



## 第 1 章

# 思想萌芽：规则判断的年代

在遥远的 1911 年，或许是深秋的某一天，久违的阳光透过帕丁顿火车站巨大的玻璃拱形屋顶，洒在站台上熙熙攘攘的人群中，空气中弥漫着蒸汽机车的煤烟味。推开站厅大门走出车站，你会看到街道两旁是大片乳白色或米黄色砖石砌成的建筑，它们大多都有着高耸的窗户、铸铁栏杆，以及宽阔的台阶，这些建筑是帝国鼎盛时期自信与秩序的象征。而街道上则是另外一幅流动的画卷：四轮马车和马拉货车川流不息，马蹄敲打在鹅卵石路面上，发出清脆的声响；偶尔也能看到少数富人们驾驶私家汽车穿梭其中，他们会不断地按喇叭，提醒行人避让。那时的英国，正处于一个新旧交替的迷人时代。

在一条安静的中产街区内，一对夫妇刚刚从英属印度回到他们的家中。次年 6 月，一个新生命诞生于此。伴随着婴儿稚嫩的啼哭，窗外那些戴着圆顶礼帽的绅士们依旧行色匆匆，并未意识到历史的齿轮已悄然开始转动。这个婴儿，将在未来赋予无生命的机器以“灵魂”，点亮**人工智能 (Artificial Intelligence)** 的黎明。他的名字，叫艾伦·图灵，其肖像如图 1-1 所示。

在本章，我们将从图灵的设计开始，来回顾早期人工智能概念的形成与发展，时间跨度大约是从 20 世纪 40 年代末至 80 年代



图 1-1 图灵肖像

末。你将看到一群极具想象力和创造力的学者，以及诸多他们试图改变世界的作品——即便现在看来，他们的作品是那样的简陋。你还将看到人工智能这个年轻的领域在最初约 40 年间的两次起伏，以及背后的故事。

## 1.1 图灵之问：机器能思考吗

图灵渐渐长大，开启了他传奇的一生。他曾在二战中主导破译了德军的密码，相当于间接拯救了无数人的生命；他发明了一种叫作“图灵机”的架构，从理论上定义了什么叫作“可计算”，这是现代计算机的理论原型；他因为性取向问题在那个人性尚显蒙昧的年代被迫接受了激素治疗。

而此处我们要讨论的是图灵关于机器智能的思考，主要内容来源于他的两篇论文，分别是他在 1948 年撰写的《**智能机器**》( Intelligent Machinery ) 与 1950 年发表的《**计算机与智能**》( Computing Machinery and Intelligence )。图灵在这两篇论文中系统性地讨论了机器表现出智能行为的可能性、方法及其哲学基础。

受到当时神经生理学的启发，图灵认为婴儿在刚一出生时，其大脑皮层的神经元结构是随机连接的，就像一台无组织的机器，其初始行为是混乱、无组织的。出生后，人类通过一系列的学习逐步获得知识和技能，最终形成智慧。并且图灵还细心地观察到了“奖励”和“惩罚”对婴儿行为的影响。他认为，在这些过程中，原本无组织的神经元被逐渐连接和修改，从而获得了执行复杂任务的能力。

从今天的视角来看，图灵的理论或许显得有些粗糙。然而在那个年代，这些理论却很大程度上让当时的很多学者相信：机器是有可能表现出智能行为的。图灵指出，既然人类婴儿的大脑可以从一种近乎空白的、无序的状态，通过后天的学习和训练获得智慧，那么理论上，一台原本“无知”的机器也可以通过类似的方式“进化”出智能。

这一观点顺理成章地引出了实现“机器智能”的方法。第一步，设计一个类似人脑具备可塑性的基础结构；第二步，通过某种“教育”让这个主体“学习”到一些所谓的“知识”，让其可以表现出智能。

20 世纪 50 年代，虽然还不具备具体的工程化能力，但是从逻辑层面来看，图灵的思想无疑是清晰并且有创意的。而如果从今天的视角来看则更加令人感到

惊讶，因为此刻我们每个人都在用的诸如 ChatGPT 的大语言模型（Large Language Model, LLM）的基本技术逻辑与图灵当年的设想非常相似。从这个意义上讲，称图灵是某种先知或许也不为过。

同时，图灵还在他的文章中提出了一种检验机器是否具备智能的方法，他称之为“模仿游戏”，后人则将类似的方式称作**图灵测试**。想象一下，假设你是图灵，需要给“机器智能”下某种定义，从而判定一台机器是否具备真正的智能，你会如何做呢？按照普通人的思路，你可能会先试图给“智能”下一个明确的定义，比方说，符合 1、2、3、4 等几个条件的就叫智能，然后再基于此去定义机器的智能。

如果是这样的思路，可能从一开始我们就会发现一个问题，即“智能”难以被定义。那么图灵是如何做的呢？我们通过图灵测试进行演示，如图 1-2 所示。

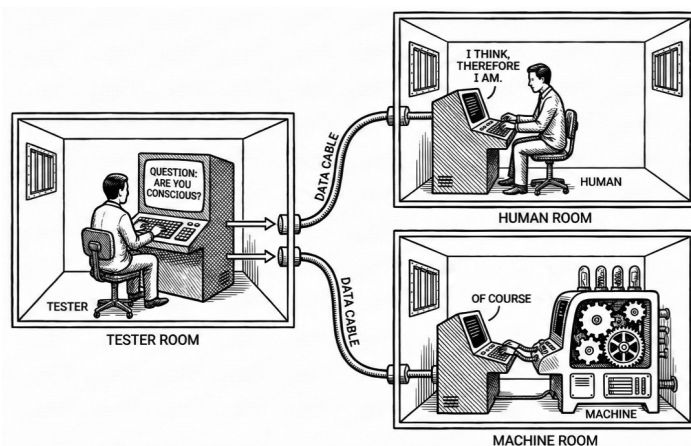


图 1-2 图灵测试示意（由 Nano Banana 生成素材）

设想有三个完全被隔离的房间，一位询问者被安排在其中一个房间中，另两个房间中分别是一个人与一台机器。询问者通过打字的方式提出问题，其他房间中的人和机器分别通过打字的方式回答，如此反复多次。如果询问者没办法有效区分回答他问题的两者哪个是人类、哪个是机器，我们就说这台机器通过了图灵测试。即，图灵绕过了对“智能”这一概念进行逻辑上的定义，转而提出了一种操作性的判定方式。

作为人工智能领域的理论奠基人，图灵的文章和思想就像打开了一个神奇的

盒子，盒子中的光倾泻而出，吸引了后续诸多的学者加入这个领域。同时，他提出了一个宏大但清晰的目标——造出能通过图灵测试的“机器”。

在图灵测试提出的5年后，即1955年，时任美国达特茅斯学院数学助理教授的约翰·麦卡锡想要发起一场学术会议，来讨论当时刚刚起步的关于机器智能的话题。举办学术会议一般需要赞助商赞助，所以麦卡锡决定联系一下洛克菲勒基金会，看看能否获得一些资助。

和现在很多创业公司融资要写商业计划书一样，麦卡锡需要一些炫酷的概念来包装他的会议，于是，他在举办这次学术会议的提案中首次使用了“人工智能”这个术语，并且给这次会议取了一个听起来高大上的名字——达特茅斯人工智能夏季研讨会。麦卡锡希望用一个夏天的时间来吸引一群学者到达特茅斯学院共同研究机器如何学习、使用语言、形成抽象概念、解决人类才能解决的问题，并能自我改进。显然，从今天的视角来看，对于研究这些内容来说一个夏天的时间太短了，但是在次年即1956年，**达特茅斯会议**还是略显随意地举办了。

-5a-

The Rockefeller Foundation is being asked to provide financial support for the project on the following basis:

1. Salaries of \$1200 for each faculty level participant who is not being supported by his own organization. It is expected, for example, that the participants from Bell Laboratories and IBM Corporation will be supported by these organizations while those from Dartmouth and Harvard will require foundation support.
2. Salaries of \$700 for up to two graduate students.
3. Railway fare for participants coming from a distance.
4. Rent for people who are simultaneously renting elsewhere.
5. Secretarial expenses of \$650, \$500 for a secretary and \$150 for duplicating expenses.
6. Organization expenses of \$200. (Includes expense of reproducing preliminary work by participants and travel necessary for organization purposes.
7. Expenses for two or three people visiting for a short time.

Estimated Expenses

6 salaries of 1200	\$7200
2 " " 700	1400
8 traveling and rent expenses averaging 300	2400
Secretarial and organizational expense	850
Additional traveling expenses	600
Contingencies	550
	<b>\$13,500</b>

图 1-3 麦卡锡用于申请经费的提案中，金额计算的部分内容



说它略显随性，主要是因为会议进行的过程中，其组织工作实在是看起来不太高明。例如，麦卡锡在会后遗失了参会人员的签到表，导致今天已经无法考证究竟有多少人参加了这次会议——关键是确定参加了会议的几位学者所回忆的人数也都各自不同。据说，完整的议程表和会议记录也被相继遗失。

尽管如此，在达特茅斯会议现场，依然有一些在后世看来神奇的研究成果被展示或者讨论。最具代表性的有两个，分别是由艾伦·纽厄尔和赫伯特·西蒙共同构思和开发的“逻辑理论家”程序，以及由阿瑟·塞缪尔开发的跳棋程序。前者可以自动证明一些数学定理，后者则展示出了机器通过“学习”来改进自身的可能性。接下来将用两个小故事来类比说明这两个程序的基本原理，在本书的后续章节中，你还会经常读到这类小故事。

需要说明的是，由于是类比，所以其表达的运作机制和细节没办法做到特别精确，只是以此帮助读者更加形象地理解相关的原理和逻辑。

想象一下，假设你是一位一直生活在中国的5岁小朋友，在电视上看到了一档介绍美国食物的节目，其中主持人展示了一种名叫三明治的美食。从电视上看来，这种食物好像是用两片面包夹了一些肉馅、菠菜、西红柿之类的食材而制成。于是你决定去厨房里面寻找食材进行实践操作。

你很顺利地找到了一个番茄、一整棵生菜和两片面包，以及两片早餐吃剩下的火腿肉。你又恰巧在厨房的案板旁看到了一本名为《一个数学家的烹饪法则》的书。你翻开书，在“食材基本处理方法”这一章中读到了以下内容：

- 切割律：任何番茄都可以切割成片；
- 组合律：在制作茄盒、饺子等两种或者多种食材组合在一起的食物时，应该把可以作为包裹物的食材放在外面，把容易散开的食材包在里面；
- 替换律：大多数情况下，菠菜、生菜、小白菜这三种食材可以互相替换，总体上做出的菜肴味道差不多。

这些内容读起来很奇怪，或许是因为这是一本数学家写的烹饪书。但无论如何，你已经跃跃欲试，而你采用的基本方法是根据目标反推。

第一步：分解目标。

你的目标是制作三明治，你将这个目标分解为三个子目标，分别如下：

- 子目标 1：找到两片面包；
- 子目标 2：制作出肉与菜混合的“馅”；

□ 子目标 3：用面包把馅夹起来。

非常明确！接下来你需要做一些基本的推理和“设计”。

第二步：基于书中的知识和厨房中的食材，思考这些子目标该如何完成。

首先，面包和火腿肉是现成的，可以直接使用。你找到的番茄是整个的，基于“切割律”，你想到只要把番茄切成片即可夹到面包中间去。接下来，你发现没有菠菜，而只有生菜，但是基于“替换律”，你认为使用生菜替代菠菜是可行的。至此，理论上你已经具备了所有的制作条件。

第三步：连接所有步骤，开始制作。

你拿出番茄，用刀切下两片放在盘子里——至于为什么一个 5 岁的小朋友可以熟练地使用刀具，我们暂时不讨论。接下来你从生菜上扯下两片菜叶，将其洗净后同样放到盘子中备用。你拿出一片面包，把火腿肉、两片西红柿和两片生菜叶放在上面，然后拿出另外一片面包盖在最上面。大功告成！你成功地制作出了自己的第一个三明治。

上面提到的“逻辑理论家”程序的基本原理概括起来跟 5 岁小朋友制作三明治的过程差不多，一共有以下三个要素：

- 目标反推：从要证明的结论出发，反推需要什么条件，然后去已知的“定理库”中寻找命中的条件，而定理库就相当于那本叫作《一个数学家的烹饪法则》的奇怪的书；
- 拆解：如果遇到了某个条件在“定理库”中找不到，尝试把它拆开继续找，就像生菜叶可以通过“扯”的方式从一整棵生菜上面获得；
- 回溯：拆解的方式可能不止一种，如果其中一条路径走不通，推倒重来，尝试另外一条路径。

不同的是，在该故事中，你作为一个 5 岁的小朋友，用人类的智能完成了上述推理和行为；而在达特茅斯会议上，“逻辑理论家”程序是基于一台古老的计算机，通过编程的方式实现了类似的推理和行为。

接下来，我们来讨论前文提到的阿瑟·塞缪尔开发的跳棋程序的基本原理，依然用一个故事来说明。

想象一下，你依然是刚才那个 5 岁的小朋友，吃了自己制作的第一个三明治后特别有成就感，但你并不满足于此，希望继续改良你的三明治，让它变得更好吃。但是说实话，对于一个三明治来说，究竟什么味道算“好吃”，你并不确定。



你又一次翻开了上面那本《一个数学家的烹饪法则》，数学家的思路果然很奇特，在第三章他提到了影响食物味觉的几个因素。数学家嘛，总是喜欢提炼出基本要素，并且把它们跟数字联系在一起。对于三明治这种食物来说，有如下要素：

- 咸度：即人的舌头对食物中盐的感知；
- 酸度：类似醋、柠檬等带来的一种刺激感；
- 脆度：类似生菜、薯片等带来的口感。

但是，书中只提到了这几个维度，并没有将它们有效量化，也没有告知具体怎样的配比才能让三明治更好吃。怎么办呢？你决定自己试一下。

首先，你根据自己的感觉给上面三个要素的重要程度排了个顺序。你平时喜欢吃酸甜口味的水果，所以认为酸度是最重要的，至于咸度和脆度，感觉没什么太大价值。你甚至做了一个量化比较，认为如果酸度的重要性是 10，那么其他两个的重要性各自最多是 2。

按照这个想法，你重新制作了一份三明治。与上次不同的是，你在里面加了很多酸黄瓜，甚至倒了一小碟醋。你满怀期待地咬下去，结果差点儿把已经吃进肚子里的上一个三明治吐出来。显然，你对酸度的重要性估计过高。

因为看了数学家写的书，所以你现在也有了一些数学思维。你将上一个故事中做出的第一个三明治的“美味程度”定为 5，这样的话，刚才那个加了很多酸黄瓜和醋的三明治美味程度最多是 2。显然，想要提升美味程度，就不能加太多酸黄瓜，更不应该加醋。于是你调整了权重，认为酸度的重要程度是 3，脆度是 2。你又想起上次的三明治好像只有火腿肉是有咸味的，生菜和面包吃起来都索然无味，所以这次你决定调整一下咸度，把咸度的重要程度调整为 4。

按照这个思路，这次你在两片面包中只放了 3 片酸黄瓜，同时稍微加了一些烧烤用的椒盐和孜然。做好之后你尝了一下，意外地发现这次做的三明治比第一次做的要好吃！你给这个三明治的美味程度评分为 6。

接下来，你又做了很多次试验，每次调整一个维度，然后试吃，打分，再调整，再试吃，再打分。就这样，一个星期过去了，在你吃完第 32 个三明治之后，你已经形成了一套属于自己的三明治配料清单，你只要按照这个清单去执行，就能做出特别好吃的三明治。

同时你知道，如果继续尝试添加新的食材，或者增加新的评价维度，然后继

续试吃，你未来有可能做出更好吃的三明治。

阿瑟·塞缪尔的跳棋程序最高明之处在于，它在与人类玩跳棋的过程中一定程度上可以自我学习，并提升自身的下棋技术。而这个程序学习的方法，跟5岁的小朋友不断试吃三明治的原理是一样的。

这个程序会在下棋的过程中针对三十几个具体的特征来不停地取值并赋予每一个特征值以一定的权重。比如说，双方棋子的数量差异是其中的一个特征，这个特征的取值方法就是直接用自己的棋子数量减去对方的棋子数量。而权重就相当于是一个重要程度的衡量指标。基于所有这些特征值和权重，程序会计算一个自己能赢的概率。这就像前面故事中的5岁小朋友把一个三明治的味觉因素分成咸度、酸度和脆度，并赋予它们不同的重要程度值，且每吃一个三明治都给它打一个分数是一样的。

接下来下棋的每一步，程序都会做上述计算，然后基于一些特定的规则对特征的权重做一些调整。在每一局棋结束之后，程序还会回顾整场对弈过程，确定哪些走法和评估导致了最终的结果，并进行一轮大规模的调整。这个过程就像前面那个小朋友不断试吃、打分的过程，如此循环往复，小朋友可以得到一个好吃的三明治配料清单，而跳棋程序则可以逐渐成长为一个下棋高手。

对于小朋友来说，这种不断的尝试是基于人类自身的聪明才智而完成的；而对于程序来说，则是基于不断的计算而完成的。从这个角度来看，冰冷的程序就像拥有了类似小朋友一样的智能，对下棋技术表现出一定的学习能力。

新罕布什尔的夏天很快过去了，达特茅斯会议也即将结束。当时的一部分与会者的合影如图1-4所示。

这是一张珍贵的历史照片，我想有必要介绍一下这里面的每一位学者，因为他们几乎都是日后的“大佬”。在介绍的过程中，会提到一些你可能尚不熟悉的技术名词，但是没关系，这些技术名词在本书的后续章节中会一一介绍。现在你只需要知道这些名词所代表的技术都很厉害即可。下面我们从左到右依次介绍。

最左侧坐在树下身着白色西装的人是奥利弗·塞尔弗里奇（Oliver Selfridge），他是我们常用的很多判别式人工智能模型底层技术的早期理论家之一，他提出的架构成为一种叫作**卷积神经网络**（Convolutional Neural Network, CNN）的思想基础，奠定了图像识别算法的发展基础，启发了现代人脸识别、智能安防和语音识别系统的设计。



图 1-4 达特茅斯会议部分与会者合影

坐在塞尔弗里奇后面穿着深色西装、打领结的人是时任 IBM 信息研究主管的纳撒尼尔·罗切斯特（Nathaniel Rochester）。罗切斯特是会议的发起人之一，他设计了 IBM 的第一台科学计算机 IBM 701，并开创了早期神经网络的计算机模拟实验，是工业界人工智能探索的先驱。

在这两位学者前面，穿格子衬衫的人是雷·所罗门诺夫（Ray Solomonoff），他开创了算法信息论和归纳推理理论，这两项理论对数据压缩、数据挖掘、自动驾驶等领域的发展有重要影响。

在所罗门诺夫旁边，同样位于前排身着浅色西装的人是特伦查德·莫尔（Trenchard More），时任达特茅斯学院电子工程系教授，后来致力于阵列理论的研究。在今天处理海量数据的并行计算中，依然能看到这种处理大规模数据阵列的数学逻辑的影子。

在罗切斯特和所罗门诺夫的后面，身着浅色衬衫、深色外套的人是马文·明斯基（Marvin Minsky），同样是会议的发起人之一，后来创立了麻省理工学院人工智能实验室。他提出了将心智拆解为许多模块化“代理人”的理论，这启发了现代 AI 工程师将复杂任务拆解成许多模块化小程序的设计思路，是复杂软件架构的基础。

位于图片最右侧身穿深色短袖衬衫的人是著名的克劳德·香农（Claude Shan-

non)，他是信息论的创始人，他的研究是所有手机通话、互联网数据传输和文件压缩的数学基础，可以说没有他就没有我们今天的数字世界。他还是计算机博弈论的鼻祖，早在1950年就写出了计算机如何下国际象棋的开创性论文，为后来“深蓝”超级计算机击败人类冠军奠定了理论基础。另外，香农同样是会议的发起人之一，并且他的影响力在很大程度上是本次会议可以顺利举办的前提。

最后，位于香农斜后方，身着白色衬衫、戴眼镜的人就是我们前面提到的发明了“人工智能”一词的约翰·麦卡锡（John McCarthy），他也是本次会议最核心的发起人。麦卡锡创立了Lisp语言和符号逻辑思想，这是早年的人工智能软件的理论基石，这些系统能根据一套清晰的逻辑规则来做医疗诊断或财务规划。此外，麦卡锡还是计算机分时系统的提倡者，正是因为他的推动，才让计算机从“独占使用”变成了可以多人同时通过终端操作，这算得上是今天云计算和互联网交互模式的远古祖先。

达特茅斯会议是人工智能发展史上重要的里程碑事件。它的意义在于，以一种看似不太正式的方式将一群拥有共同愿景的天才学者们聚集在了一起，明确了研究领域并激发了未来数十年的研究方向，构建了合作。更重要的是，该会议第一次提出了“人工智能”概念，并且让学术界中的很多人坚信，通过数字计算机实现机器智能是完全有可能的。

## 1.2 符号主义：用逻辑规则模拟人类推理

达特茅斯会议之后，与人工智能相关的研究继续如火如荼地进行。那时，随着电子计算机的飞速发展，用软件程序来模拟人类的推理过程成为可能，在这样的背景下，人工智能发展史上的第一个主流学派诞生了，这就是**符号主义学派**（Symbolicism）。

我们暂时先放下历史故事，来介绍一下符号主义学派的主张，以及由该学派产生的一些具体科研成果和行业应用。

在人工智能概念诞生之前，哲学家和数学家们就曾提出过一个观点，即人类的心智是一种“符号操作系统”。人类在思考和解决复杂问题时，会倾向于把现实中的概念和对象进行抽象，并使用相应规则进行推理。

举个简单的例子，我们平时开车或者过马路的时候，都会遵守交通信号灯的指示，最基本的规则之一是“红灯停，绿灯行”，这其实就是一个基础的“交通符号系统”。我们之所以会遵守这些规则，是因为我们的大脑中已经内置了一套形式化的逻辑结构。在符号主义的视角下，我们的大脑并不需要意识到“闯红灯会带来危险”，它只需要严格执行一条逻辑规则：“如果看到红灯（符号），就必须停下来（行为）。”而与之相反，可能还有一条规则：“如果看到绿灯（符号），就继续前行（行为）”。这种规则将复杂的现实行为简化成形式化符号和逻辑规则链，这是符号主义认为实现智能的关键步骤。

符号主义学派相信，他们不需要模拟大脑细胞，不需要特别关心这些规则是如何习得的，只需要完美地模拟规则本身就能实现智能。换句话说，如果我们可以收集世界上的所有知识，同时通过编程的方式来模拟人脑的判断过程，就可以实现人工智能。

这是一个非常清晰的思路，并且看起来是完全可行的。之后数年，很多学者基于这个方向进行了各式各样的研究，其中比较有代表性的作品是一个叫作伊丽莎（ELIZA）的聊天机器人，以及一个叫作 SHRDLU 的可以根据用户的指令搭积木的程序。

我们先来讲述伊丽莎，它可以模拟一位采用罗杰斯式疗法的心理治疗师跟用户对话，这种疗法通常通过重复、反射或鼓励患者自己阐述来引导对话。一段用户与伊丽莎对话的示意如图 1-5 所示。

```
Welcome to
          EEEEE LL   IIII  ZZZZZZ  AAAAA
          EE   LL   II    ZZ   AA  AA
          EEEEE LL   II    ZZ   AAAAAA
          EE   LL   II    ZZ   AA  AA
          EEEEE LLLLLL IIII ZZZZZZ  AA  AA

Eliza is a mock Rogerian psychotherapist.
The original program was described by Joseph Weizenbaum in 1966.
This implementation by Norbert Landsteiner 2005.

ELIZA: Is something troubling you ?
YOU:   Men are all alike.
ELIZA: What is the connection, do you suppose ?
YOU:   They're always bugging us about something or other.
ELIZA: Can you think of a specific example ?
YOU:   Well, my boyfriend made me come here.
ELIZA: Is it important to you that your boyfriend made you come here ?
YOU:   He says I'm depressed much of the time.
ELIZA: I am sorry to hear that you are depressed.
YOU:   It's true. I am unhappy.
ELIZA: Can you explain what made you unhappy ?
YOU:   -
```

图 1-5 伊丽莎扮演心理医生与用户对话

在这段对话中，伊丽莎问用户：“你有什么烦心事吗？”

用户回答：“男人都是一样的。”

伊丽莎问：“你认为这两者之间有什么联系？”

用户说：“他们总是为这样或那样的事来烦我们。”

伊丽莎问：“你能想出一个具体的例子吗？”

用户说：“是我男朋友让我来的。”

伊丽莎问：“你男朋友让你来这里，对你来说很重要吗？”

用户说：“他说我大部分时间都很沮丧。”

伊丽莎回答：“听到你说情绪沮丧，我很难过。”

用户说：“是真的，我不开心。”

伊丽莎问：“那你为什么不开心呢？”

……

第一次读到这段对话的时候，我的感觉就像在读小报上的星座运势分析，类似这种：

“本月金牛座的运势稍有起伏。你可能感到思绪纷乱，难以找准真实想法，面对选择时容易犹豫不决。这种状态虽可能让你错失一些小机会或感到疲惫，但也提供了重新审视自我、多角度思考需求的机会。无须过分担心，随着时间推移，思路会逐渐清晰，自我认知也将更进一步。”

这种文字的特点是：每一句话听起来好像都有一定道理，甚至会对你产生某种心理暗示，但细想一下，好像又没说出什么有价值的信息。但是不得不说，从整体的对话来看，屏幕上的伊丽莎好像真的像一个有点儿絮絮叨叨、有时思维跳跃、有时又能够切中你的所思所想的心理咨询师。

实际上，伊丽莎的“智能”并非基于对语言的真正理解，而是一种巧妙的符号操作技巧。

首先是识别关键词。伊丽莎会扫描用户输入的句子，并试图寻找预先设置好的一些关键词，例如我、你、父母、朋友、悲伤、高兴等。一旦识别到关键词，它就会执行相应的脚本规则，把句子结构进行重组或反射。

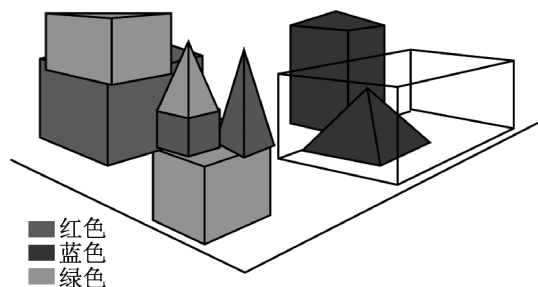
举例来说，用户输入“我很伤心，因为我养的猫离开了”。伊丽莎可能识别到了“我”，同时识别到“伤心”和“猫”，它会相对生硬地执行一个规则，把我换成“你”，把原文提到的“伤心”接在后面，然后在“猫”前面添加类似“谈

谈”之类的引导语，最终输出的句子可能会是“你似乎很伤心，能跟我谈谈你的猫吗？”

当然，程序的设计者也考虑到了一些特殊情况，比如有时可能没办法识别到有效的关键词，那么就会回复用户一些看起来通用的回答，例如，“请再说得详细一些。”

另外，伊丽莎所针对的场景也恰恰强化了用户对其“智能”的感受。因为罗杰斯式疗法又称个人中心治疗，其过程本来就是不断地引导用户针对自己的情况做一些阐述。可以想象，设计伊丽莎时，如果让它扮演的不是心理医生，而是客服的话，在这套规则作用下，可能聊不上几句，用户就要骂人了。

接下来再看 SHRDLU，这个名字好像没有标准的中文翻译，英文也很难发音<sup>①</sup>，所以我们不妨根据它的功能来给它取个名字，暂且叫作“积木世界”。顾名思义，这是一款搭积木的游戏，它的创新之处在于，这个程序能对用户输入的完整英文句子进行语法和语义解析，将其转换为机器可执行的符号命令，即用户可以用打字的方式操作这个程序以便在模拟世界中搭积木。积木世界示意如图 1-6 所示。



Person: Pick up a big red block.  
Computer: OK.  
Person: Grasp the pyramid.  
Computer: I don't understand which pyramid you mean.

图 1-6 积木世界示意

用户可以使用自然语言对该程序发出指令，该程序可以识别类似“在绿色立

① 据说这个名字来源于某种键盘上的字母键配置，所以本来就不是英文单词。

方体上面放一个红色的棱锥体”之类的指令，并在模拟世界中执行，即实现在模拟世界中搭建积木。如果用户的命令指代不清，它也会返回相应的提示信息并且拒绝执行任务。例如图 1-6 所示，用户说：“抓住棱锥体。”但是画面中有三个不同的棱锥，这时程序就会返回提示：“我不知道你指的是哪个棱锥体。”

积木世界背后的实现逻辑也是符号主义的思路。首先，程序已经预先设定了模拟世界中所有积木的符号属性，例如形状、颜色等。其次，它会解析用户的输入，并对照其符号库，以便识别要操作的积木是哪一块。另外，这个程序内部还设定了很多其他规则，例如类似“积木不能悬空”之类的物理规则。基于此，它可以在一定程度上将用户的自然语言表述转化为程序逻辑，并在屏幕上的虚拟世界中展现这种逻辑的结果。

它甚至实现了用上下文记忆功能来理解指代关系，例如，用户说：“把红色锥体放在绿色立方体上”，它会执行；此时如果用户继续说：“把它拿下来”，程序则会根据上下文识别此处的“它”就是指红色锥体。

所以，从用户的角度来看，感觉就像程序能够听懂他们的语言并且执行任务一样。然而，显而易见，积木世界程序的“智能”是基于其高度简化、可控的虚拟世界。在这个世界中，符号的数量有限、特征明确，用户发出的指令也相对单一且清晰。所以它只能在搭积木这一个细分领域表现出“智能”，如果用户问它类似“今天天气怎么样？”这样的问题，它会因为缺乏相关知识而无法理解。

“伊丽莎”和“积木世界”看起来都还像是实验室里面的简单模型，好像除了做演示和研究外，并没有太大的实用价值。然而，当时的科学家和工程师已经在思考如何通过这种方式来开发真正有用的人工智能。在符号主义的探索过程中，诞生了人工智能历史上第一次真正意义上的商业化浪潮，其核心产品就是**专家系统**（Expert Systems）。

顾名思义，专家系统就是像专家一样，可以解决某种专业问题的系统，类似的系统在 20 世纪 70 年代开始流行。它的基本逻辑是将人类专家的知识 and 经验，以大量的类似“如果……就”的形式存储在知识库中，并配合推理程序来解决实际问题。

迄今为止，专家系统依然在一些专业性较强的领域发挥作用，例如，在医疗诊断领域，输入症状，计算机可以给出可能的诊断。我们依然以一个虚拟的故事来类比这类专家系统的工作原理。注意，你即将读到的是一个关于问诊的故事，



这个故事是为了说明专家系统的原理而虚构的，如果你真的感觉有类似症状，还是要尽早就医，不要相信故事里面的“剧情”。

假设某医科大学附属医院刚刚发布了一个“在线医生”服务<sup>①</sup>，它的功能是根据用户对症状的描述，帮助用户判断可能的疾病。

你最近时常有鼻塞的感觉，有时还会感到很疲劳，其实你对这类症状的可能性有一定经验，可能是感冒，也可能是花粉过敏，但是并不确定，也不知道该吃什么药。于是你打开了在线医生服务，屏幕上显示出一个类似微信的对话界面，“医生”已经主动发来了第一条消息：“你好，我是人工智能刘医生，觉得哪里不舒服吗？”

这个感觉就像在医院看病一样，于是你回答：“鼻塞<sup>②</sup>。”此时，对话界面背后的专家系统正式开始运行。系统根据“鼻塞”这个线索检索了它的知识库，发现了3条相关的规则，每种规则对应一种疾病，分别如下：

- 如果鼻塞症状持续小于10天，且症状多为流涕或咳嗽，且无发高烧症状，则可能是普通感冒<sup>③</sup>；
- 如果鼻塞症状持续小于10天，且伴有全身酸痛和突发高烧，则可能是流感；
- 如果鼻塞症状持续大于10天，且伴有眼睛发痒，症状随季节变化，则应该是花粉过敏。

所以，系统需要从这3种可能性中进行判断。于是你看到了人工智能医生的下一个问题：“有没有突发的高烧？超过38.5℃的？”你回忆了一下，昨天好像有些许发热，但是肯定到不了这么高，于是你回答：“没有。”

此时系统将流感那一条规则优先级降低，然后人工智能医生继续问：“你是从

---

① 写作本书时，大语言模型已经开始流行，所以你现在用到的类似线上问诊的服务很可能是基于大语言模型来运行的，这是另外一种更先进的技术，其实现思想与后文提到的联结主义密切相关。但是此处阐述的是20世纪70年代流行的另外一种问诊的实现方式，即专家系统方式，其实现的主要思想是符号主义。

② 你可能觉得只回复两个字会有些奇怪或不礼貌，但这个故事试图模拟的是20世纪70年代的专家系统，那时的人工智能还没办法对自然语言进行准确识别，所以这里把这名患者的输入故意设定为类似“关键词”的形式，而不是句子。

③ 感冒一般是大众对“上呼吸道感染症状群”的俗称，并不是一个精确的医学诊断，但此处为了方便阅读，沿用类似的称呼。

什么时候开始感到不舒服的？”你想了想回复说：“3天前。”系统发现“症状持续小于10天”这个符号成立，此时，普通感冒的那一条规则被强化。但是，为了保持严谨，还需要继续问诊。于是你又看到了一条消息：“有没有鼻塞、咳嗽或者全身肌肉疼痛的症状？”

你回答：“有咳嗽，但没有感觉全身肌肉疼痛。”此时，系统发现你的描述已经满足普通感冒对应的所有规则，于是它告诉你：“从你的症状来看，应该是得了普通感冒，一般情况下，普通感冒会在7~10天内康复。如果你希望缓解症状，也可以服用一些药物，[点击此处查看用法与用量](#)。”

点击屏幕上的链接，你发现医生推荐了两种相关的感冒药，都是非处方药，于是你记下药名，然后前往你家附近的药房购买了这两种药。

总结一下，符号主义是人工智能历史上第一个主流学派，他们的核心观点可以总结成一个公式，叫作“智能 = 知识表示（符号）+ 逻辑推理（规则）”在这种思维的作用下，他们成功研发出了第一类可以商业化的人工智能软件——专家系统。虽然这种软件只能在某些特定领域进行相对有限的推理，但是从外部来看，显然其行为已经有一些智能的意味了。

但是，伴随着学者们的乐观态度，危机却在悄悄地降临。

### 1.3 寒冬来临：知识获取瓶颈与常识鸿沟

在符号主义学派的推动下，越来越多的人工智能程序被编写出来，而这类技术的局限性也随之显现。接下来用另外一个故事来说明这种局限性。

假设你当时是一所大学里的研究生，且刚刚参加了麻省理工学院举办的一次人工智能学术交流活动。在会议上，你看到了关于积木世界的演示，会后甚至上台亲手操作了几轮。你明白，之所以可以通过简单的句子操控软件程序移动虚拟世界中的积木，是因为程序中写满了各种各样的类似“如果……就”的逻辑规则。

在回程途中，你一直在思考这个问题：如果可以通过编程的方式来让计算机“听懂”指令，并移动虚拟世界中的积木，那么是否也可以使用同样的方式来指挥机器做一点儿更酷的事情呢？比如在虚拟世界中开一家“虚拟咖啡馆”。

你为这个想法感到兴奋，甚至觉得明年的图灵奖非你莫属。之后的几天，你



开始着手开发这个做咖啡的人工智能。

方法是明确的，首先你试图建立虚拟咖啡馆中的符号库。你发现，自己面对的符号明显更多、更复杂。对于积木世界来说，它可能只需要定义 10 种形状的积木和 3 种颜色，但是在咖啡世界中，则必须将所有相关实体和属性全部符号化。于是你尝试列出这些符号，甚至将它们分门别类。

第一类是对象符号，包括马克杯、预制咖啡粉<sup>①</sup>、水、滴滤咖啡机、玻璃咖啡壶、勺子、奶精壶和糖。其实就在昨天，你为了实现这个想法跑到图书馆旁边新开的一家咖啡馆里仔细观察过整整一天，知道除了上面这些，其实咖啡粉的研磨颗粒度还分粗细，咖啡杯也可以再细分。例如，只有客人要求在店里喝咖啡的时候，才会使用马克杯；如果客人需要打包带走，一般使用的是那种白色的泡沫塑料杯。此外，你还观察到了一些自己不知道用途的工具，但是转念一想，算了，别搞那么复杂，对象符号就暂时设置为上面那几个吧。

第二类是状态符号，这个更复杂，比如可能有冷水、热水、温水、空的马克杯、满的马克杯、脏的马克杯、干净的马克杯、关机状态的咖啡机、开机状态的咖啡机、干净的玻璃咖啡壶、脏的玻璃咖啡壶……你隐约觉得，事情好像有些不妙，因为这些状态可能都是基本状态，你还能列出更多。

还有第三类，即动作符号，例如抓（固体）、倒（液体）、打开（设备）、关闭（设备）、放（固体）、搅拌（仅针对勺子）、混合（两种或以上液体）……有一些动作甚至难以定义清楚，比如把两张粘在一起的滤纸分开。

你花费了整整两天时间，列出了满满 2 页草稿纸那么多符号。

接下来你开始尝试编写规则。你发现，随着步骤、对象和状态数量的增加，规则的复杂度会呈几何级数增加。你又花了大半天的时间，在计算机上输入了上百行类似“如果……就”的判断逻辑，回头一看，才只是把咖啡机状态判断、杯子状态判断、舀咖啡粉、加水等几个场景写完。这意味着目前在理论上你的人工智能只能做出普通的黑咖啡。

而真正让你崩溃的是，当你尝试让人工智能做第二杯咖啡的时候遇到的问题——在开工之前，你想到应该检查一下马克杯，确保它“足够干净”。但是发现，你根本定义不清楚什么算“足够干净”。相比之下，积木世界对于符号特征的

<sup>①</sup> 我并不是故意要让主角做“预制菜”，原本我想写咖啡豆，但是查询了相关资料发现那个年代的美国不流行现磨咖啡，全是预制咖啡粉，所以为了让剧情更加真实，只能写“预制菜”了。

定义非常简单，一块积木如果是正方体，它就不可能是锥体；如果是红色，它就不可能是蓝色。但是“干净”这个状态，你想不到该如何写程序进行判断。

在崩溃之余，你只好选择暂时离开实验室出去透透气。天色已经明显暗了下来，你漫步在校园中，不知不觉又来到了那家咖啡馆门口。你推开门，进去点了一杯冰茶，坐在窗边，一边喝茶，一边看着窗外行色匆匆的景象。这时，坐在你邻桌的一位女士叫住了路过的店员，她说：“麻烦给我一些冰块。”店员点点头，熟练地进入工作台，取出一只玻璃杯，用铲子装了半杯冰块，然后回到女士桌前，放在了桌上。

作为计算机专业的你，突然之间意识到了另外一个问题——比列出符号、写出逻辑更难的问题，即如何让程序理解一些常识。你可以想象得到，如果刚才路过的不是一名店员，而是一个计算机程序的话，当它收到“给我一些冰块”（Bring me some ice）这样的指令时，它很可能会直接去工作台上抓起一把冰块，然后向那位女士的桌子的方向扔去。人类掌握了很多常识，比如冰块不能直接运输，而应该把它们放到某种容器里面再运输，但计算机程序没有这类常识。

结合你学习过的专业课程，你明白了，在人工智能的判断规则里，试图用“如果……就”的方式来把某类常识写清楚几乎是不可能完成的任务。

最终在第三天晚上，你放弃了“即将获得的”图灵奖，显然，在符号主义主导的1973年，用人工智能做咖啡是个不太现实的想法。

根据上面这个故事，我们可以总结出符号主义学派所主张的人工智能实现方法上的三大缺陷。第一个缺陷叫作**知识获取的瓶颈**，如果我们希望将整个世界的知识全部抽象成符号和规则，这个工程量是难以想象的。第二个缺陷是**组合爆炸**，对于稍微复杂的推理问题，机器多思考一步，需要的选择可能就增加上百甚至几百倍，最终会像滚雪球一样失控。程序的计算量增长速度远远超过了当时计算机的算力上限。第三个缺陷是**常识的鸿沟**，人类的智能除了体现在知识和推理层面之外，还体现在很多常识层面上，例如“冰块需要用杯子装”，以及情感、文化、习俗等。而这些常识性的元素几乎是不可能使用逻辑判断的方式来表达的——就像你很难用逻辑写清楚你的女朋友为什么生气一样。

然而，雪上加霜的是，当时的人工智能除了技术局限性之外，还面临着另外两方面的质疑和压力。

一方面的质疑和压力来自社会大众。1956年至1970年这段时间是人工智能



发展的黄金年代之一，符号主义的先驱们对机器的潜力做出了极度乐观甚至不切实际的预测，并且这种预测经常会被媒体公之于众。

举例来说，明斯基作为麻省理工学院人工智能实验室的联合创始人，在1970年对《生活》杂志做出了一个著名的预测，他认为：“在3到8年内，我们将拥有一台具有普通人类智能的机器。”而西蒙这位诺贝尔奖得主和符号主义先驱，早在1957年就乐观地预言：“在10年内，一台数字计算机将成为国际象棋世界冠军，并在定理证明方面发现重要的数学定理。”

事实上，真正成为国际象棋世界冠军的人工智能，要到40年后的1997年才问世。在西蒙抛出豪言后的若干年内，别说是国际象棋，人工智能连“读懂”一本3岁小孩的童话书都做不到。而明斯基所谓具有普通人类智能的机器，迄今为止尚未实现。虽然现在的大语言模型，如ChatGPT、DeepSeek等在处理语言、编程和常识推理方面展现了惊人的能力，但它们仍然不被认为具有普通人类的通用智能（AGI）。显然，当时人们看到的结果与学者们预测的结果大相径庭，完全达不到预期。

另一方面的质疑和压力则来自经济和政治因素。众所周知，大学的科研往往需要经费资助，当时资助人工智能相关研究的机构，很大程度上都有相关政府背景，但是这些资助者往往对相应的科研成果并不满意。1966年，美国国家研究理事会资助的“自动语言处理咨询委员会”发表了一份关于机器翻译进展的报告。报告认为机器翻译“没有取得任何突破性的成果”，并且人工翻译比机器翻译“更便宜、更准确”。1973年，英国科学研究委员会发布了著名的《莱特希尔报告》，认为相关研究“未能在实践中解决任何有实际价值的问题”。

如果你对上述内容没什么感受，这里可以举一个机器翻译界广为流传的经典案例。当时的研究人员经常使用“回译”的方法来测试机器翻译的能力，即把句子从源语言翻译成目标语言，然后再翻译回来，以此来看翻译回来的意思跟源语言表达的意思是否一致。

但是，由于当时的机器翻译系统几乎完全依赖符号主义的方法，导致对于多义词、语境等问题的理解堪称灾难。举例来说，有这样一句英语：“The spirit is willing, but the flesh is weak.”中文的意思是“心有余而力不足”。研究人员尝试把它译成俄语，再将翻译后的俄语译回英语，最终得到的英语是“The vodka is good, but the meat is spoiled.”中文意思是“伏特加是好的，但肉坏了”。与之前的

源语言的意思大相径庭。

出现这种问题的原因是，当时的人工智能没办法有效判断多义词在句子中表达的意思。这句话中 **spirit** 这个词有两个常见的意思，分别是“精神”和“烈酒”，我们把原句中的 **spirit** 理解成中文的“心”其实取的是第一个意思。但是人工智能将这个词翻译成了“烈酒”，而在俄语中，“烈酒”又与“伏特加”是近义词，所以 **spirit** 这个词从英语翻译成俄语，再翻译回英语就变成了 **vodka**，完全打破了原有的含义。至于后面翻译成“肉坏了”是类似的道理，人工智能直接把 **flesh**（肉体）翻译成了俄语的“肉”，而在“烹饪”的语境下，形容肉“软弱无力”的 **weak** 则变成了“不新鲜、变质了”。于是，一句充满哲理的话就这样变成了关于糟糕晚餐的抱怨。

从政府机构的视角来看，相当于数千万美元的投入换来的只是一些几乎没办法使用的技术，他们对此很不满，最终停止了很多资助，相应的研究项目也因此被迫中断。

在技术本身的局限性、公众的失望及政府机构撤资三方面压力下，人工智能的发展进入了历史上第一个寒冬。在这个时期，“人工智能”这个概念在学术界和投资界声名狼藉，很大程度上成为“不切实际”“浪费资金”的代名词。有不少学者甚至开始避免在他们的项目名称中使用“人工智能”这个术语，以寻求资助。

接下来，20世纪80年代的故事有些乏善可陈。专家系统曾经引领了人工智能行业的短暂复苏，在这一发展过程中，更多更大型的专家系统被开发出来，并且成功用于一些不同的行业，如医疗、企业资源规划、供应链管理、工业管理等。

其中最著名的产品莫过于DEC公司的XCON系统，它由卡内基梅隆大学研发，其功能是自动配置DEC销售的计算机系统，这类系统高度模块化，客户可以定制处理器、内存、硬盘、接口卡等数百个组件。其实，所谓自动配置听起来很高级，实际上可以把它简单地理解为一个“组装电脑选配件”的工作——但是可选的配件比一般电脑复杂。之前这类工作由人类专家人工完成，经常会产生错误配置，即可能把两个没办法搭配在一起工作的配件选定在一起。XCON系统投入使用之后，据说将错误率由最高时的35%降低至2%。据评估，该系统每年可以为DEC公司节省2500万美元到4000万美元的成本。

但是，一方面这些专家系统并没有解决之前提到的知识获取的瓶颈、组合爆炸，以及常识的鸿沟等问题；另一方面，人们还发现了一些新的问题，即这种系



统一且庞大，维护和扩展都会变得非常麻烦——不但没解决之前的问题，反倒又有了新的问题。

除此之外，在这个过程中也穿插了一些其他故事，例如日本投入了大量资源，希望研发出“第五代计算机”。在日本学者的设想中，这种新型计算机将采用全新的信息处理架构，以便它可以更加方便、高效地完成自然语言理解、逻辑推理等任务。在很大程度上，相当于把符号主义学派的理论硬件化。

后来，日本的学者们虽然成功做出了这种新型计算机的原型机，但是实际测试效果并不理想，最终随着符号主义本身的衰落，五代机也被战略性放弃。

到了20世纪80年代末，随着个人计算机（PC）性能的飞速提升和价格下降，专门用于专家系统开发和运行的硬件体系市场崩溃，人工智能的发展在这个阶段进入了第二个寒冬。

然而，时代的车轮滚滚向前，虽然这两次寒冬在很大程度上标志着符号主义黄金时代的结束，但并不能抹去这些先驱们对人工智能的贡献。另一方面，随着计算机行业的快速发展，越来越多的电子化数据正在全世界各个角落产生，后来互联网的流行则成百上千倍地加速了这个过程。基于这些数据，人工智能行业正在酝酿着新一轮的变革与发展。在第2章，我们会看到另外两个学派一明一暗地发展，它们会以完全不同的思路来再次点燃人们对人工智能的热情。

在本章的最后，我认为还是有必要交代一下图灵的结局。就在他发表那篇著名的论文《计算机与智能》的两年后，1952年1月的某天，图灵的住所被盗。图灵随即报警，在调查过程中，警方怀疑窃贼跟一个名叫阿诺德·默里的人有关，在警方的盘问下，图灵承认了他与默里的同性恋关系。据说，图灵当时提供了一份长达5页的详细陈述，但这份文件却成为警方眼中的“绝佳供词”。

后来戏剧性的一幕出现了，警方的调查方向迅速由入室盗窃转向了“严重猥亵罪”。两个月后的3月31日，英国政府对图灵提起公诉，案件的正式名称叫作“女王诉图灵和默里”（*Regina v. Turing and Murray*），其中 *Regina* 在拉丁语中意为女王，在英联邦国家代表公诉方。在法庭上，图灵没有进行抗辩，他对所有指控均表示认罪。当庭的法官 J·弗雷泽·哈里森认为图灵是国家的功臣，因此给了他一个“仁慈”的选择，要么入狱服刑，要么缓刑，但前提条件是必须接受强制性“治疗”，即俗称的“化学阉割”。

或许是希望继续他的学术工作，或许是由于恐惧，图灵选择了后者。接下来，

强制性的“治疗”持续了一年，给他的身体和精神都带来了严重的影响。最终，1954年6月7日，图灵被他的女管家发现在家中去世，年仅41岁。

关于死因，验尸官的报告确定为氰化物中毒。在图灵的床边，人们发现了一个吃了一半的苹果，因此有人怀疑他是为了模仿他最喜欢的童话故事《白雪公主》而食用了浸泡过氰化物的苹果自杀的。但是，据说当局从未对那半个苹果进行过氰化物检测。即这些致死的氰化物究竟来源于何处并没有一个明确的答案。所以严格来讲，图灵的死成了谜。

伟大的人总是应该有一些异于常人之处。不论是苹果还是氰化物，都为图灵的死披上了一层神秘感。但毫无疑问的是，在学术上图灵奠定了现代计算机和人工智能的基础，无愧于“计算机科学之父”和“人工智能之父”的称呼；在战场上图灵成功破译了德军的密码，让盟军在战争中取得了优势，间接拯救了数百万人的生命，但最终却被他所拯救的国家以“法律的名义”推向了死亡。

近20年后，即从20世纪的70年代中期开始，二战时期的文件逐渐被解密和公开，直至20世纪80年代以后，图灵的工作和贡献才被民众广泛知晓。后来，英国乃至全世界的民众的反应经历了震惊、愧疚、愤怒及尊敬几个阶段。在公众舆论的巨大压力下，2009年，时任英国首相戈登·布朗发表了一份官方声明，代表英国政府正式道歉。随后，要求对图灵进行皇家赦免的呼声日益高涨，并受到了包括斯蒂芬·霍金在内的著名科学家的联名支持。最终，英国女王伊丽莎白二世在2013年破例宣布赦免已去世近50年的图灵。

讽刺的是，宣布赦免图灵的这位女王，跟当年提起公诉时作为公诉方象征的女王是同一个人。

历史总是充满了这样的唏嘘或荒谬。或许在一百年之后，那个当年审判了图灵的法官，如果不是因为图灵，没有人会再记得起他的名字；而那些象征世俗权力的王冠与权杖，也终将褪去光芒。但毫无疑问，艾伦·图灵的名字将永远镌刻在人类的科技史上。即便在一百年以后的大学课本中，即便在未来的“赛博科技”世界里，即便到了硅基生命与碳基生命的人类平起平坐的那一天，他依然是那个最初开启了智慧曙光的先驱。