第五堂课

复杂自动机的一些考量

——层次和进化的问题

摘要/关键词：

- 自动机可以作为一个整体来研究，也可以对各个零件分别研究。当然，有了对于各个零件的知识以后，我们还需要懂得如何把零散的知识组成统一的理论，但是目前我们还不知道怎样做。
- 问题一：我们这里先不谈细节问题，而是仅仅针对中继组件（relay organs）的性质进行讨论。
- 问题二：如何与自动机和信息理论保持一致，我们将重新考虑在第二堂课结尾的地方已经触及到的将程序看作一种自动机模型的理论探讨。
- 如何合成一个自动机，并且它自己就能实现这一合成过程。
- 有关 “复杂度”的直观理解。我们猜测这个复杂度具有一种衰退的特征，这种特征是与该自动机过程的描述以及由它完成的自动机合成有关的。
- 有关复杂度衰退这个概念的性质和难点。
- 严格的讨论：自动机和其“基础”零件。有关这些基础零件的定义和列表。由自动机合成的自动机。自我复制的问题。
- 与此有关的构造性自动机的主要类型：通用指令的概念。能够执行指令的通用构造自动机。通用拷贝自动机。组合自动机来实现自复制。
- 自复制和其他类型的自动机合成过程：例如催化过程，同已知的主要遗传和变异机制的比较。

一、自我创生的自动机

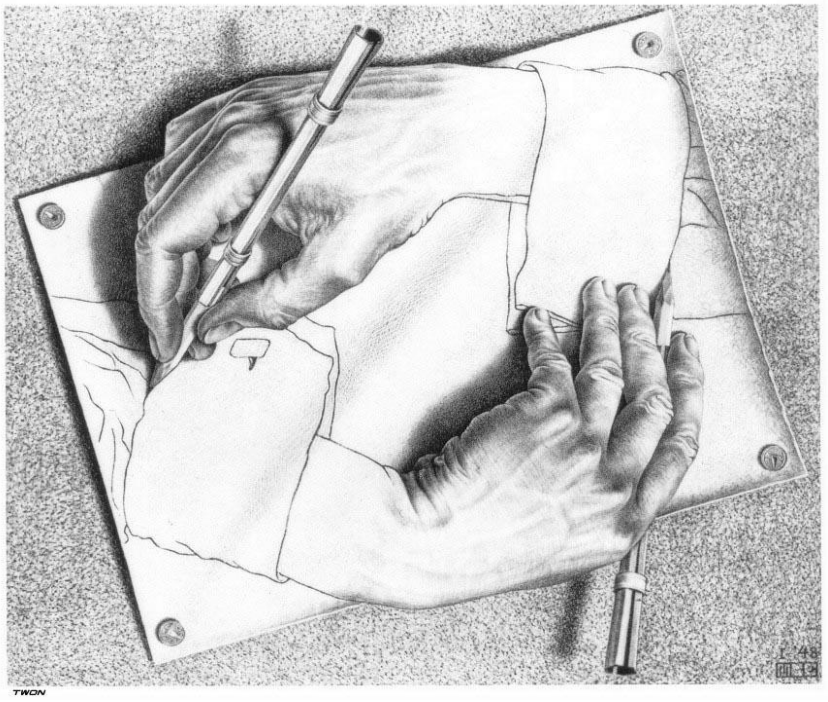
在前几堂课中，我们讨论的自动机都不是直接对自身进行操作的，因此它们产生的输出与自动机自身具有完全不同的性质。在我提到的三个例子中，这点都很明显。

例如，图灵自动机就可以看作一个包含有限状态的盒子，这个盒子的输出是储存在另外一种实体上，可以简单称为打孔纸带。这条纸带本身并不像图灵机一样具备不同的状态，并能够在状态之间来回切换；此外，与有限状态的盒子不同的是，我们假定纸带是无限长的，因此可以包含的状态也是无限多的。所以，这条纸带从性质上说，同在纸带上打孔的自动机是完全不同的，也就是说，自动机是在完全不同性质的介质上运行的。

对于 McCulloch-Pitts 的自动机模型来说，情况也一样。这里的自动机是由神经元组成的，并且能够向外界产生脉冲信号。这意味着，自动机的输入输出不是神经元本身，而是神经脉冲。当然，这些神经脉冲可以进入周边的组件，并导致完全不同性质的反应。虽然神经脉冲也可以输入到运动系统或者内分泌器官中，导致机械运动或者化学物质的合成，这些输

入输出的性质同自动机本身，也就是神经元仍然完全不同。

最后，对于计算机器来说，这个结论也是完全适用的。计算机器可以看成一种被“喂食”并且“吐出”纸带一类介质的机械。不管这种媒介是打孔卡片、磁化的钢丝，还是刻录了许多条平行磁性轨道的磁带，或者是包含黑点的电影胶片，它们都是储存信息的介质，或是用来喂给自动机，或是由自动机产生。这些媒介本身的性质则是和自动机完全不同的。事实上，自动机根本没有产生任何媒介，而是对与自动机本身非常不同的媒介进行了修改。很容易想象在另一种情况下，有一台计算机器输出某种脉冲信号，用来控制完全不同的其他实体。然而即便如此，自动机仍然是同它输出的脉冲本质不同的。在所有这些情况之下，媒介和自动机都存在着实质性的差异。



画手（译者加）

图片来源：《魔镜——艾舍尔的不可能世界》

如果要对自动机的性质进行彻底的研究，我们必须开阔视野，让我们考虑以下的问题：如果自动机输出的是自动机本身的话，这将会怎样？当然，当我们谈论到这个问题的时候就需要小心了。物质上的“无中生有”当然是不可能的。但是我们可以想象在某个环境中有很多零件，自动机可以从中拣起一些零件装配成更复杂的设备；当然，也可以把已有的设备拆散成零件，从而修改成类似于它自己的东西。为了使讨论更清楚，我们需要清晰列出我们所需要的所有基本零件，假设这些零件漂浮于一个大容器里面，并且每个零件的数量都是无穷的。接下来，假设在这个容器中间还生存着一台自动机，它也游弋于这个池塘中，它的主要活动就是不停地收集各种零件，把它们组装在一起；或者把已经组装好的设备拆散成基本零件。

以上对于这种生命的“公理化”定义体系，当然是略嫌粗线条了一些。的确，这样简单地看待一个复杂的问题，当然存在着很大的局限。但是这些局限恰恰就是公理体系本身内禀的局限。我们用这种“粗线条”的体系能够得到的结果，实际上完全取决于我们如何定义基本零件。通常来说，不存在一套确定的规则来指导我们如何选择公理体系中的基本单元，因

此这完全取决于体系设计者的常识判断。很难确切地解释为什么一个设计合理，另一个设计却不合理。

首先要排除的一种做法是，我们把每个零件定义得很大，具有很复杂的功能和各种联系。比如就把零件定义成活着的生物体，那么这样就把问题“定义没了”。因为我们的目的就是要通过零件的组合，来描述和理解生命的功能；现在既然定义本身已经包括这些功能在内，那显然就没有什么可以研究的了。所以，如果零件定义得过大，每个零件包含的功能太多太复杂，这样就丧失了问题的意义。

另一方面，如果把零件定义得太小，比如说，规定零件不能大过一个分子、一个原子或者其他的基本粒子，这也不能很好地体现我们所要研究的问题。因为这种情况下，我们就把问题过于还原了。这方面的问题当然也是很重要，很有趣的，然而却同我们打算研究的自动机完全无关。因为我们感兴趣的问题是复杂的生命结构是如何组织起来的，而不是去用量子力学计算化学键能和物质结构。所以，通过以上常识性的分析，大家应该已经理解，研究对象既不能定义得太大，也不能定义得太小。

即使选择了恰到好处的零件尺度，还是有很多种不同定义的方式，也很难说自然地哪一种就比其它的更好。在形式逻辑中，也有过类似的困难：整个系统取决于公理的选择，但是没有确定的法则来规定公理应该如何选取，我们只能依靠我们对于要研究系统的一些常识，并且尽量保证不要把这些公理直接定义成这些问题本身，也不要把其他领域的问题牵扯进来。比如说，在对于几何学建立公理体系的时候，我们应该把集合理论的定理当成现成的直接用。因为我们并不关心集合与数字的转换，也不关心数字与几何的转换。同样，我们也不能将更复杂的比如解析数论中的定理作为几何学的公理运用，因为这样“抄近路”，就导致我们的研究失去了意义。

退一步说，即便公理的设计符合常识，如果由两个不同的人独立地来做这件事，结果往往仍然会大相径庭。比如在形式逻辑中，名词符号的数量几乎和作者一样多，任何人只要使用其中一套符号系统一段时间，就会觉得自己的这套系统会比别人的略高一筹。所以，对于公理体系的应用，零件和符号的选择既非常重要，又非常基本，因此怎样的设计才称得上确切，就不是一件能够严格地判别或者精确地比较的事情了。下面我会说明我对于系统设计的看法，但是我要强调，这只是一个相对主观的陈述。

首先我们想要提出“肌肉”的概念，就像神经元对于大脑一样，肌肉也是自动机的基本零件。也就像 McCulloch 和 Pitts 利用神经自动机理论来抽象实际的神经元一样，在这个定义里，肌肉零件具有连接或者断开连接，提供能量等内在功能。那么，我们就可以在这个定义框架之内，相当简便地描述肌肉、连接组织、断开组织、以及提供代谢能量的组织等，而不用陷入到这些组织的实现细节中去。按照这套思路，约需要 10-15 种这样的基本单元。

通过以这种方式来定义自动机，我们已经把问题的一半“丢到窗外”了，而且这可能是比较重要的那一半。我们已经放弃了解释这些零件是如何由实际的基本粒子或者化学中的大分子构成的；而且也放弃了对那些令人着迷的关键问题，譬如自然中的生命零件究竟是如何构成自身的？为什么这些结构有时候是大型的分子，有时候却是大分子的复合？又为什么细胞的尺度总是在微米和分米之间？这对于基本零件来说，是一个很奇怪的尺寸，它离物理上真正的基本尺度，至少还有五个数量级的差别。

这堂课我不打算解释这些问题，而将简单地默认具有这些功能的零件已经存在。那么我们希望回答，或者至少可以进行探索的问题就成了：把这些零件组合成具有功能的有机生命的过程，究竟遵循怎样的规律？这样性质的生命具有怎样的特点，尤其是具有哪些定量特征？以下的讨论仅限于本范围。

二、关于复杂度

【从这里开始，冯纽曼谈到了信息、逻辑、热力学以及各个参数之间的平衡关系等问题。按照冯纽曼的讲义安排，这部分内容现在被放到了第三堂课的结尾部分。这些内容同自动机的联系主要在于，冯纽曼将要介绍的“复杂度阈值”概念是属于信息论范畴的。】

复杂度的概念对于我们的讨论是很有用的，但是我们现在也只有一个直观的、模糊的、不全面的、也不太科学的了解。它显然属于信息论这个主题，同热力学领域的知识也有相当的联系。我不知道应该怎样命名这个概念，所以就不妨就把它叫做“复杂度（complication）”吧。这里的复杂度就是指复杂的有效程度，或者说是做事情的潜力。这里我说的并不是一个具体对象牵涉到的复杂程度，而是它有目的地去做事的时候牵涉到的复杂程度。从这个意义上说，具有最高复杂度的对象就是那些可以做很困难的，牵涉很多事情的东西。

我们之所以这样刻画复杂度是因为在研究那些主要功能就是把基本零件组装成其它机器的自动机的时候（包括生命本身和人工的机械自动机），会遇到一件特别的奇事：似乎我们的心智会观察出两种不同的图景（Mind）。你会在这两种不同的图景中切换，也能根据某种图景得到一个显而易见的结论，但是这些结论却是截然相反的！

任何人只要对生命稍加观察，便会知道生命可以复制同自身相似的其他生命。这是生命的最常规的功能，如果没有这个功能的话，生命便根本不会存在；或者可以说，恰恰因为复制，才使得生命无所不在。从另一方面来看，生命是由基本零件构成的非常复杂的组合，从概率论或者热力学的角度看，这种组合的出现是极不可能的。生命居然能够存在，这件事本身就是一个了不起的奇迹；而唯一能够使得这个奇迹显得不那么神奇的解释是：生命可以复制自身。因此，如果由于某种特殊原因，一个生命偶然出现了，那么，从此以后，生命就不再被概率法则所束缚，只要环境合适，更多的生命就会跟着出现。然而，从热力学的角度讲，这种“合适的环境”，比起生命本身的存在几率已经要高出很多了（But a reasonable milieu is already a thermodynamically much less improbable thing）。所以，从某种程度上说概率运算在这里存在着一个**漏洞**（loophole），而自我复制的过程恰恰正是利用了这个漏洞。

不但如此，比起单纯的自复制，自然界中的生命更胜一筹，因为随着时间的流逝，生命会变得更加精巧。今天的生命是从那些非常简单的生命发育进化而来的。实际上，生命开始的时候是如此简单，很难想象后来的任何复杂生命的描述，已经被包含在更早的生命之中了。基因这种复杂度比人低一个数量级的东西，是如何蕴含了如此复杂的人类个体的信息的呢？不过也许你会说，因为基因仅仅在人体之中的时候才能起作用，因此它可能并不需要包含将要发生的事情的全部描述，而只要提供几个标记来代表可能的选择就行了。但是，在发育进化史中情况就没那么简单了，因为没有现成的生命个体可以利用。我们知道，一切生物都来自无生命混沌环境中简单的个体，它们逐渐演化出更复杂的东西，这些生命具有产生比自己更复杂之物的能力。

另一方面，当我们分析人工自动机的时候，却可以得到截然相反的结论。大家都知道通常一台机器总是比它能够制造出的零件更复杂。因此，一般地说，如果自动机 A 能制造出自动机 B 的话，那么 A 一定包含关于 B 的全部信息，这样 A 才能按照这些信息把 B 制造出来。如此这般，我们就发现，自动机的“复杂度”，或者说它的生产潜力，是不断降级的。也就是说，一个系统的复杂度总是比它制造的子系统要高一个数量级。复杂度是不断降低而非升高的，这个分析人工自动机得到的结论，和上面的分析生命本身得到的结论完全相反。

然而我认为，如果将我们对人工自动机的各种知识综合起来，并考虑自动机组合起来产生的效果，就能够消解这个悖论。在这里，我们并没有向大自然寻找答案，这是因为我们对自然界的生命了解还很少。而另一方面，因为我们自己设计了自动机，所以我们完全了解自

动机的性质。不管是实际的人工自动机还是抽象的公理体系所描述的自动机，我们都有足够的信心来设计一种可以复制它们自身的机器。至少从原则上，我们可以说明，为什么从表面上看复杂度的衰退不可避免，但是实际上却不一定如此；并且，复杂的个体的确是可以自动地被比它简单的个体制造出来的。

我们的结论是这样的，存在着一个复杂度的阈值，如果系统低于此阈值则它的复杂度就会衰退。这个结论是完全符合我在之前讲座中曾多次提到的形式逻辑中的一些结果的。虽然我们现在对于什么是复杂度以及如何测量复杂度还不甚了解，但是我认为即使我们用最粗糙的衡量标准，也就是系统中所包含的零部件的数量来衡量系统的复杂度的话，本结论仍然成立。如果零件数量少于某一个限度的话，复杂度就会不断下降，也就是说，自动机只能制造比自己简单的自动机；而如果高于这个限度，自动机才能造出同样的，或者更加复杂的新自动机出来。至于这个阈值的具体大小，它就取决于我们该如何定义基本的零件了。在合理定义零件的前提下，我想这个数字也不算小。比如我接下来要说明，如果用十几或者二十几种简单的零件来构造自动机，那么实现自复制至少也需要几百万个这样的零件。虽然我还没有研究其中的细节，但我猜想这样的机器在不算太遥远的未来是会通过我们的艰苦努力而制造出来的。

复杂度阈值是一个决定性的临界点，低于它，组装生成自动机的过程就会走向衰退；而一旦超越了这个临界点，组装进化在适合条件下就会发生爆炸性的突变，也就是说每个自动机所制造出来的新自动机都比自己更加复杂，更加具备潜能。

到现在为止，不管怎么解释，这些东西还显得很模糊，因为究竟什么是复杂度，一直没有定义清楚。但是反过来，如果我们不清楚一些关键实例的细节，也就是一些结构究竟是如何展现出复杂度的那种悖论性质的，就不可能给出复杂度阈值合理的定义。不过，这样的困难并不是第一次出现，在物理学发展过程中，人们也曾遇到过类似的问题，如守恒和非守恒量、能和熵等关键概念的提出等。人们必须先对简单的热机和实际机械问题作大量讨论，才能正确地抽象出能和熵的概念。

三、自动机的构建

【冯纽曼只是简单地提了一下他打算使用的原件的种类。这些零件和 McCulloch-Pitts 的神经模型很像。有些零件是“除了在两端之间建立刚性的几何连接以外没有任何功能的”；另一种零件被称为“运动组件”，起到“类似肌肉的作用”，这种零件在受到刺激时长度会收缩为零；还有一种零件遇到脉冲就“建立或者断开一个连接”；他说至多只需要十几种这样的零件就足够了。由这些零件所组成的自动机能够自动地捕捉偶然撞上来的零件：“我们可以发明一种装置”使它能够识别出捉住的零件类型。

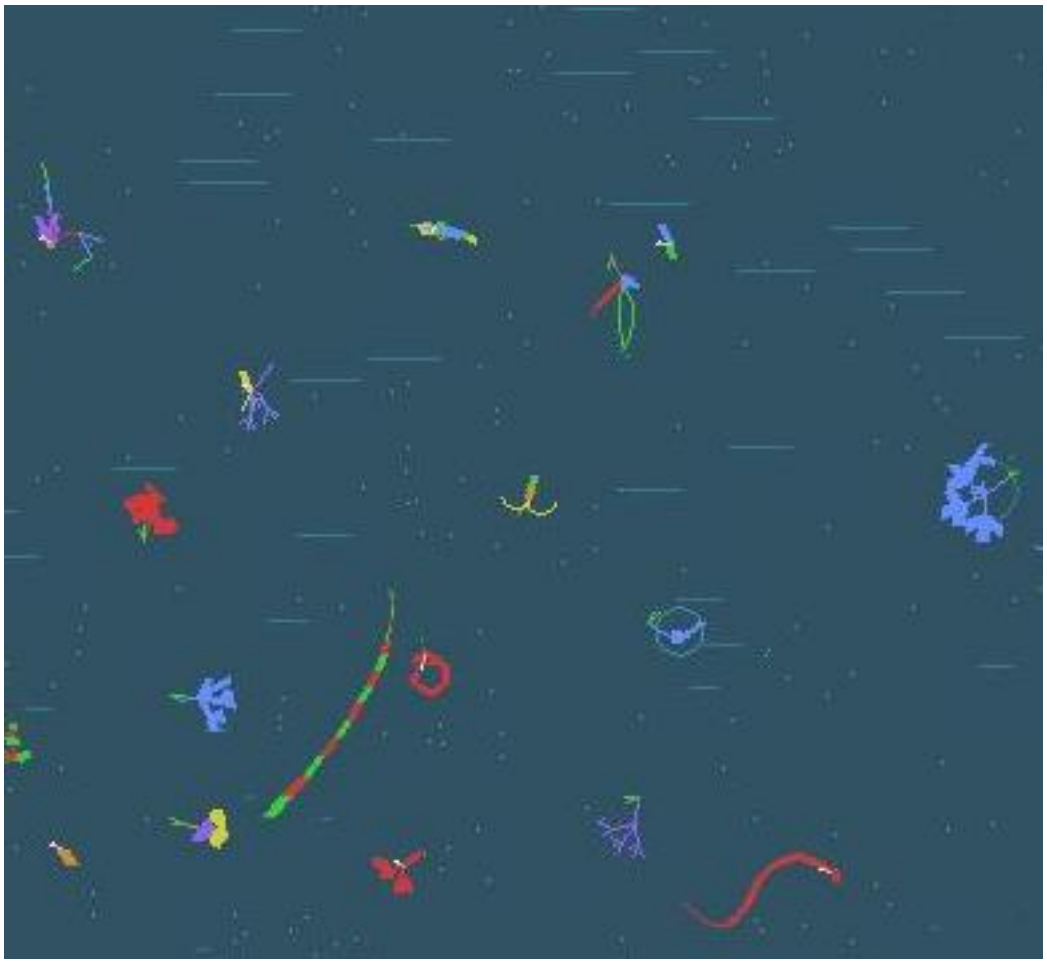
在 1948 年 6 月，可能是为了准备 9 月份的希克森会议，冯纽曼在普林斯顿的高等研究院（IAS）给他的一些朋友讲了三堂有关自动机的课。据我所知，这几堂课包括了冯纽曼对于自复制自动机的最详细的分析。因此，编者做了一番努力，尽量从听课的笔记和与会者的记忆中拼凑出关于零件以及它们的功能的细节，兹叙述如下：

冯纽曼描述了八种不同的零件，并用直线来代表它们，且标出了位于中间或者两头的输入和输出端子。自动机在离散的时间下运行，每个零件都要用一个单位的时间做出反应。我们不知道由八种零件构成的这个列表是否完成了冯纽曼的意图，但编者猜想，就此问题，他还没有最后得出结论。

有四种零件是用于处理逻辑和信息的：应激零件（Stimulus organ）是用于接收和发出刺激信号的，并且接收这些信号是相分离的，也就是该零件的真值函数相当于“p 或 q”。合并零件(Coincidence organ))则实现“p 与 q”这个布尔函数。抑制零件（inhibitory organ）则实

现“p 与 非 q”布尔函数。激发信号产生器（stimuli producer）则起到刺激信号源的功能。第五种零件是刚体零件（rigid member），这是一种刚性的组件，被用来当作自动机实体的结构骨架。刚体零件不接收任何刺激，它对于信号是绝缘的。刚体零件可以同其它的刚体零件组合起来，也可以用来架设其他的零件。另有一种连接零件（fusing organ），当它受到刺激的时候会把两个零件“焊接”在一起。编者认为连接零件的工作方式如下，假设一个结构上的 A 点将要和另一个结构上的 B 点焊接在一起，这时候连接零件上面活动的输出端子就会同时与 A 和 B 接触，在 t 时刻向连接零件发送一个刺激信号，于是导致 t+1 时刻 A 和 B 两点就会被焊接在一起。然后连接零件就可以离开现场。此外还有切割零件（cutting organ），受到刺激的时候，切割零件就会把连接切开。

第八种零件称为肌肉零件（muscle），可以用来产生运动。肌肉零件一般是刚性的，可以连接到其它的零件之上。如果在 t 时刻肌肉零件受到刺激，在 t+1 时刻它就会收缩到零长度，并保持之上的所有连接。只要刺激信号不消失，肌肉零件就一直保持收缩。编者认为肌肉零件这样工作：比如肌肉零件 1 把一个结构上的 A 点和另一个结构上的 B 点连在一块，肌肉零件 2 又把 A 点和一个连接零件的输出端子 C 连在一起。然后一旦肌肉 1 和 2 都受激发，它们就会收缩，从而把 A、B、C 三点凑在一起，这时候激发连接零件，A 和 B 点就会被焊接起来。最后，当肌肉上的刺激消失之后，它们就会恢复到原有的长度，肌肉 1 上至少有一个端子也会同 AB 点分开。至于一开始肌肉和其他零件的连接是怎样实现的，后来又是怎样断开的问题，冯纽曼似乎没有提到。



漂浮着机器零件的池塘（译者加）

摘自 Gene Pool 人工生命软件

按照冯纽曼的设想，自动机按照以下的方式制造其它的自动机：在一个平面上漂着母自

动机，周围是无穷无尽的零件海。母自动机在存储器中包含有将要制造的子自动机的描述，并按照描述，捡起所需的零件装到正确的位置上去。要做到这点，母自动机必得包含一个能够抓住并且辨认零件的装置。在 1948 年 6 月的课上，冯纽曼对这个问题只是稍微提了几句话：两个激发零件从母自动机上像触角一样伸出来，当它们碰到其它零件的时候，就可以通过激发一个信号去测试所遇零件的性质，如激发零件能够传送信号，而结构零件就不能；肌肉零件受到刺激则会收缩，等等。

冯纽曼在他首次的设计尝试中，忽略了能源和能量的问题。他打算之后再考虑这个问题，比如把电池作为一种新的基本零件引入进来。除此以外，冯纽曼的这个早期自复制模型处理了以下的几何动力学问题：包括运动、接触、定位、连接、切断；但是没有考虑真正机械和化学意义上的能量和力的问题。因此我把这个模型叫做自复制的动力学模型（kinetic model）；本书的第二部分介绍的是冯纽曼之后提出的自复制元胞模型（cellular model），读者可以对比此两者。

在 1948 年 6 月的课上，冯纽曼提出了动力学自复制模型是否需要三维空间才能实现的问题。他怀疑只有在三维空间或者黎曼曲面（多连接的复数平面）上才能实现该模型。但在本书的第二部分中，二维平面已经足以实现自复制的元胞模型了。这似乎说明，二维平面也足以实现动力学自复制了。

让我们继续回到伊利诺斯的讲座中：冯纽曼讨论了自复制自动机的一般设计方法。他说，在理论上只要有足够的时间和原材料，就可以建立一个能够复制任何机器的车间。这个车间中有一台具备如下能力的机器 B：对于任何一个结构或者机器 X，则机器 B 会自动扫描 X，并把 X 上面的所有零件，以及这些零件的相互连接方式都列成表格，从而得到 X 的完全描述。再根据以上的描述，机器 B 就可以把 X 同样地复制出来。“这同自复制已经非常接近了，因为我们可以把 B 喂给它自己从而得到自己的一个复制品”】

但是，相对说来比较简单的，而且同样可以实现最终目的的做法是，不去直接建立能够复制任何给定结构或者样本的自动机，而是去做一个能够根据逻辑描述来组装目标的自动机。因为，按照任何可设想的方式，复制一个对象都必定分成两步，先是从具体实物抽象出描述，然后再从这个描述到具体实物。因此，先做后面这步会比较简单一些。

要做到这件事，我们必先得有一个自动机的公理化描述。如我们所见，我的做法和图灵的通用自动机很像，都是从一个理想机器的通用描述开始。我之前比较含糊地提到过，我们一共有大约十几种基本零件，如果把它们的细节都具体写出来的话（估计两页纸就够了），我们就可以得到一套刻画自动机的无歧义的形式语言。现在，我们可以把这些形式记号转变为二进制，并记录在打孔纸带上面。因此，任何自动机的描述都可以记录在打孔纸带上。我们可以不描述对自动机每一个零件以及这些零件之间的连接方式；而是直接描述这个自动机是如何被一步一步组装出来的，后者会比较方便一些。

【冯纽曼接下来说明了怎样把刚体零件转换成二进制的打孔纸带的过程。见下图，其中每一个基本链的交汇处都可以用一个二进制字符编码。1 表示对应位置有零件连接，0 表示没有。如果对这个数字串进行读写，那么对应位置上的零件也相应地被修改。】

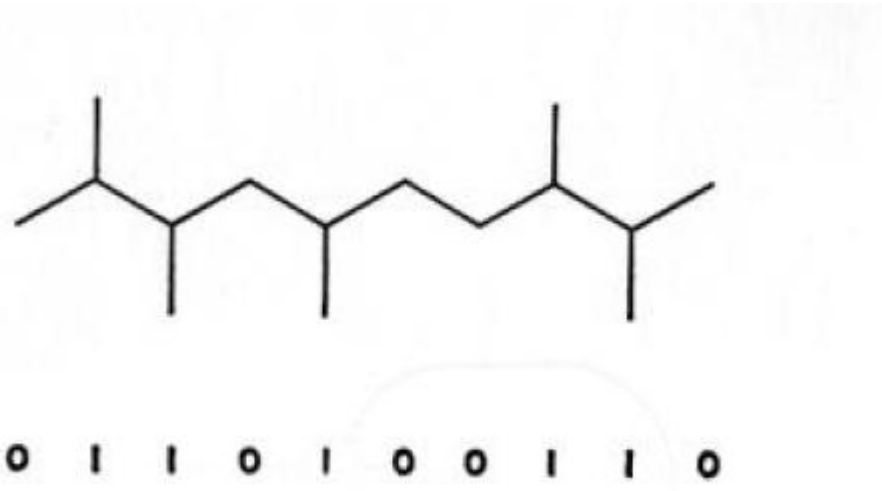
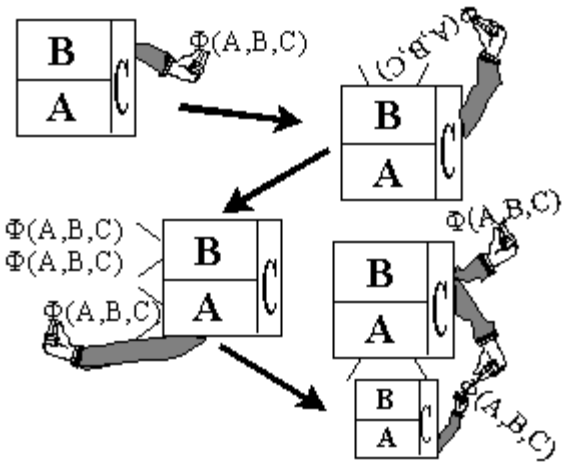


Fig. 2. A binary tape constructed from rigid elements

由于我有一种纯数学上的习惯，喜欢把一切东西用最简单的记号表示出来，所以这里的表示可能有些过于简化了。因为我现在用的是二进制表示，所以在上图中，我们不考虑支链的问题，或者说每个位置只能连接一个零件。现有的描述语言和符号系统所用的符号要比二进制更多，但是这里只要用二进制符号就足够了，我们可以毫无困难地表示出更多的符号，只需把支链也分别表示出来，并接上去就行了。其实，我们并不一定要用线性的符号系统。我们也可以用复杂得多的，包括循环的环状结构来表示自动机，但那个时候非线性的代码就不灵了。有理由怀疑我们对于看似简单的串行编码的偏好，只不过是一种来自语言的习惯；很可能存在着非串行的，效率却高很多的自动机描述方式，由于我们对于这类组合却缺乏直觉。

四、自复制自动机的核心



冯纽曼自复制自动机模型的形象展示（译者加）

图片来源：http://informatics.indiana.edu/rocha/ss504_5.html

要给出一个完全的公理化描述体系并不太困难，由此我们可以把任何可设想的自动机用二进制代码表示出来。任何这样的描述都可以用类似上图的一串零件来代表。假设自动机 X

的符号串是Φ(X)。接下来，我们可以设计一台通用构造器（Universal constructer）A，当我们把Φ(X)喂给 A 的时候，它就能够逐步地利用悬浮在周围的零件，把 X 一点点的组合出来。实际设计工作当然是很麻烦的，但是理论上却是可以办到的。因为这个过程可以归纳为形式逻辑的分步推理，从性质上说，这和通用图灵机并无区别。

还有一件事要说，我之前提到过，要制造一台能够直接复制任何自动机的机器是比较复杂的。所以最好是从要复制的机器的描述而不是从机器的实体来进行复制。但是我想补充一点，存在着某复制机器可以直接拷贝线性的刚体零件链，这是很简单的。因为导致实体复制困难的真正原因是，实体自动机的结构同我们串行的思考习惯完全不同，各种各样的零件朝着各个方向相互连接。仅仅是把已经扫描过的零件排除在列表外，便是很麻烦的事情。但是拷贝一根长链并无这种困难。所以，我可以假设存在一台通用拷贝机器 B(copy automaton)，当我们把任何描述输入 B 的时候，B 就会制造出同样的两份描述出来。

在完成以上两步之后，可能给人一种错觉，在此过程中复杂度衰退的原理似乎仍然没被打破。在表面上看，似乎在复制过程中，既没有产生更微妙的东西，也没有建立更多的联系。A 只能按照描述来制造 X。而按照对于复杂度的一般认识，X 的复杂度是和 X 的描述相等的。另一方面，B 拷贝得到了两份Φ(X)，但是两个同样的事物放在一起，没有理由说它们作为整体的复杂度要高于其中一个的复杂度，并且，我们还需要额外的机器 B 来完成复制。但实际上并不是这样的。

现在我们可以做下面这件事，我们可以把机器 A 和 B 组合在一起，并给 A+B 添加一个控制器 C。C 按照下列方式对 A 和 B 施加控制：C 先命令 B 拷贝两份描述Φ(X)；然后再命令 A 按照Φ(X)来实际制造 X，并把其中的 1 份Φ(X)拷贝去掉；最后，C 会把 X 和剩下的那份Φ(X)捆在一起，并把它们从机器 A+B+C 的组合中间分离出去，这样一来，我们就制造出了 X+Φ(X)这样的组合。

按照以上原理，如果我们用（A+B+C）来代替 X，并进行上述同样的操作的话，那么(A+B+C)+Φ(A+B+C)的自动机组合，就可以制造出自动机组合：(A+B+C)+Φ(A+B+C)出来。因此，自动机自复制得到了实现！

【这个过程的细节如下：

1、 现有自动机(A+B+C)，并附有它的描述Φ(A+B+C)。
2、 从(A+B+C)+Φ(A+B+C)开始复制流程。
3、 C 控制 B 拷贝两份描述，得到：(A+B+C)+Φ(A+B+C)+Φ(A+B+C)。
4、 C 命令 A 按照Φ(A+B+C))来实际制造出 A+B+C，
得到(A+B+C)+(A+B+C)+Φ(A+B+C)+Φ(A+B+C)。
5、 最后 C 把新得到的自动机 A+B+C 和它的描述Φ(A+B+C)捆在一块，并把自己和新自动机分开，这就得到 2 个(A+B+C)+Φ(A+B+C)；复制完成。】

这个过程并不是循环论证：我首先把 A 和 B 做了清楚的定义。在我提到 X 之前，我已经说明，C 可以适用于任何形式的自动机 X。接下来定义了一个变量 X，它描述了 C 将要怎么做，然后，我再让这个变量 X 和 C 产生联系。所以，A、B 和 C 的定义是完全独立于 X 的，在此之后，我再让这个 X 指代 A、B 或者 C。因此，整个过程并非循环。

以上的通用构造器（general constructive automaton）A 具有一定意义上的创造力，也就是说，A 可以从抽象的描述来“制造”出实体的机器来。同样的，通用拷贝机 B 也有一种能够把一份描述变成两份的“创造能力”。但 A 和 B 都不具备自复制能力，此外，控制器 C 也远远没有具备任何形式的创造或复制能力，它唯一能做的就是刺激其它的两个组件去做一些事情，把一些东西连接在一起，或者把一些东西从原来的系统中分割出去。然而，一旦 A、

B 和 C 组合在一起，它们作为一个整体却能够复制自身。故而我们可以把一个自复制系统分割成不同的部分，每一个部分虽然都不能够复制自身，但对于自复制机器整体却又都是必不少的。

五、自复制自动机的进化

我们还可以做另外一件事，让 X 代表 A+B+C+D，这里 D 代表任何自动机。那么(A+B+C)+Φ(A+B+C+D)就可以制造出(A+B+C+D)+Φ(A+B+C+D)。换句话说，我们的自复制机器不仅仅有复制自己的能力，还可以顺便生产出其他的组件 D 的能力。这就是任何自复制生命都具备的功能：在复制自身的时候，它还会创造出副产品。

作为一个系统，(A+B+C+D)可以发生类似变异的过程。在定义“自复制”究竟是什么意思的时候，我们会遇到这样的困难：有些结构，比如晶体的生长的确也是在复制自己。但是我们都觉得把晶体称为自复制，显然是名不副实的。有一个办法可以绕过这个困难，就是把发生变异的能力，以及制造类似却不等同于母体的生命的能力包括在“自复制”的定义中间。

现在考虑(A+B+C+D)+Φ(A+B+C+D)这个自动机。“变异”是指中间有一个零件发生随机的变化。如果是 A、B 或者 C 的一个零件发生了变化，那么系统通常就会失去自复制的能力。比如 C 的一个零件被修改以后，C 很可能就不能在正确的时间上给 A 和 B 发射刺激信号，或者无法在需要的时候进行连接和分割，这样的变异就是致命的。

但是如果变异发生在描述Φ(A+B+C+D)上面，那么系统制造出的就不再是它自己，而是修改后的自己，下一代自动机能否继续复制取决于变异发生的具体位置。如果 A、B 或者 C 发生了变化，那么子代自动机就会“绝后”。但是如果变异发生在 D 的描述上，那么除了 D 变成了 D’之外，变异的子代同母体系统完全相同。之后的子代会把这个变异 D’继承下去。这就是可遗传变异的基本过程。

总之，虽然这套系统还非常原始，但它已经具备了可遗传变异的基本特性。大多数随机变异都是致命的，但是也可能偶尔会发生非致命乃至是可遗传的变异。这是遗传所特有的性质，这套系统也同样具备了。

Jake 点评

一、生命本质的另类研究

“生命是什么？”这是一个非常古老的哲学问题，然而，正是对这一问题的探索才催生了今天蓬勃发展的生命科学。细胞生物学、分子生物学、生物化学、生物物理学，……所有这些学科都是围绕着这个问题展开的。主流的大多数生物学研究都基于这样一种假设：生命这种形态一定是与物质构成有关的，也就是说，只有以碳、氢、氧元素构成的物质才有可能具备生命。在外星生命探测中，科学家们拼命想找到水这种碳基生命存活的基本要求就是这种思想的产物。

然而，另外一种不怎么主流的观点认为，实际上生命的本质并不在于物质构成，而在于基本原件的组合方式。所以，按照这种组合方式，我们可以用碳、氢、氧元素去构成，也可以用其它的元素来构成，甚至可以是抽象的数字。这就是人工生命的基本思想，即“生命如其所能（life as it could be）”，而非“生命如吾所识(life as I know)”。因此，生命的本质实际上属于一种“软件逻辑”。

如果我们能够透彻地理解了真实的“吾所识”的生命，我们当然可以较有把握地掌握生命的本质，但是，这完全不能阻挡我们直接从“其所能”的纯粹抽象逻辑出发来研究“规范”的生命理论。而这恰恰是冯纽曼研究的出发点，他就是在寻找生命的最小逻辑内核。

由于冯纽曼的这种对生命本质的探索实属另类，从而造成了我们对他的研究的理解偏差。很多人非难冯纽曼的研究闭门造车，他不去考察真实生命的运作机制，而是自己创造一套非常不符合现实的自复制自动机。还有的人指出，现在对生物学的研究已经对生命自复制的过程有所了解了，因此冯纽曼做的这些粗糙的假设都可以被扔进垃圾桶了。

首先，我们要知道，冯纽曼研究自复制自动机理论的时候，人们还没有发现 DNA 双螺旋结构（沃森和克里克在 1953 年发现了双螺旋结构，冯纽曼死于 1957 年，而关于自复制自动机的工作多在 1940 年代做出的）。其次，冯纽曼追求的目标与生物学家并不一样。就像是康托尔、图灵、哥德尔等数学家对逻辑本质的追求催生了计算机的发明一样，冯纽曼正在沿着一条规范研究的道路趋近生命的本质。有可能这条道路最终将能创造出与真实生命完全不同的生命形式出来。

另外一种对冯纽曼的误解在于冯纽曼所追求的生命本质的软件就是这种可以自复制的程序。而如今这种会自复制的程序已经比比皆是了，它很难和真实生命的复杂性进行类比。但实际上，如果你仔细读冯纽曼这五篇讲稿就会发现，其实冯纽曼追求的真正问题并不在于自复制本身，而在于“复杂度的阈值”以及概率论中的“漏洞”，他怀疑自复制的自动机恰恰就处在复杂度阈值的边缘，而利用了概率论“漏洞”实现自我复制的。也就是说，其实，这本书里讲的自复制自动机仅仅是冯纽曼为了探索诸如复杂度阈值等概念的一个起点而已。

所以，冯纽曼的自复制自动机有点类似于 19 世纪人们讨论的理想模型——卡诺热机。卡诺热机并不能直接用来当发动机使，但是围绕着这个理想模型，人们却能找到“熵”这个物理量的精确科学表述。同样的道理，冯纽曼的自复制自动机的精髓也并不是在探讨自复制问题，而是想围绕着这样一个理想模型而探索复杂度的概念。

从这点出发，就不难看出实际上继冯纽曼之后的若干自复制自动机的研究，例如人工生命之父朗顿的环，还有很多自复制的元胞自动机模型其实已经偏离了冯纽曼一开始的初衷。他们更加注重如何实现自复制的结构，而忽视了与自复制相关的诸如复杂度阈值等概念。

二、概率论中的漏洞

个人非常喜欢冯纽曼的这个富有“诗意”的说法，生命就好像是宇宙中物理、数学法则的黑客，专门寻找后门，从而利用它完成自己的复制。而如冯纽曼所言，为什么偏偏是“概率论”中的黑洞，而不是诸如“群论”的漏洞，“解析几何”的漏洞？这说明，冯纽曼已经深刻地认识到生命这种现象是一种“统计的”规律，或者用上一章的语言来说，生命是用一大堆不可靠的原件搭建起来的一台可靠的机器，这台可靠的机器一定会操纵概率法则来“统计地”实现自身的存在。

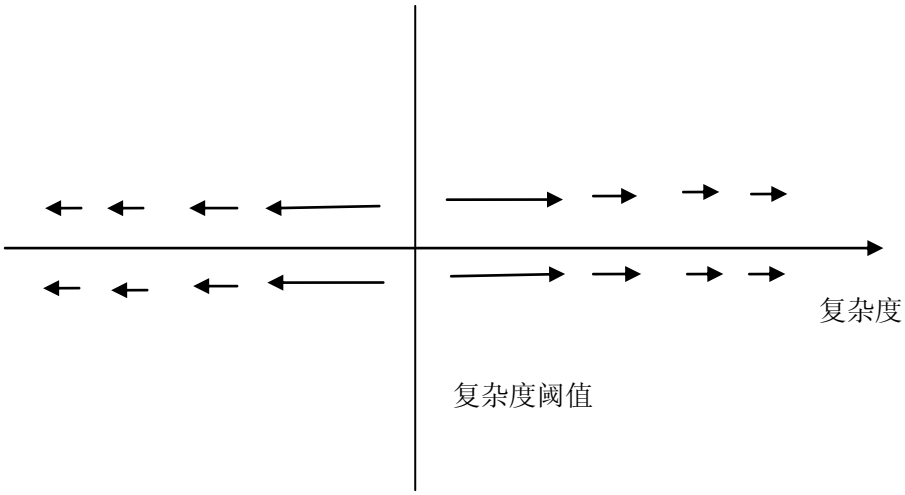
那么，要成功地利用这个概率论中的漏洞，需要具备什么条件呢？在此，冯纽曼就指出了“复杂度阈值”这个条件。也就是说，只有那些能够达到一定的复杂性并且突破了某个阈值的系统才有可能成功的利用“概率论中的漏洞”。接下来的问题就是，这个复杂度如何度量？复杂度的阈值到底是多少？这两个问题就问到点子上了，因为冯纽曼自己也不知道如何来定义这个复杂度的概念。更不用说如何计算出这个阈值的大小了。

不过，冯纽曼还是给出了这样一种定性的描述，“这里的复杂度就是指复杂的有效程度，或者说是做事情的潜力。这里我说的并不是一个具体对象牵涉到的复杂程度，而是它有目的地去做事的时候牵涉到的复杂程度。从这个意义上说，具有最高复杂度的对象就是那些可以做很困难的，牵涉很多事情的东西。”

其实，他这里所说的“做事情的潜力”的说法是来源于对一般的自动机的观察：“大家都知道通常一台机器总是比它能够制造出的零件更复杂。因此，一般地说，如果自动机 A 能制造出自动机 B 的话，那么 A 一定包含关于 B 的全部信息，这样 A 才能按照这些信息把 B 制造出来。如此这般，我们就发现，自动机的‘复杂度’，或者说它的生产潜力，是不断降级的。也就是说，一个系统的复杂度总是比它制造的子系统要高一个数量级。”

对于这样一种衰退现象的描述，我们都有亲身体会：屋子长时间不扫就会变脏；车子长时间不保养就会出问题；甚至你的计算机长时间不重装系统就会经常死机，……。所有这些经验都牵扯到一个非常著名的科学定律：热力学第二定律。热力学第二定律说，任何一个封闭的系统最终必然会导致熵增。这里的熵就是指混乱度，所以，任何一个封闭的系统都会朝越来越混乱、无序的方向发展。因此，在这一部分，冯纽曼所说的复杂度概念应该至少可以覆盖熵的概念。它体现出了一种降级、衰退的性质。

但是，现实世界中的确还存在着欣欣向荣的一面，生物的进化、科技的进步、经济的腾飞，所有这些方面就体现为复杂度的升级和前进。那么对于这一部分恐怕我们就不能简单地用熵来刻画了，因此，复杂度指标还必须能够描述复杂性的升级。总体来说，我们可以总结出这样一张图：



如图所示，这个复杂度阈值就好比是一个排斥子，只要系统的复杂度没有达到这个阈值，系统就会在热力学第二定律的作用下自发地实现复杂度的衰退；而系统的复杂度一旦超过这个阈值，那么也许同样是因为与热力学第二定律类似的作用（冯纽曼在讲后面关于进化的部分，已经暗示了，如果热力学第二定律所主导的扩散是作用在数据描述Φ(D)上面的话，那么进化就能产生），那么系统就会呈现复杂度的升级。

所以，我们可以断言，复杂度阈值是一种信息论的概念（因此它跟熵、Kolmogorov 复杂性有着异常紧密的联系）。另外，所谓的衰退，自发衰减等效应都跟热力学第二定律有关，所以这也让我们看到了这部分概念实际上与上一章提到的热力学、信息论是紧密结合的。最后，也是最有意思的一点，尽管冯纽曼没有明确指出来的：也许进化与衰退恰恰是一枚硬币的两面。对于突破了复杂度阈值的系统，它就会由于在数据层的扩散和变异作用而不断进化，演化出复杂的结构；而对于低于复杂度阈值的系统，热力学第二定律就会无情地让它衰退、耗散。

三、蒯恩程序

于是，一切的矛头都指向了这个神奇的复杂度阈值，它到底是多少？虽然冯纽曼到死也没有给出任何关于复杂度的计算，但是他用一名数学家和哲学家敏锐的头脑，深刻地指出了其实这个复杂度阈值与数学、逻辑学以及哲学上的一个重要概念“自指”有关。

冯纽曼在本章最末部分所讨论的能够自复制的自动机的抽象描述：(A+B+C)+Φ(A+B+C)其实就是一个哲学家们称之为“蒯恩”的程序。蒯恩（Willard.V. Quine）是一位美国的哲学家，他创造了一种称之为蒯恩的方法，使得人们可以不使用“我”或者“这句话”等词语就能创造出可以谈论自身的句子来。例如下面这句话就是一句不使用“这句话”的自指悖论：

把“把中的第一个字放到左引号前面，其余的字放到右引号后面，并保持引号及其中的字不变得到的句子是假的”中的第一个字放到左引号前面，其余的字放到右引号后面，并保持引号及其中的字不变得到的句子是假的

当你按照这句话的指示将引号中的文字放到引号后面的时候，你就得到了一句自我否定的句子，所以上面那句话与下面的话等价：

这句话是错的

之后，数学家 Kleene 将蒯恩这种语言上的操作进行数学化就得到了一种更加普适的 Kleene 递归定理。有了这个递归定理以后，数学家就可以在严格的数学公理体系中玩各种各样的自指游戏。之后，哥德尔利用这种技术构造了公理系统中著名的哥德尔句子“本命题不可证明”，从而推出了哥德尔定理这个被纽约时代杂志评选为 20 世纪最伟大的数学定理。

那么，究竟这个蒯恩程序是什么呢？其实，它非常容易理解，它就是一个能够把自己的源代码打印出来的程序。

```
S (x){
   q='S(x){\\n   q=\\\'\'+q+\'\\\';\\n   Print(\\\'\'+p(q)+\'\\\');\\n}';
   Print ('S(x){\n   q=\''+q+'\';\n   Print(\''+p(q)+'\');\n}');
}
```

这里面的“\n”表示换行符，即如果执行 Print('A\nB')，则程序会输出下面的字符串：

A

B

“+”表示将两个字符串进行串联形成一个新的字符串，例如 A='123',B='456'，则 A+B='123456'。

这个自打印程序调用了一个简单的解码函数 p(q)，p 的作用是将字符串 q 变换成更浅一层次的字符串。例如，如果 q 是“\\\'\'\\n\\”，那么 p 这个函数就会计算输出“\''\n\”。也就是说 p 完成了一组映射：它把“\\”映射成“\”，把“\'”映射成“'”，而把“\n”映射成回车符。显然 p 是可以写出来的。大家可以利用 java 或者 VB 来实现这个程序，运行它就会发现它能够自我复制。

让我们来分析一下这个程序是如何运作的。首先，看程序的最后一行，即 Print ('S(x){\n q=\''+q+'\';\n Print(\''+p(q)+'\');\n}'); 这句话的作用是让程序在屏幕上打印出一个字符串。注意观察，这个被打印出的字符串其实是由“+”号被分割成了 5 个部分，第一部分是“S(x){\n q=\'”，第二个部分是 q 这个字符串的原封不动的拷贝，第三部分是字符串：“\';\n Print(\'”，第四部分是函数 p 作用到 q 上面的结果即 p(q)；第五部分还是一个字符串：“\');\n}”。然后当我们把 q 字符串的数值代入第二部分和第四部分，并进行运算 p 之后，就得到了和源程序一模一样的结果。你不妨在计算机上运行这段程序，就会发现这段程序会在屏幕上赤裸裸地把自己的源代码打印出来。

我们不妨把这段程序的 5 个部分进行归并，写成由下面的三部分构成的：Copy∘ Popup ∘ Control，其中 Copy 就是 5 部分中的第二部分，即相当于一个拷贝字符串的程序，你输入给 Copy 什么字符串，Copy 就会把那个字符串再原封不动地吐出来；Popup 这部分就是原来的 5 部分中的第四部分，即函数 p，它的作用相当于一个弹出操作，也就是为输入的字符串脱去一层引号。如果输入的字符串原来是在第 n 层虚拟世界，则 Popup 的作用就是让字符串跳到第 n-1 层；最后 Control 这部分就相当于原来的第 1、3、5 这三部分以及最一开始的语句 Print 的总合，它的作用就相当于是为 Copy 和 Popup 制造出来的字符添加适当的连接词，使得最后的字符串能够拼接成与原来的程序一模一样的源程序，并将其打印到屏幕上。所以这句“Print ('S(x){\n q=\''+q+'\';\n Print(\''+p(q)+'\');\n}');”就可以改写成(Copy∘ Popup∘ Control)(q)。其中“∘”表示将不同的程序连接为一体。

如果我们把一个计算机程序 X 的描述（或者称源代码）写为Φ(X)，则自打印程序的第一条赋值语句就相当于给 q 赋予了Φ((Copy∘ Popup∘ Control))，即(Copy∘ Popup∘ Control)这三个程序连在一起的源代码。最后我们可以将自打印程序简写为：

```
S(x){
      q=Φ(Copy∘ Popup∘ Control)
      (Copy∘ Popup∘ Control) (q);
}
```

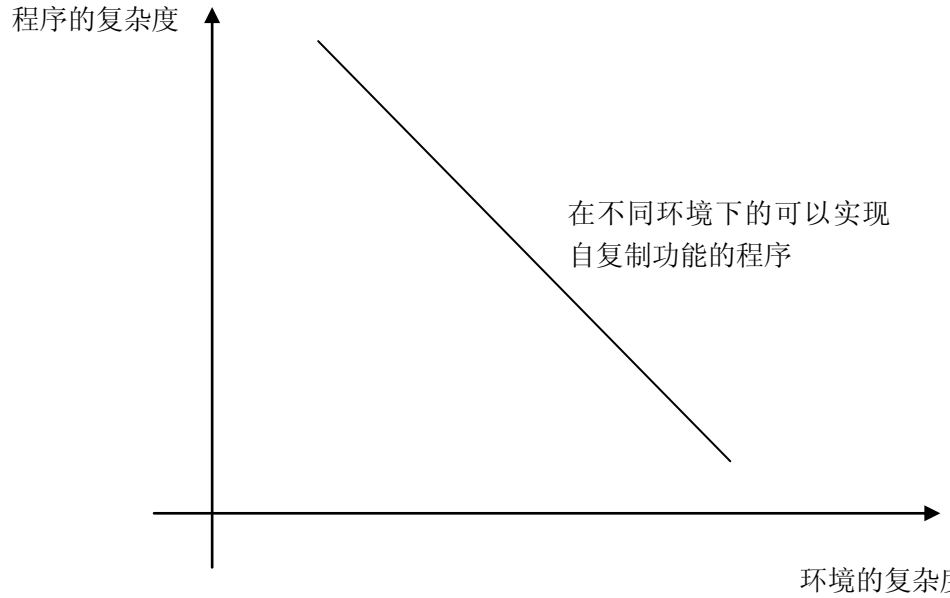
那么，观察这个程序，就会发现实际上它就是冯纽曼所说的那个自复制程序：(A+B+C)+Φ(A+B+C)了。在这里 Copy 就相当于冯纽曼程序中的拷贝器 B，它能将输给它的数据原封

不断地再打印出来；Popup 就相当于冯纽曼说的通用构造器，它能够根据一段数据而把数据对应的自动机的源代码打印出来（这相当于从描述中构造出自动机）。Control 这部份就对应了 C。而 q=Φ (Copyo Popupo Control)就对应了描述：Φ (A+B+C)。因此，我们实际上可以很容易地用我们的个人电脑实现自复制的计算机程序。

关于蒯恩、递归定理以及自指更多的讨论请参看本人写的科普文章：《系统中的观察者（5）——自指——一条连接图形与衬底的金带》，以及更多的参考书，包括：侯世达：《哥德尔、艾舍尔、巴赫》，商务印书馆，1996；R.M. Smullyan: Diagonalization and Self-Reference, Oxford University Press, 1994。

你也许会提出这样一个质疑，即使是这个蒯恩程序也没有实现真正的自复制。因为这个程序仍然需要调用一些字符串处理程序例如函数 p，以及字符串运算符“+”等等，而这些功能并不是这个程序自己完成的，而必须借助比它更高一层的编译器才能完成。于是，这个蒯恩程序实际上已经与一个平庸的自复制程序（例如就包含了一个平凡的高层次语句”Copy me”）没有任何区别了，只不过对于平庸的自复制程序来说，我们为自复制的环境赋予了过高的能力，而自复制程序本身只起到一个触发器的作用。而对于蒯恩程序来说，翻译蒯恩程序的环境变得相对简单一些，但是实现自复制的代码就要承担更大的责任。原则上讲，我们的确不能做出来一个完全不借助于环境，而凭空实现自复制的机器。但是，我们总可以降低对环境的要求，把编译器的复杂度变小，从而凸现出自复制程序的能力来。

由此我们可以看出，在不同的情况下，编译环境和在此环境下实现的程序会呈现出连续的复杂度变化。而自复制程序实际上是对环境的复杂性和程序本身的复杂性的一种折衷。因为如果环境的复杂性越高（编译环境可以支持很复杂的指令集合），要实现一个可以打印自身源代码或者复制自身的程序就变得很简单（程序自身的复杂度下降）；反过来，如果环境的复杂度越低，那么你要实现一个自复制的程序也会变得越困难。于是，我们可以得到下面这张图：



如图所示，如果我们把一切程序按照它所运行的编译环境以及它自身的复杂程度列出这样一个连续的空间，那么在不同的编译环境下的自复制程序就会近似地形成一条如图的直线。我们可以肯定地知道，它是一条单调下降的曲线，因为随着环境复杂度的升高，要实现

自复制的程序的编写也会变得越来越轻松。所以，我们大概可以估计出来这样一个自复制程序所满足的复杂度方程：

$$C_E + C_P \geq h$$

其中 C_E 表示环境的复杂度，C_p 表示程序的复杂度，h 是一个常数。上述公式表示了实现自复制程序的复杂度的允许条件，当等号成立时，它就对应了图中的向下的直线。不等式对应了直线右上角的半平面。这个式子让我们联想到了第二章提到的不确定性原理。因此，自复制程序的复杂度讨论也许的确存在着类似量子理论中的不确定性原理。这种从不同层次来考虑程序复杂度的概念也许真的蕴藏着我们尚不知道的真理。

总之，我认为自复制自动机才真正抓住了人们常说的“涌现”这个关键概念的本质。因为从单个组成单元来看，无论是通用构造器 A，还是拷贝器 B 还是控制器 C，以及它们的数据Φ(A+B+C)都不具备自复制的功能，而当把它们恰当地合在一起的时候，它们的确可以实现自复制，从而实现了自复制功能的涌现。与其它对涌现的机制讨论的不同之处在于，自复制自动机具有相应的数学对应物：递归定理，因此我认为如果要从数学上理解涌现，自复制自动机是绕不开的。只可惜，冯纽曼的最终梦想：从热力学和信息论的角度来理解自复制自动机的自发涌现仍然没有实现。