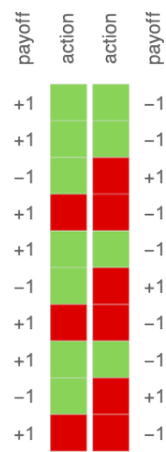
那么将它们对运行的连续步骤给出



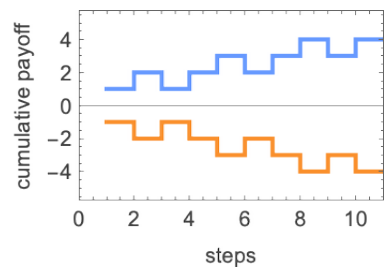
在我们的图片中，关于第二条规则的一切都被反转了。每一步采取的行动现在可以从对手的初始条件中读取，或者从最终生成图案的外对角线上读取。



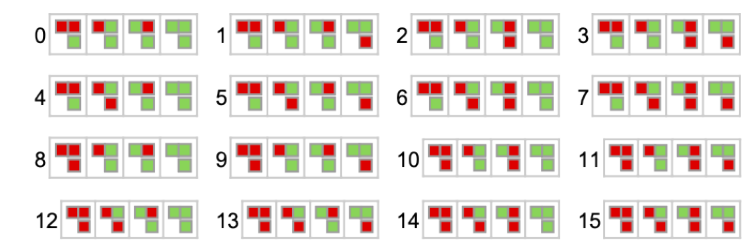
为了分析规则之间的“竞争”，我们可以指定收益，例如来自匹配与否游戏：



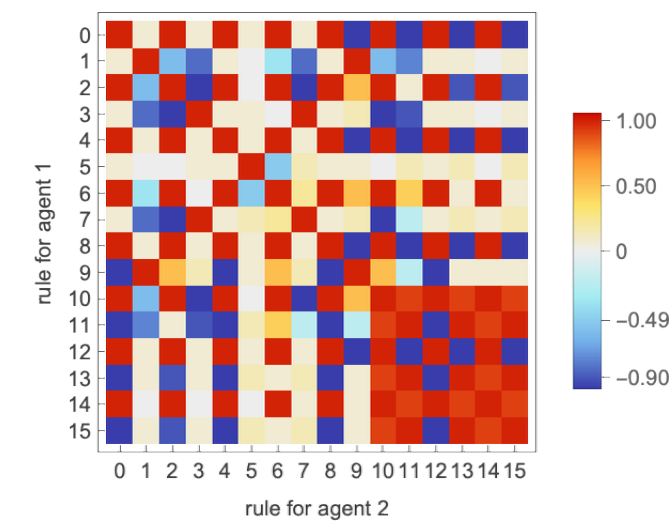
在这种情况下，我们得到以下累积收益：



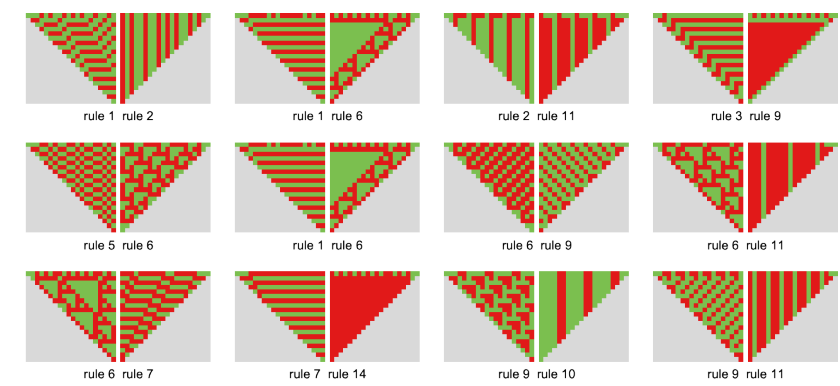
这里使用的这种元胞自动机总共有 16 种可能的规则：



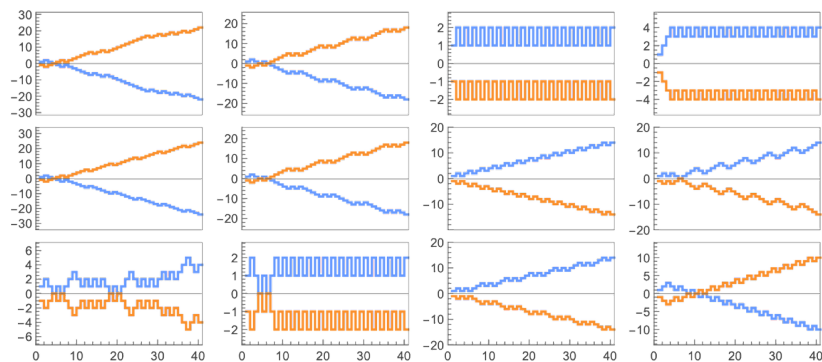
将每一个与其他每一个进行对抗，我们得到以下极限平均收益矩阵：



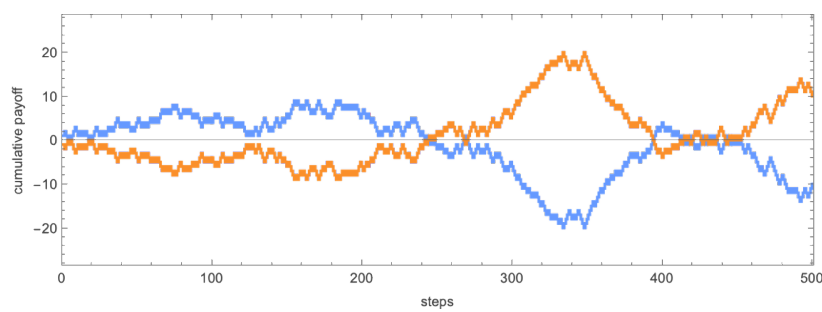
一些值得注意的“竞争”包括：



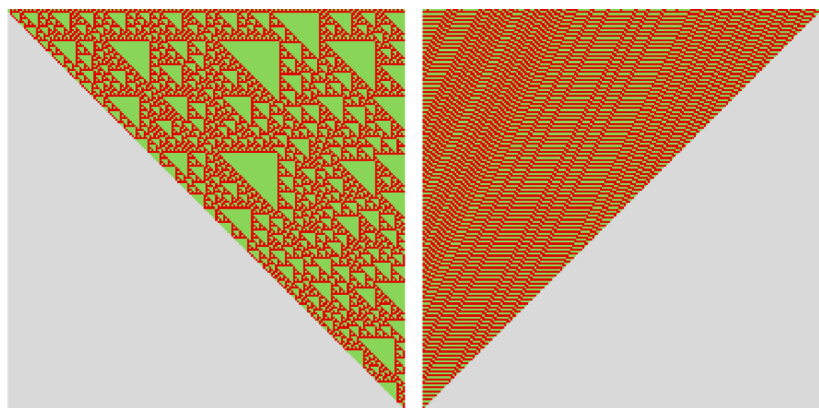
在这些情况下，累积均值（匹配或不匹配）收益为：



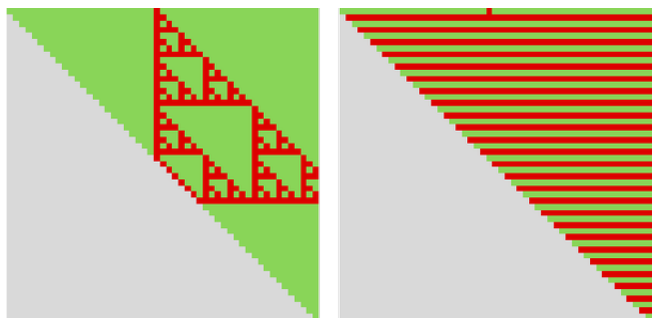
对于大多数规则对，赢家很快就会明朗。但规则 6 与规则 7 的情况则更为复杂——经过 500 步后，仍然完全不清楚哪条规则会获胜：



底层行为是：

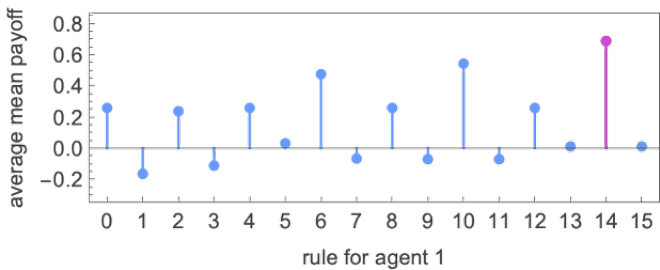


单独来看，这两个规则的行为相当简单（实际上，规则 7 就是异或运算）：

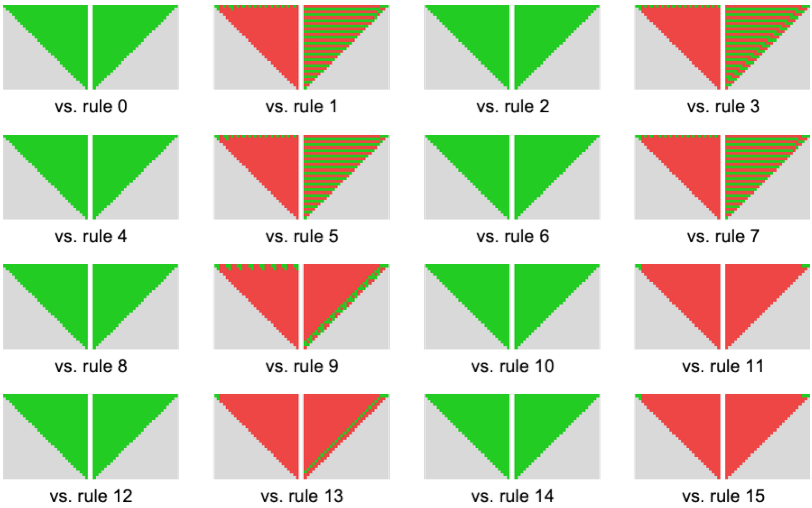


但当它们被置于竞争环境中时，涌现出的有效规则表现出更为复杂——且明显不可预测——的行为，例如，没有任何周期性的迹象。

纵观所有规则，平均收益最大的那个竟然是规则 14：



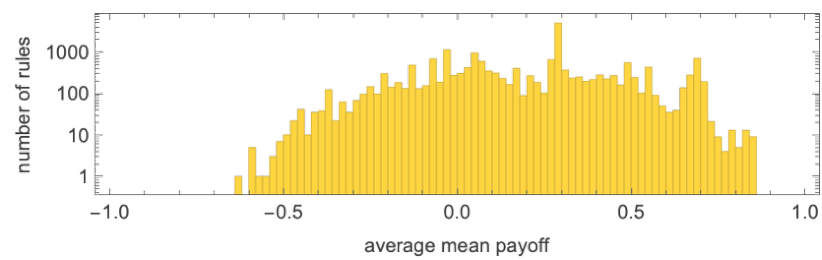
从某种意义上讲，规则 14 找到了一种非常“简单的解决方案”，它要么产生恒定行为，要么产生周期为 2 的行为，并迫使对手也如此——最终给出恰好为 $-\frac{11}{16} \approx -0.69$ 的平均收益：



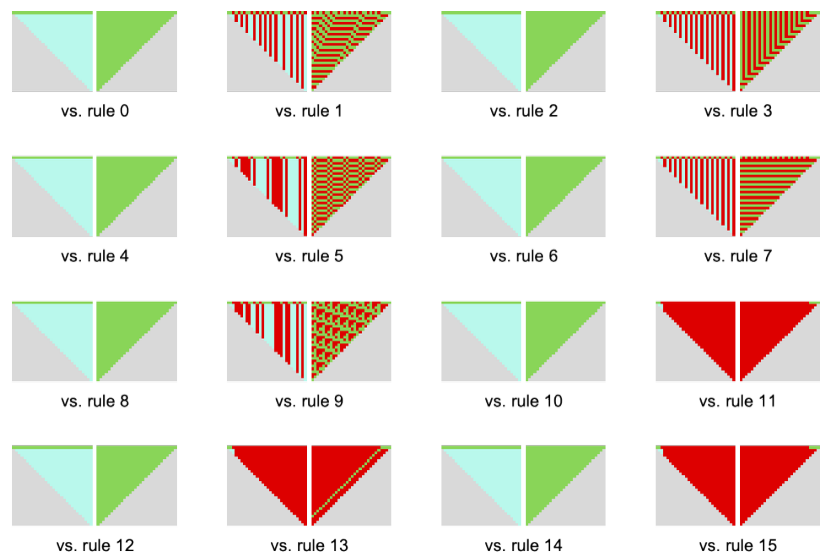
对于更复杂的元胞自动机规则呢？胜者是否仍然是那些行为简单的规则？

让我们看看元胞自动机规则的 3 色模拟。这样的规则共有 $3^{3^2} = 19683$ 条。在每种情况下，我们可以通过查看最终值模 2 来“决定下一步行动”。将所有这些规则与 16 条 2 色规则进行对抗，得分分布

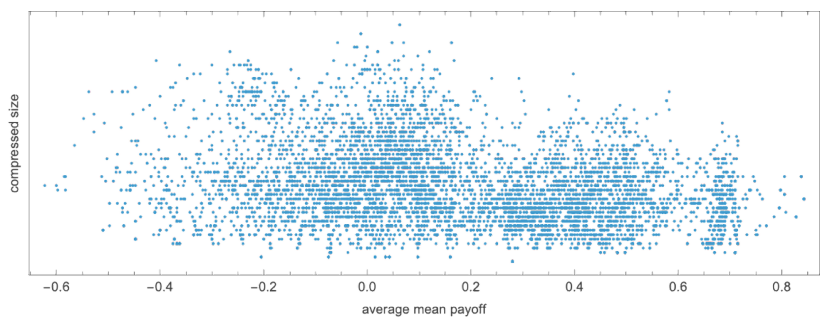
如下：



再一次地，表现最好的规则（如规则 15911）以相当简单的方式运行：

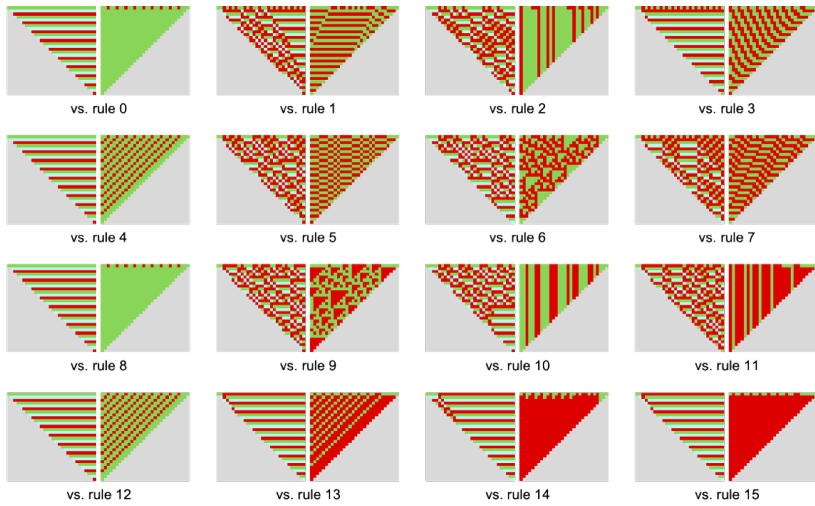


就像我们对有限状态机所做的那样，观察模式的压缩大小与相应竞赛中的平均收益之间的关系，

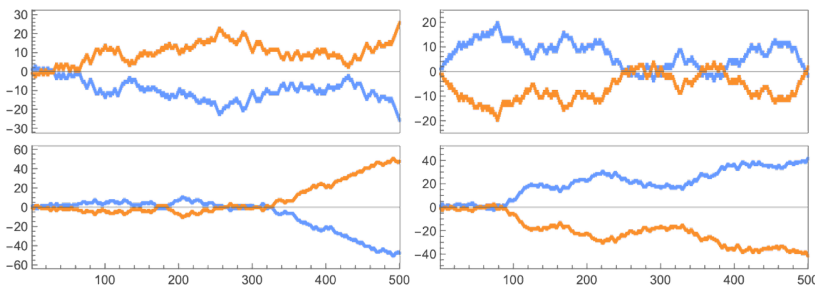
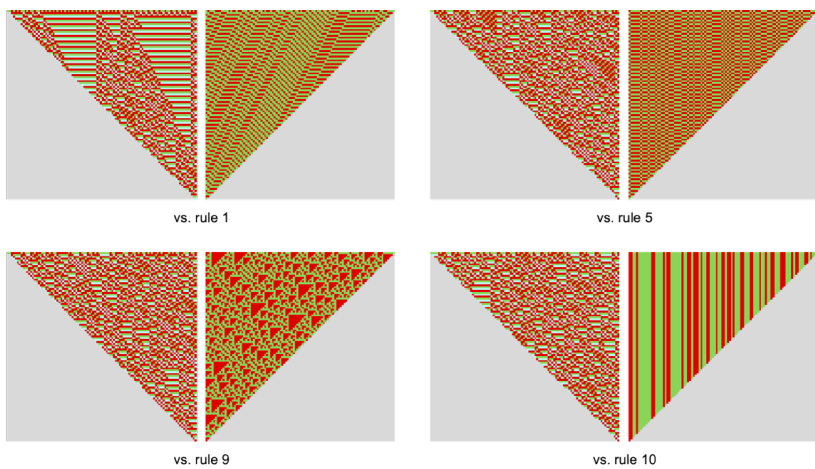


我们可以看到，收益最高的规则往往以更简单的方式运行。

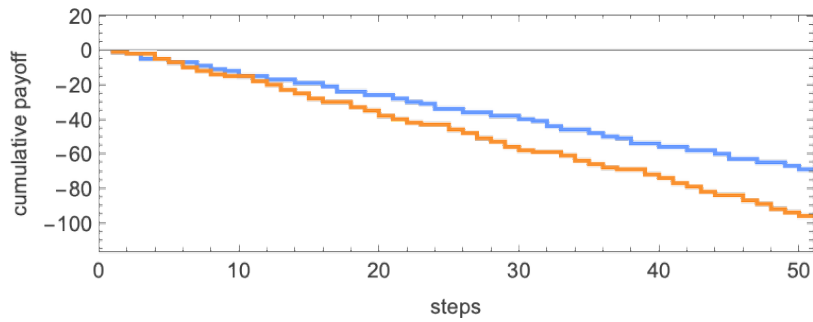
具有最复杂行为（至少按照这个标准）的规则，其平均收益接近于零。一个典型的例子是规则 11948：



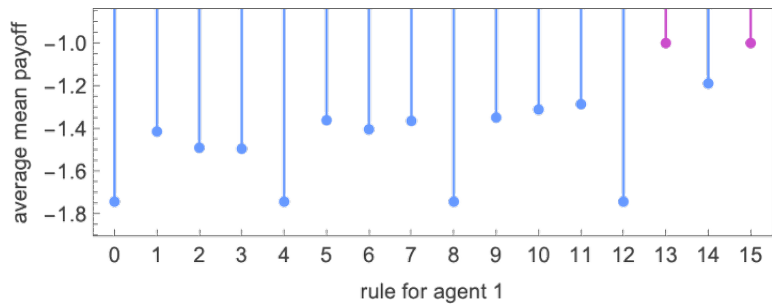
此案例中一些更为复杂的竞争包括：



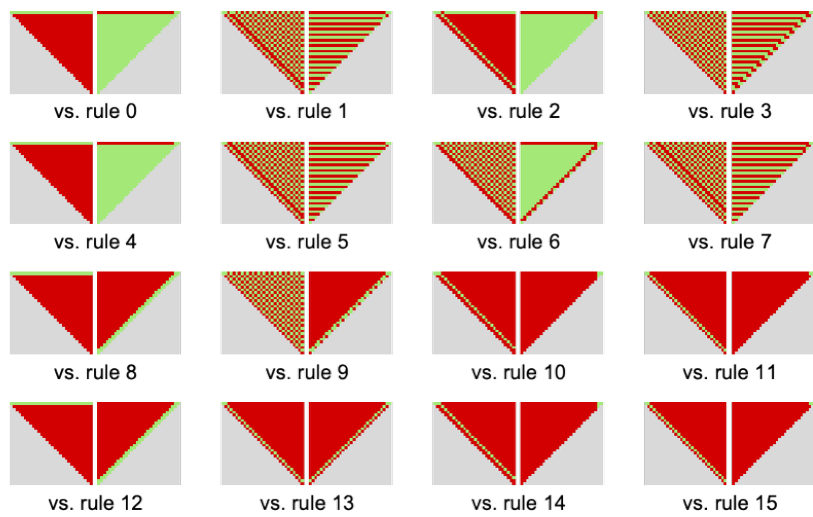
不同收益值的不同博弈又如何呢？特定规则相互竞争时的基本行为总是相同的，但它们的收益值会不同。因此，例如在囚徒困境中，双色规则 6 与双色规则 7 的累积收益现在为：



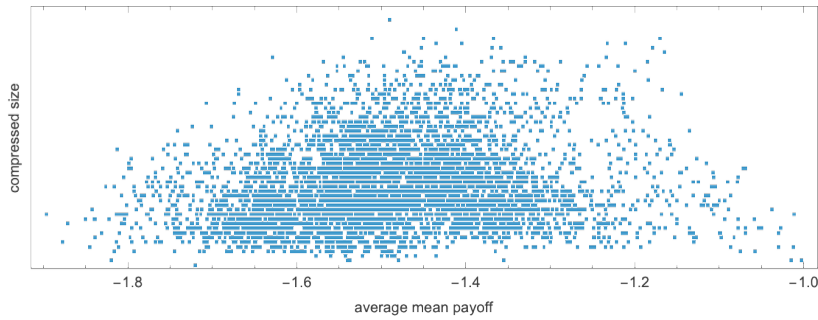
将每个双色规则与所有其他规则进行对抗，得到的平均收益值为：



规则 13 具有最高的平均收益值（-1），并且表现出相当简单的行为：



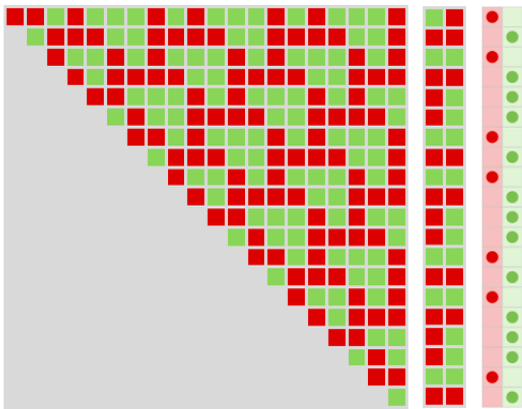
观察 3 色规则与 2 色规则游戏中的压缩大小与平均收益之间的关系，囚徒困境中高收益与更简单行为相关的现象似乎比匹配与否更为明显：



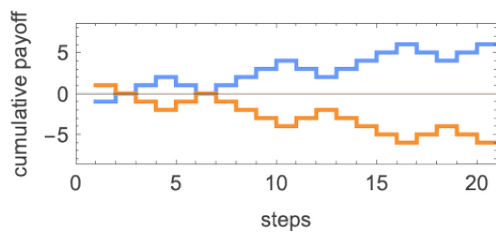
细胞自动机 vs. 有限状态机

我们已经探讨了有限状态机之间的竞争，以及细胞自动机之间的竞争。但细胞自动机与有限状态机之间的竞争又如何呢？

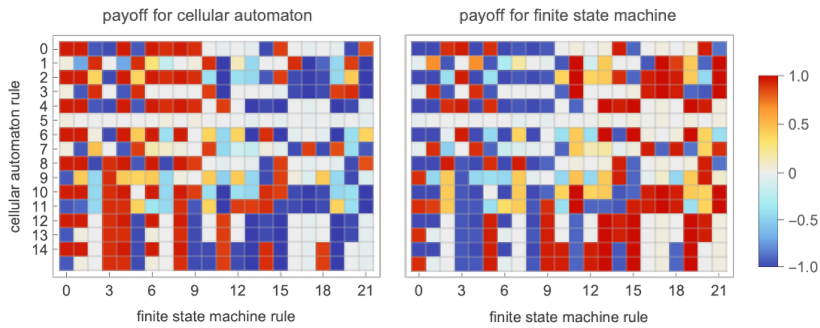
以下是一个细胞自动机与有限状态机在某特定步骤中竞争的示例：



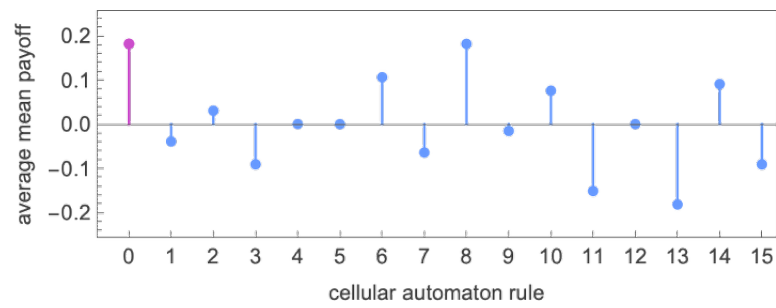
以下是这个匹配或不匹配游戏中累积的收益：



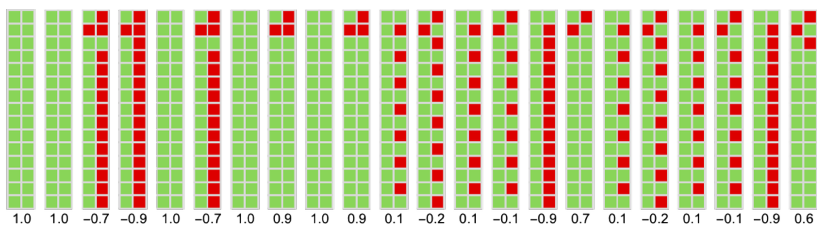
针对所有 2 态有限状态机运行此类型的所有 16 种元胞自动机规则，平均收益为：



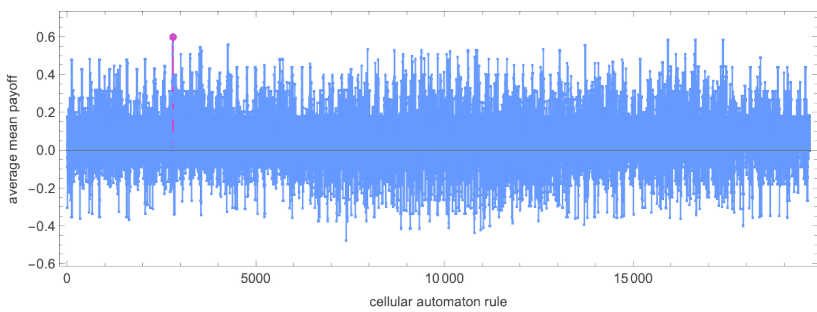
对所有有限状态机取平均后，各元胞自动机可能获得的平均收益如下：



颇为无趣的是，胜出的元胞自动机是规则 0——无论有限状态机如何行动，它都只会生成 ■ 作为响应：



其平均收益仅为+0.181。但若我们改用三色元胞自动机呢？以下是该情形下的平均收益情况——其中胜出者已高亮标出：

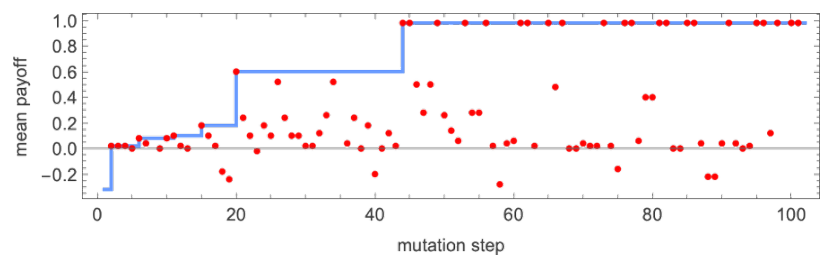


总结不同策略类型间的各类竞争可见——在与二元状态有限状态机对弈时，最具竞争力的选手以微弱优势胜出，即三色元胞自动机：

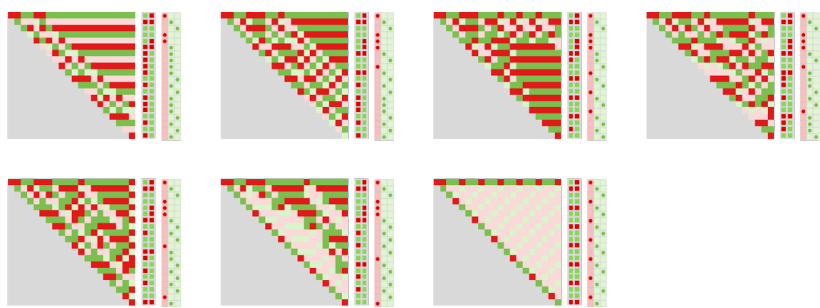
	winning value
2-state finite state machine vs. 2-state finite state machine	0.151
3-state finite state machine vs. 2-state finite state machine	0.593
2-color cellular automaton vs. 2-color cellular automaton	0.687
2-color cellular automaton vs. 2-state finite state machine	0.181
3-color cellular automaton vs. 2-state finite state machine	0.597

元胞自动机策略的自适应演化

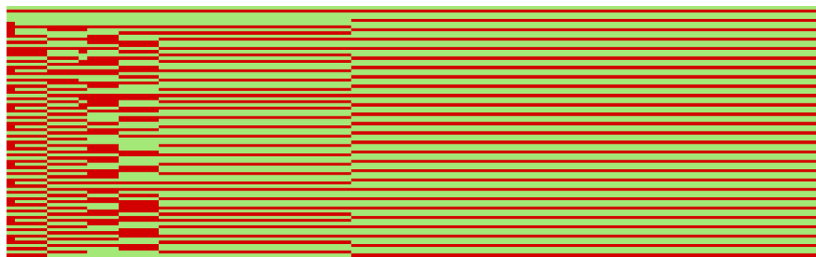
正如我们刚才对有限状态机所做的那样，我们也可以考虑元胞自动机规则的自适应演化（这也是我在其他场合广泛研究过的内容）。作为第一个案例，让我们考虑自适应演化一个 4 色元胞自动机规则，以在与上述最成功的 3 状态有限状态机（机器 1165）的对局中获得最佳平均收益。在自适应演化的每一步，我们将随机改变元胞自动机规则中 $4 \times 4 = 16$ 种情况之一，如果这次突变能使我们获得至少与之前相同的收益，则保留该突变。我们得到了一条典型的自适应演化适应度曲线，平均收益极限为 +1：



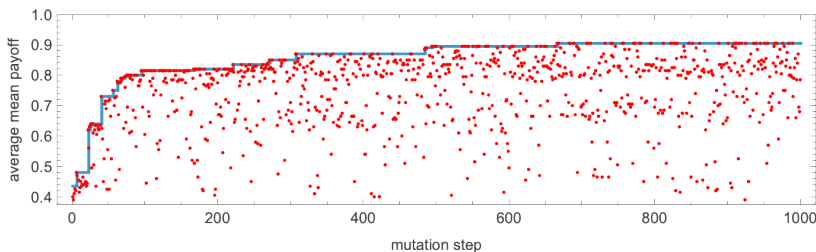
这些“突破”对应于以下规则：



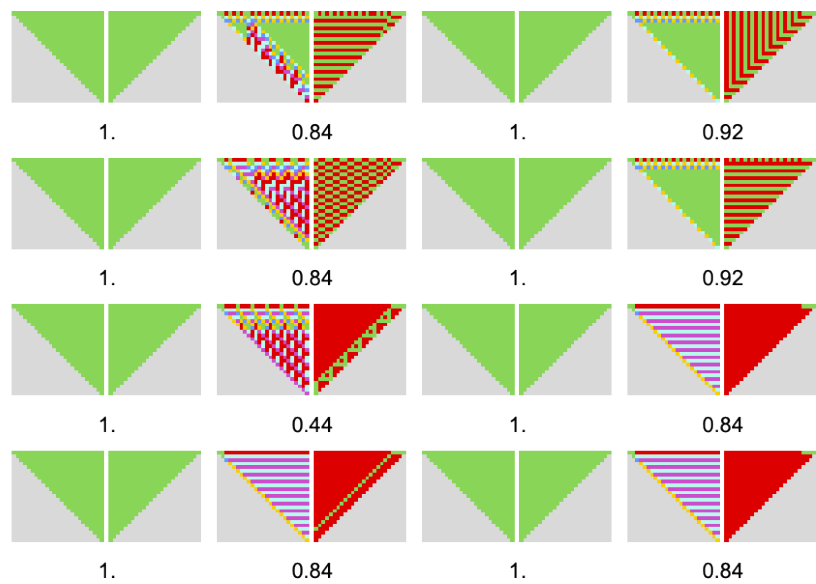
正如常见的情况，早期的突破较为复杂，但最终出现的“解决方案”表现出相当简单的行为——如果我们将连续突变步骤的结果放在一起，至少能看到一些证据：



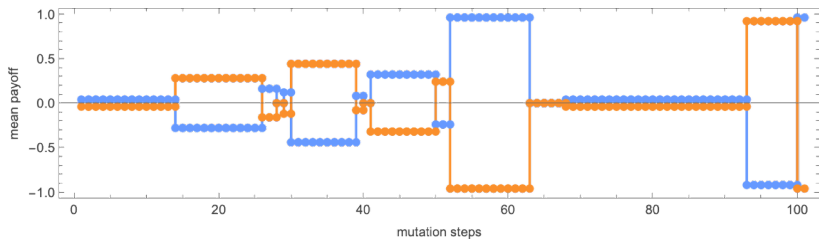
那么，让元胞自动机与其他元胞自动机竞争呢？举例来说，我们可以利用自适应进化来寻找一个 6 色元胞自动机，使其在与我们考虑过的全部 16 个 2 色元胞自动机竞争时，平均收益最大。下图是这种情况下的典型适应度曲线：



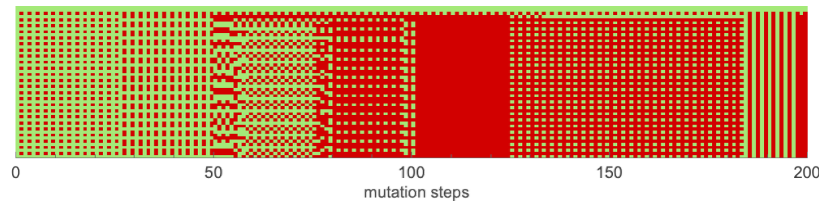
经过 1000 次突变步骤后，它达到了一条平均收益为 0.91 的规则。以下是该规则与所有双色规则竞争时发生的情况：



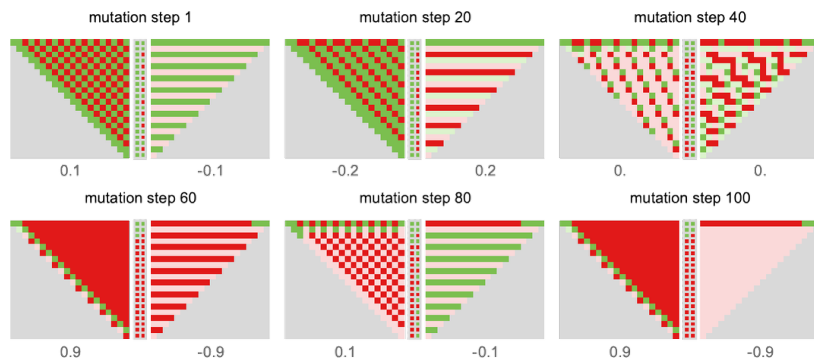
如果（如上述有限状态机那样）规则及其对手都在进行适应性进化——比如说在交替步骤中？ 以下是用一对四色规则得到的连续收益示例：



以下是相应的实际行为：



底层的元胞自动机在做什么？以下是一系列突变步骤的结果——说明适应性进化既能选择行为非常简单的规则，也能选择行为稍显复杂的规则：



图灵机策略

我们已经探讨了基于有限状态机的策略和基于元胞自动机的策略。现在让我们来谈谈基于图灵机的策略。从我们的目的来看，可以将图灵机视为在有限状态机和元胞自动机之间某种程度上的插值——尽管它们也引入了一些全新的特性。

我们的基本设定是：将对手的行动作为初始值放置在图灵机的纸带上，最新值位于最右侧，这也是图灵机读写头初始所在的位置。然后我们运行图灵机，直到其读写头移动到比之前任何时候都更靠右的位置，此时我们根据初始读写头位置出现的值来确定下一步行动。

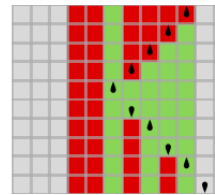
例如，考虑一个由图灵机定义的规则：



那么假设到目前为止对手的动作序列为：



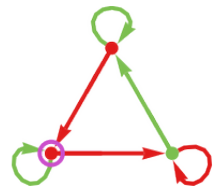
以此作为初始条件运行图灵机，我们得到以下结果：



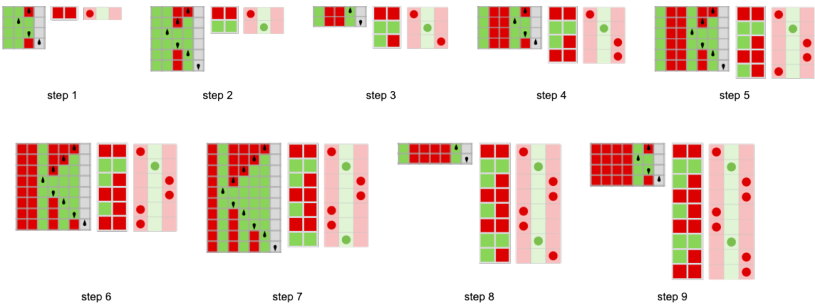
据此，我们可以根据“图灵机策略”读出“下一步动作”，本例中为 ■。

在我们有限状态机和元胞自动机的设定中，游戏每一步只执行一次演化。而在图灵机的设定中，游戏每一步都需要运行图灵机，直到磁头向右移动超过初始位置为止。

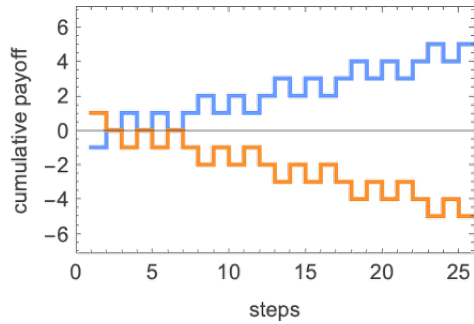
以下是当我们选取一个具体的 3 态有限状态机样本时的情况



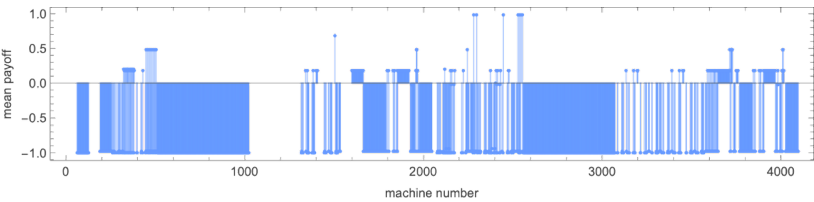
并将其与上述图灵机进行对抗：



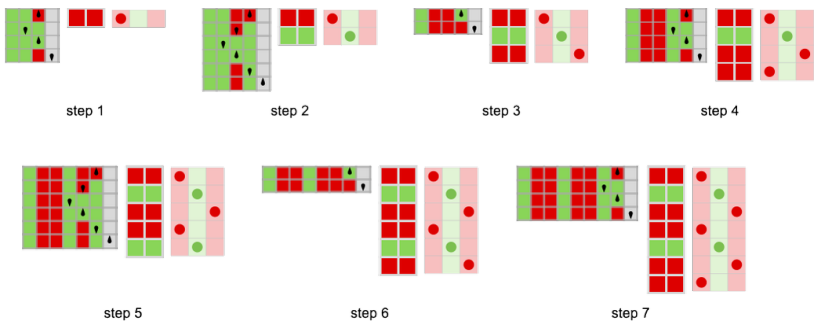
在“匹配与否”游戏中，这里的累积平均收益为：



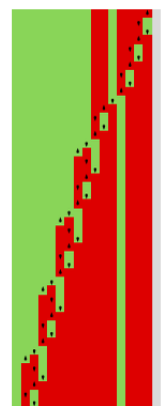
我们这里使用的图灵机共有 4096 种（状态数 $s=2$ ，颜色数 $k=2$ ）。将每种图灵机与我们的 3 状态样本机进行对局，所有图灵机在“匹配与否”游戏中的平均收益如下：



有几种图灵机的极限平均收益达到+1。例如，机器 2529：



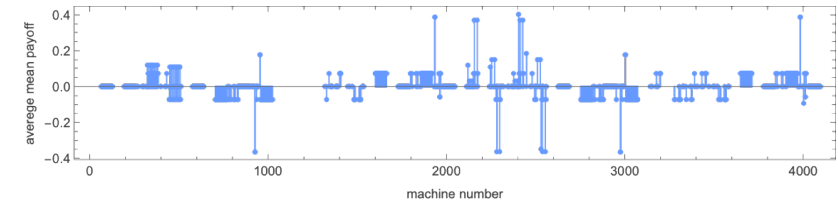
但这里出现了一个棘手的问题。我们的图灵机策略是通过运行图灵机直到其磁头比初始位置更靠右——从而认为它停机。但如果它从未停机，比如：



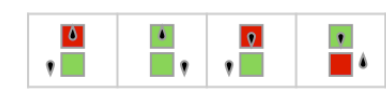
就我们的目的而言，我们只是说明在这种情况下，收益是未定义的。如果在某个特定博弈中出现了

这种未定义的收益，我们就认为整个博弈的平均收益也是未定义的——从而在上图中留下了一个缺口。

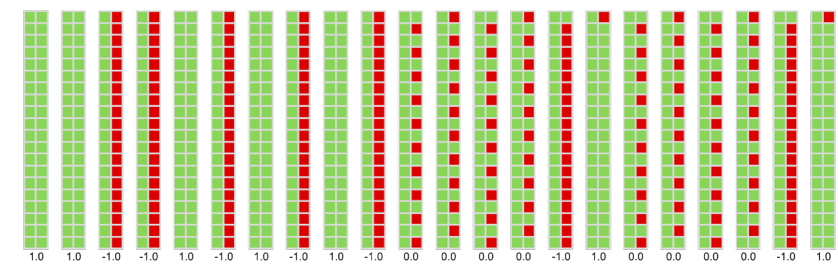
如果让图灵机与所有不同的 2 状态有限状态机竞争，结果会怎样？下图展示了该情况下的平均收益（缺口对应那些不会停机的机器）：



图灵机 2403 达到了+0.4 的最大值



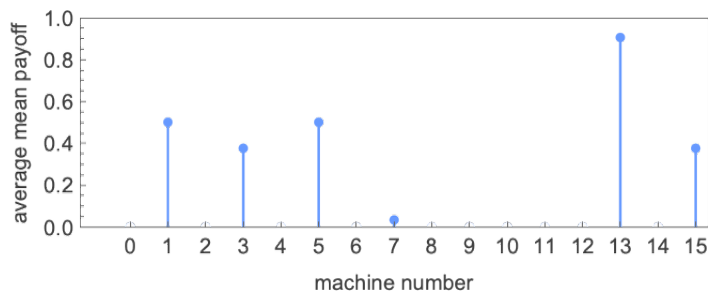
当它与 22 个不同的 2 状态有限状态机分别竞争时，会产生以下行为和极限收益：



那么，图灵机与图灵机相互竞争时会怎样？为了便于管理，我们可以考察单状态图灵机（即 $s=1$, $k=2$ 时仅有 16 种）。将它们两两配对运行，得到的平均收益矩阵如下（灰色条目对应其中一台图灵机未停机的情况）：



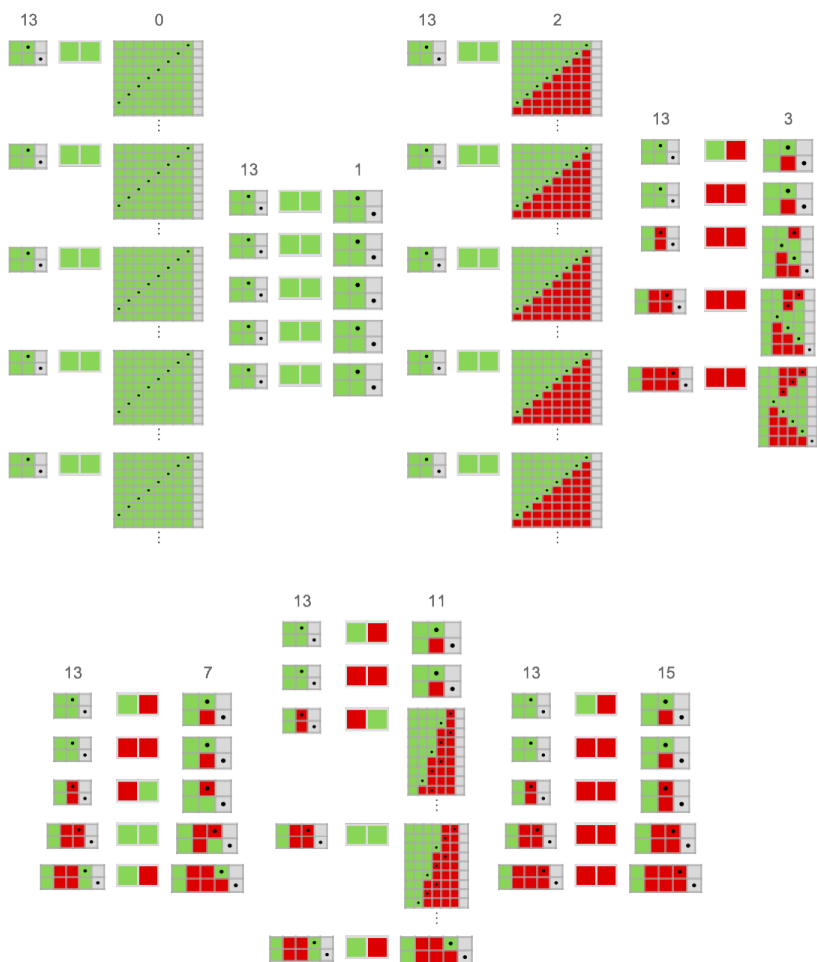
每台图灵机的平均收益为：



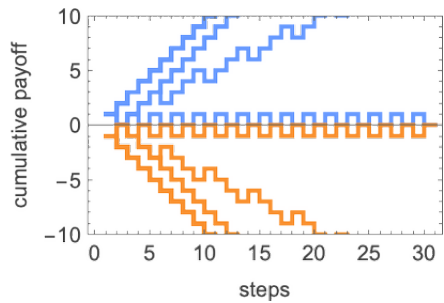
这些机器中的“赢家”是图灵机 13：



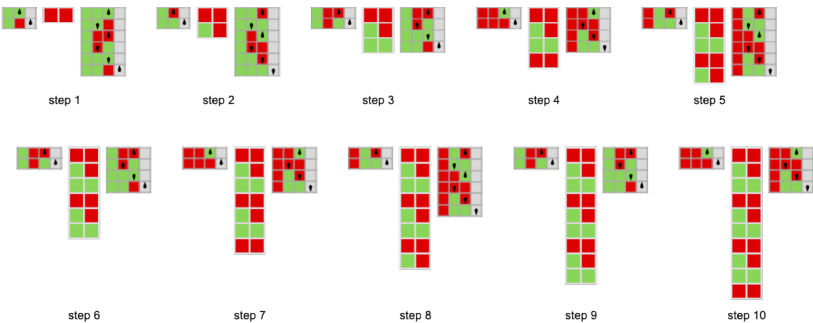
将这台机器与所有其他 $s=1$ 、 $k=2$ 的图灵机对弈，我们得到的行为如下：



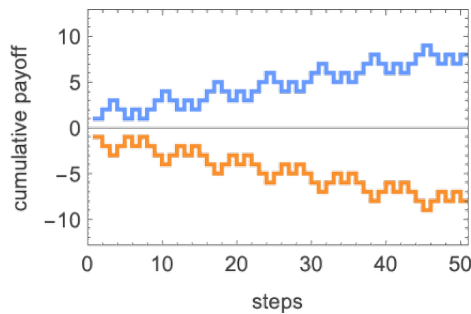
如果我们观察累积收益，会发现许多博弈的平均收益趋近于 1，尽管有些并非如此，最终平均收益约为+0.81：



两个状态的图灵机之间的典型竞争是



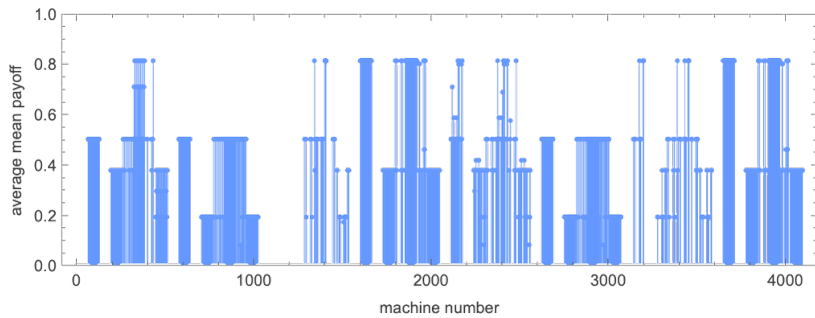
这产生了一个稍微更复杂的累积收益模式：



如果两状态和一状态的图灵机竞争会发生什么？ 以下是所有 4096 个两状态机与 16 个一状态机对抗的平均收益数组：



两状态机的平均收益如下——同样最大值是 0.81：



讨论

我们现在已经看到了许多竞争规则学的例子。也许最重要的是，当我们从规则学的角度审视特定类型的所有可能程序时，即使所有涉及的程序都很简单，竞争运作的图景也相当复杂。

从某种意义上说，这是计算不可约性的典型结果：要了解程序之间的竞争将如何发展，基本上别无选择，只能运行它们并观察结果。

有时，获胜的程序以非常简单的方式获胜——实际上是“利用简单的技巧”。但在其他情况下，事情更为复杂。有时两个竞争程序都表现出复杂行为，从某种意义上说，其中一个会“恰好”获胜。但有时获胜会更加系统化。通常这是因为这种行为有效地嵌入到某个计算可约性的口袋里，从而系统地击败了特定类型的对手。

我们之前主要研究了极为简单的程序，它们在某种意义上必然对所有竞争者“暴露相同的规则”。但特别是当我们面对的竞争者集合规模较小时，一个足够大的程序实际上可以对不同竞争者展现其规则的不同部分，从而拥有一种“定制化子策略”，针对不同的潜在竞争者分别获胜。

在研究策略的自适应演化时，我们经常处理更大的程序。通常我们看到，自适应演化在寻找获胜策略方面相当成功。但——正如自适应演化中常见的情况——生成策略的“机制描述”并没有显而易见的方式。相反，它更像是我们在自适应演化的其他研究中看到的那样：演化过程将某些“不可还原计算的块”组合在一起，在我们这里，这些块实际上“恰好”在竞争中获得成功。

不同的博弈，对应于不同的收益模式，会产生细节上有所不同的结果。如果构建关于博弈过程的详细叙述，不同博弈的叙述可能看起来截然不同。但在整体层面上，不同博弈之间似乎存在着显著的相似性——关键现象看起来非常一致。

这一切对于存在代理方竞争的实际情境意味着什么？一方面，通常很难“事先预测”或“证明一个定理”来说明什么是最佳策略。计算不可约性足够强，以至于我们基本上只能尝试运行不同的竞争，看看会发生什么。某种意义上，我们在这里看到的极其多样化的行为正支持了规则学研究至关重要

的观点。仅仅找到可能策略的简单参数化不足以准确了解所有可能发生的情况。别无选择，只能系统性地列举某种“所有计算可能的策略”。这正是我们在规则学研究中可以做到的。

是的，我们在这里所做的只是浅尝辄止地研究了竞争的规则学。首先，可以扩大程序规模，观察会出现哪些新现象。可以预期，大多数情况下事物将保持相同——计算不可归约性将成为主导力量。但也可能会出现新的、意想不到的可归约性区域，或许每个区域都有其独特的“通往竞争成功的路径”。

人们还可以设想研究不同种类的计算系统——它们作为适合不同应用的元模型。计算等价原理表明，整体结果将具有一定的普遍性。但细节会有所不同。而这些细节可能很重要，特别是在解释截然不同领域的结果时。即使对于竞争的终极目的而言，有限状态机或元胞自动机能很好地捕捉关键要素，但从微观生物学、人类决策、社会互动、AI 竞争等途径达到这些的方式可能非常不同。

历史与个人笔记

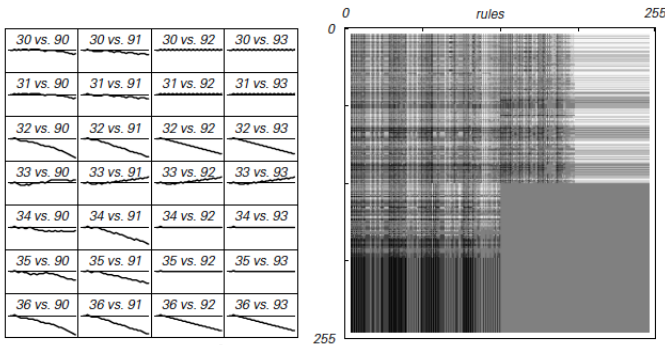
游戏的形式化研究有着悠久的历史——事实上，组合学和概率论等领域的早期发展很大程度上正是由游戏所驱动的。现代称为博弈论的学科兴起于 20 世纪 40 年代，主要关注在特定收益模式下如何制定最优策略。其核心思想通常是通过多次实验取平均值，分析每位玩家进行单一行动（尽管可能是概率性的）时会发生什么。对于这类设定，数学上已推导出相当完整（尽管有时较为复杂）的结果（例如，现已在 Wolfram 语言中实现）。但至于我们一直在讨论的重复博弈或迭代博弈呢？在博弈论早期，曾有讨论将策略定义为从历史到行动的任意映射，并证明了各种相当抽象的数学结果——尤其是在经济学应用中。

但到了 20 世纪 70 年代，开始出现这样一种观点：应将个体行为者建模为具有“有限理性”，并对应于有限的计算系统。到 70 年代末，人们开始通过计算机实验来研究相当于简单程序之间的竞争。一个显著的例子是鲍勃·阿克塞尔罗德组织的囚徒困境博弈锦标赛。在这场锦标赛中，不同参与者提交了一系列特定程序，让它们相互对抗。结论是“以牙还牙”策略（可视为一种有限状态机）表现最佳——这一结果后来被广泛用于强调合作的价值等。

我必须承认，我一直对这个结果持怀疑态度。仅仅考察人们碰巧提交到锦标赛中的程序，这种做法似乎非常不科学。为什么不系统地枚举所有可能的程序，看看会发生什么？在我自己的工作——从 20 世纪 80 年代初开始——中，我经常这么做，特别是针对元胞自动机。我总觉得博弈论的设定有些随意且琐碎，而且我在研究单个程序的行为时已经发现了太多令人应接不暇的新现象，无需再让它们相互竞争。不过，最终在 20 世纪 90 年代中期，我还是考察了一系列可能的程序（当时是元胞自动机）相互竞争时会发生什么。我在著作《一种新科学》末尾的一则简短注释中总结了这一结

果：

■ **Games between programs.** One can set up a game between two programs generating single bits of output by for example taking the input at each step to be the concatenation of the historical sequences of outputs from the two programs. The pictures below show what happens if the programs operate by applying elementary cellular automaton rules t times to $2t + 1$ inputs. The plots on the left show cumulative scores in the Evens and Odds game; the array on the right indicates for each of the 256 possible rules the average number of wins it gets against each of the 256 rules. At some level considerable complexity is evident. But the rules that win most often typically seem to do so in rather simple ways.



我一直打算回头更详细地研究这个问题。最终，我最近在生物进化基础方面的工作让我觉得是时候付诸行动了。我发现有一些文献使用有限状态机之类的模型作为迭代博弈的策略。但据我所知，我想象中的那种系统性的规则学研究从未有人做过。这就是为什么我最近决定，终于到了该做这件事的时候了.....

谢谢

感谢沃尔夫勒姆研究所的 Willem Nielsen、Brian Ashiundu 和 Júlia Campolim 的大力帮助。我们暑期项目的几位参与者对我提出的程序间游戏项目做了研究：Rodrigo Bazaes、Kantaporn Danchaivijitr 和 Aziz Sahibnazarov。多年来，我与许多人讨论过博弈论及相关思想，包括 Brian Arthur、Bob Axelrod、Seth Chandler、Roger Germundsson、Paul HARRALD、Jozsef Konczer、Pedro Marquez-Zacarias、Eric Maskin、Zsombor Méder、Chrystopher Nehaniv、Scott Page、Jordan Pollack、John Maynard Smith、Stan Reiter、Nassim Taleb、Valeriu Ungureanu 和 Marc Vicuna。（著名博弈论学者 John Nash 长期使用现在的 Wolfram 语言，并参加过相关会议，但我个人从未见过他。）

相关文章：“Games and Puzzles as Multicomputational Systems” »

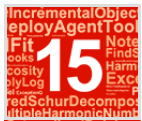
引用格式 >

Stephen Wolfram (2026), "程序之间的博弈：竞争规则学," Stephen Wolfram Writings.
writings.stephenwolfram.com/2026/06/games-between-programs-the-ruliology-of-competition.

发布于：人工智能、计算科学、生命科学、规则学

Join the discussion

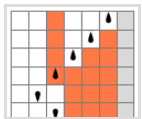
相关文章



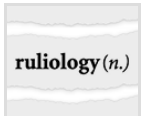
发布 Wolfram 语言和 Mathematica 15 版本：内置（有用的）AI 及大量新核心功能
June 16, 2026



使 Wolfram 技术作为 LLM 系统的基础工具可用
February 23, 2026



P vs. NP 与计算难度：一种规则学方法
January 30, 2026



什么是规则学？
January 12, 2026

热门分类

人工智能

上

入门

公司与商业

计算科学

计算思维

数据科学

教育

未来展望

历史视角

语言与交流

生活与时代

生命科学

Mathematica

数学

一种新科学

新技术

个人分析

哲学

物理学

规则学

软件设计

Wolfram|Alpha

Wolfram 语言

其他

按年份的文章

2026 | 2025 | 2024 | 2023 | 2022 | 2021 | 2020 | 2019 | 2018 | 2017 | 2016 | 2015 | 2014 |
2013 | 2012 | 2011 | 2010 | 2009 | 2008 | 2007 | 2006 | 2004 | 2003 | 所有

