单位代码 10006

学　　号 22371554

分 类 号 TP301.1

北京航空航天大学

BEIHANG UNIVERSITY

图灵机模型综述

学 院 名 称	计算机学院
专 业 名 称	计算机科学与技术
学 生 姓 名	曾华旭
指 导 教 师	吕江花

2024 年 4 月

图灵机模型综述

学　　生：曾华旭

指导教师：吕江花

摘　　要

图灵机模型，作为理论计算机科学的核心基石，源于艾伦·图灵在 1936 年发表的一篇具有划时代意义的论文。这一模型构建了一个抽象的计算设备，旨在模拟和实现任何可计算的过程，从而奠定了现代计算机科学发展的理论基础。图灵机的提出，不仅标志着计算理论的诞生，更为后续计算机设计、编程语言研究等提供了重要启示。

本文回顾了图灵机模型的提出背景，探寻其理论源流，并对图灵机的直观表达、形式描述、基本概念和所解答的问题予以简要阐述，梳理其中蕴含的可识别、可判定和可计算等问题。在图灵机的基础上，研究者们提出了多种变形模型，这些模型虽然在形式上有所不同，但都与图灵机在计算能力上等价。本文将对这些变形模型进行讨论，并探讨其等价性论证的思路。此外，随着数字计算机的出现，计算的可能性即复杂性问题逐渐成为研究的焦点。研究者在图灵机模型为基础上提供了定义时间复杂度、带复杂度等概念的方法，从理论上深入地研究计算的效率和资源消耗问题。本文将介绍其定义和研究成果。

最后，本文还将基于图灵机模型的研究现状，尝试分析未来的潜在发展方向。

关键词：图灵机，可判定性，多带图灵机，非确定性图灵机，时间复杂度，带复杂度

Turing Machine Model Overview

Author：ZENG Hua-xu

Tutor：LYU Jiang-hua

Abstract

The Turing machine model, a core cornerstone of theoretical computer science, originated from a groundbreaking paper published by Alan Turing in 1936. This model constructs an abstract computing device intending to simulate and implement any computable process, thus laying the theoretical foundation for the development of modern computer science. The proposal of Turing machines not only marked the birth of computational theory, but also provided important insights for subsequent computer design, programming language research, and more.

This article reviews the background of the Turing machine model, explores its theoretical origins, and briefly describes its intuitive expression, formal description, basic concepts, and the problems it addresses. It also examines the problems of recognition, decision-making, and computation that are inherent in it. Based on the Turing machine, researchers have proposed various modified models, which are different in form but equivalent in computational power to the Turing machine. This article will discuss these modified models and explore the ideas for demonstrating their equivalence. In addition, with the advent of digital computers, the possibility of computation, or complexity, has gradually become a focus of research. Researchers have provided methods for defining concepts such as time complexity and tape complexity models based on the Turing machine model, and have conducted in-depth theoretical research on computational efficiency and resource consumption issues. This article will introduce their definitions and research results.

Finally, this article will also try to analyze the potential future development directions based on the current research status of the Turing machine model.

Key words：Turing Machine, Decidability, Multi-tape Turing Machine, Nondeterministic Turing Machine, Time complexity, Tape complexity

目 录

1 引言

1936 年，为了回答希尔伯特的判定性问题 Entscheidungsproblem，图灵构造了图灵机模型，给出了计算过程的清晰视图，并证否了上述问题。

图灵机模型启发了许多数学家和计算理论研究者关于语言识别和可计算问题的研究，沟通了可计算性的数学问题和真实计算机器的实现，不仅是自动机研究的原型，更启发了数字计算机的架构；在计算机真实机器出现后，计算面临着效率和资源利用的问题，计算复杂性问题提上日程，图灵机被用于定义和研究计算复杂性。

作为计算理论的初学者，有必要对图灵机模型的来源和发展进行梳理。本文第二章将概述图灵机模型，介绍其来源，讲述其定义，分解其基本概念，并对图灵机衍生的识别和可判定性问题做简要介绍；第三章介绍了图灵机模型的发展，主要探讨了几个具有等效识别能力的变形，以及在图灵机变形上定义的计算复杂性问题。第四章尝试从图灵机模型的发展历史和更近的研究中分析研究现状和未来方向。

2 图灵机模型概述

2.1 图灵机的研究动机

20 世纪初期，以希尔伯特为代表的数学家希望能以形式逻辑方法得到全体数学问题的完美解决，提出了“希尔伯特纲领”，包括完成数学的形式化、证明数学的完备性、一致性和可判定性。其中形式化指全体数学命题能够表示为一套严格的语言并依据严格的规则来使用，完备性指的是依据系统的推演规则能够推演出全体语义（Semantics）为真的命题，一致性指一个系统不能同时推演得到一个命题和它的否命题，可判定性则指的是存在一种确定的算法来判断任何一个数学命题的真伪[3]。

1929 年，维也纳学派数学家哥德尔证明了完备性定理，指出了一阶谓词逻辑系统的完备性。即如果一个公式在逻辑上是有效的（对于任一模型均为真），那么这个公式就有一个有限的形式证明，即具有完备性[4][6]。然而，完备性不能在更强的公理系统中成立。1931 年，哥德尔证明了不完备性定理，其一是任何满足一致性的形式系统，只要蕴涵了皮亚诺算术公理，就可以构造出在体系中不能被推演出的真命题，即包含

皮亚诺公理的体系是不完备的；其二是前一结论附带的推论，任何逻辑自洽（满足一致性）的形式系统，只要蕴涵皮亚诺算术公理，它就不能用于证明其本身的自洽性[5][7]。

这一论证打破了希尔伯特关于完备性的构想，但是关于希尔伯特在 1931 年表达的可判定性问题[11]尚未给出否定的答案。为了回答该可判定性问题（Entscheidungsproblem），首先需要刻画什么样的对象是可计算的，一些数学家提出了纯数学过程进行刻画，如丘奇的“有效计算”[8]，波斯特系统[9]，以及克林的一般递归函数[10]等。

同样为了回答这一问题，图灵提出了模型“计算机（Computing Machine）”，即“图灵机（Turing Machine，简称 TM）”，用以描述“任何可能计算[1]”的过程，并用其论证了一阶谓词系统不可判定[1]。这一模型不同与其他模型的特点在于不仅描述了可计算性，解答了可判定性问题，还给出了计算过程的清晰视图，沟通了可计算性的数学问题和真实计算机器的实现。

2.2 图灵机的定义

2.1.2 图灵机的直观描述

图灵机有有限的状态 $q_0 \sim q_R$，称作机器格局（m-configuration）。一条包含离散单元（Square）的无限长一维纸带穿过该机器，每个单元对应一个字符；任意时刻，只有一个单元处在机器内部。图灵将当前机器读到的字符内容和机器格局的二元组称作格局（configuration），它决定了接下来机器将往纸带上写下的内容，机器移动选择，以及机器格局的更改[1]。

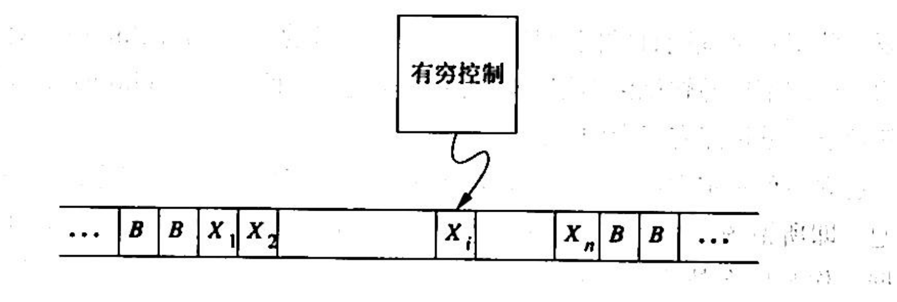

图 1 图灵机示意图[2]

2.1.2 图灵机的形式定义

结合图灵的直观描述和初步的形式化描述，后人给图灵机进行了一些更现代的形式化定义，这里介绍课程中讲授的定义，该定义被有关教材[2]采用。图灵机可用一个七元组表示：

$$M=(Q,\Sigma,\Gamma,\delta,q_0,B,F)$$

其中，Q 表示图灵机状态的有限集合；Σ 表示输入字符的有限集合；Γ 表示纸带上的字符的有限集合，$\Sigma\subseteq\Gamma$；$\delta:=Q\times\Gamma\to Q\times\Gamma\times\{L,R,S\}$ 作为状态转移函数规定了某一状态的图灵机在某个输入字符的格局下图灵机的下一步操作；q_0 是初始状态；B 为空白字符，$B\in\Gamma\wedge B\notin\Sigma$。$F\subseteq Q$ 表示接受状态集合，当图灵机状态属于 F 时图灵机接受了该输入；若图灵机经过任意转移步骤后，状态都不在 F 中，则表示图灵机拒绝给输入，可以看出，拒绝包含了两种情况，即转移函数 δ 在运行中的某个格局下无定义或图灵机永远不停止。

2.1.3 图灵机的瞬时描述和状态转移

由于输入有限，且在有穷多步移动后图灵机所访问的单元有限，因此有限步移动后非空单元有限。对从初始状态进行若干步移动后的图灵机，可以用最左边非空格与最右侧非空格之间的符号构成的串来进行瞬时描述（Instantaneous Description，简称 ID）。即：

$$X_1X_2\cdots X_{i-1}qX_iX_{i+1}\cdots X_n,\ X_{\mathrm{k}}\in\Sigma,\ \mathrm{p}\in Q,$$

称作图灵机的格局[2]。

记号├用来表示格局的转换，而├*则为├的自反、传递闭包，表示 TM 的零步、一步或多步移动。状态转移表或图片用于可视化呈现图灵机的状态转移函数。“计算”的过程可借此模型形式化表达出来，即计算是从初始格局到终止格局按照动作函数规定的规则进行的一系列转换的格局转换序列。

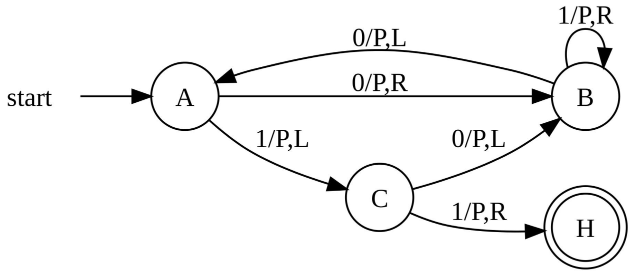

图 2 图灵机状态转移图示例[13]

2.3 图灵机的识别和判定问题

2.3.1 图灵机识别（recognize）的语言

被某个图灵机接受的输入集合称为图灵机识别的语言。形式化地表述，即

$$L(TM)=\{\omega\in\Sigma^*\mid \mathrm{q}_0\omega\vdash^*\alpha q_f\beta\},q_f\in F$$

为了研究图灵机能力的上限，有必要对其能够识别的语言的集合进行研究；全体图灵机所能够识别的字符串集合为递归可枚举（Recursive Enumerable，简称 RE）集合[2]。图灵构造了任何图灵机都不能识别的语言 L_{d}，借鉴哥德尔数的思想，他提出了通用图灵机，将每个图灵机进行编码，证明了图灵机集与自然数集等势；借鉴康托尔对角线方法，他将图灵机序列对于串的接受表的对角线取反得到了 L_{d}。可以证明 L_{d} 不被任何一个图灵机识别[1][2]，这论证了图灵机能力存在上限。

2.3.2 可判定性（decidability）和停机问题

图灵借助图灵机描述了普遍的计算过程，如何将此模型用于回答希尔伯特的可判定性问题呢？图灵首先证明了不存在 TM E，对任意 TM M，E 确定 M 是否打印出一个给定的符号（如 0）。接着对每个 TM M，构造了一阶谓词公式 Un(M)（在 M 的某个格局下 S1（即 0）出现在纸带上），并且证明，如果存在通用的方法判定 Un(M)是否可证明，就存在通用方法判定 M 是否可打印 0。利用前后的矛盾证明了一阶谓词系统的不可判定性[1]。

由于图灵的论文中，图灵机以计算和停机的方式被使用，因此上述问题也被总结为“停机问题”。所谓“停机问题”即对任意图灵机和任意输入，是否能在有限的步骤中进入接受格局或拒绝格局。该问题被证明是不可判定的，即存在图灵机，其停机问题不可递归解决[12]。

可以使全体图灵机都停机的集合称作递归集，或形式化地表述，

$$L = L(M), s.t.(w \in L \rightarrow M\text{接受}w \wedge w \notin L \rightarrow M\text{停机但不接受w})$$ [2]

则 L 称为递归语言。可以看出全体递归语言可以被图灵机识别，因此一定是递归可枚举的，两者的关系如图 3 所示。

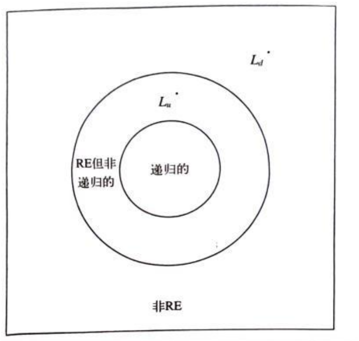

图 3 递归集、递归可枚举集以及非递归可枚集的关系[2]

3 图灵机模型的发展

图灵借助图灵机模型建模了计算的过程，并借以回答了希尔伯特可判定性问题，推动了数理逻辑的发展。随着真实世界数字计算机研究需求和实现发展，许多更接近现实机器、面向更丰富情景的图灵机变形开始出现，它们被称作自动机。W. S. McCulloh 和 W. Pitts（1943）以及 D. A. Huffman（1954），Mealy（1955），Moore（1956）等人研究的有限自动机（Determinist Finate Automata）、P. C. Fisher（1963）和 M. P. Schutzenberger（1963）等人研究的下推自动机（Push-Down Automata）、John Myhill（1960）等人研究的线性有界自动机（Linear Bounded Automata）和它们的各种变体等都可看作以图灵机为起点的识别不同范围语言的自动机。

自动机理论已发展形成计算理论的一个完整分支，因此本章仅聚焦与图灵机具有同等识别能力的几种图灵机变形，即恰能识别递归可枚举语言的图灵机变体，而不深入探讨范围更广的其他自动机；对这些变形的识别能力等价性和时空复杂度性质进行讨论。

3.1 图灵机具有同等识别识别能力的变形

3.1.1 多带图灵机（MutiTape TM）

该图灵机具有多条纸带穿过机器，每个带上有一个带头，各个带头可以独立移动。可以使用形式化方法描述这一机器。其七元组结构除了转移函数δ之外的其他部分均与图灵机一致，δ定义为$\delta := Q\times\Gamma^k \to Q\times(\Gamma\times\{L,R,S\})^k$，其中 k 是纸带的条数。

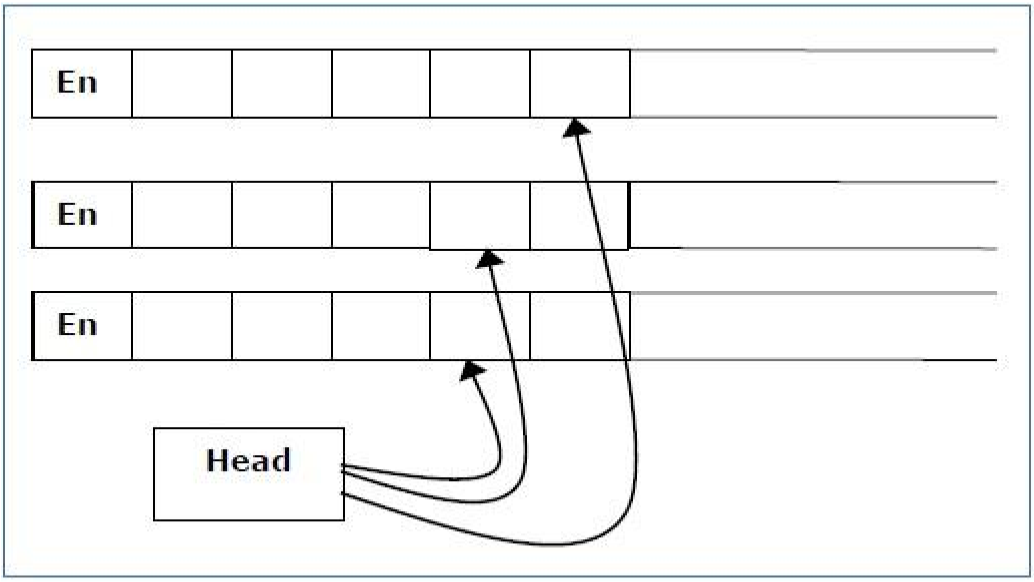

图 4 多带图灵机[14]

可以论证此图灵机变形同图灵机识别的语言集合一致：单带图灵机是多带图灵机的特例，因此每个单带图灵机都可以用自身这个“多带图灵机”模拟；故只需论述每个非确定型图灵机都可以用一个确定型图灵机来模拟即可说明两个模型等价。其思路是对一个具有 k 条纸带的多带图灵机 M，只需要用具有 2k 道的单带图灵机 N 进行模拟，其中奇数道记录了对应带带头位置，偶数道对应了原来的带的信息。通过把 N 左侧的带头标记个数记录在 N 状态中，N 可以实现依次访问 M 的带头对应位置，并用 k

轮的移动和迁移（从一个带头附近一直寻找到下一个带头）逐一模拟 M 各个带头的读、写、移动行为，其中移动体现为改写奇数道的带头位置，从而模拟 M 的一次移动[2]。这个 N 可以如下形式化表达：

$$N=(Q\times\{0,1,2,\cdots k-1\},\Sigma\times\{B\}^{k-1},\Gamma^{k},\delta',<q_0,x>,B^{k},F\times\{0,1,2,\cdots,k-1\})$$

$$\text{其中}\ \delta':=(Q\times\{0,1,2,\cdots,k-1\})\times\Gamma^{k}\rightarrow(Q\times\{0,1,2,\cdots,k-1\})\times\Gamma^{k}\times\{L,R,S\}$$

3.1.2 非确定图灵机（Non-Deterministic TM）

非确定图灵机在状态转移上和图灵机有所不同，转移函数并非使从机器某个状态转移到另外一个状态，而是使其从某个状态转移到某几个状态中的任意一个。同样可以形式化表达这一机器。其中七元组结构除转移函数 δ 外和图灵机一致，而 δ 定义为 $\delta:=Q\times\Gamma\rightarrow\rho$（$Q\times\Gamma\times\{L,R,S\}$）， ρ 指幂集。非确定图灵机可以从三元组集合中任选一个进行作为下一步移动，只要最终存在一个转移序列使图灵机进入接受格局则认为图灵机接受了该输入。

同样可以论证此图灵机变形和图灵机识别的语言集合一致，由于确定型图灵机是非确定型图灵机的特例，因此只需论述每个非确定型图灵机都可以用一个确定型图灵机来模拟即可。用多带图灵机M_D来进行模拟上述多带非确定图灵机M_N，其中M_D的一条带记录M_N的全体瞬时描述 ID（可以用分隔符如*分开）[2]，一条带记录M_N的当前位置。如果M_D读到的 ID 不是接受格局，则将其在带末尾复制 k 次，根据转移函数的指定的 k 个下一状态对这 k 个 ID 一一修改，最后把当前位置向后移动一个 ID 单元。由于具有 m 状态的非确定图灵机的在 n 步中可能达到的格局数不超过 nm^n 个，因此M_D一定能够在 nm^n 之内的步骤遍历M_N的 n 步转移的全体序列，即M_D可以模拟M_N[2]。

3.1.3 具有半无穷带的图灵机（Semi-Infinite Tape Turing Machine）

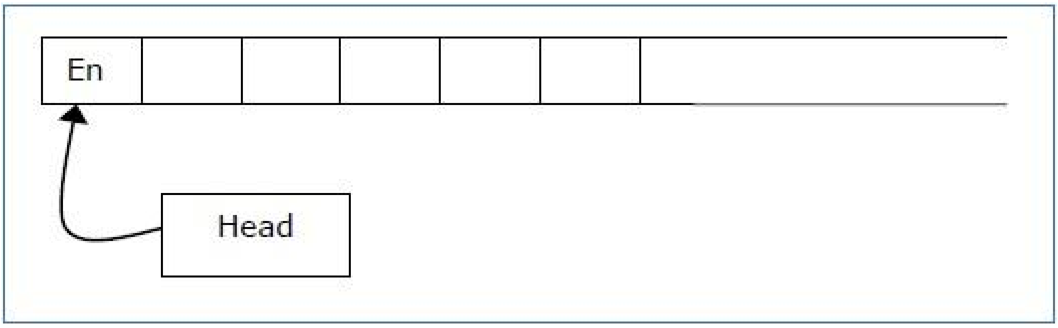

图 5 具有半无穷带的图灵机[15]

具有半无穷带的图灵机指的是纸带的一个方向是有限的[2]。其同样满足与图灵机的等效性。只需证明具有半无穷带的图灵机可以模拟图灵机即可。模拟的思路是使用具有两个道的半无穷带图灵机，其中上道记录起始位置以及右侧的全体字符，下道记录左侧字符，同时标记起始位置。这样，再通过记录此时正在读写上道或下道，即可以将半无穷带当作双向无穷的带使用[2]。

$$M=(Q\times\{U,D\},\Sigma\times\{B\},\Gamma\times\Gamma,\delta',<B,B>,F\times\{U,D\})$$

考察以上图灵机的变体可以发现，对图灵机功能进行纸带或状态转移的扩展和进行纸带的限制不会使图灵机的识别范围扩大或缩小。进一步研究表明，同样与原始图灵机等价的模型还包括多头图灵机、多维图灵机、多堆栈机器、计数器机器[2]等。由此看来，图灵机是一个通用的计算模型，可用于拓展出新的模型，也常可以用于检验新提出的一些计算模型识别语言的范围；同时具有较强的计算识别能力，新模型的计算能力往往不能超过它，这也符合学界普遍认可的丘奇-图灵论题：任何自然的称作有效过程的处理都可以由图灵机实现[2]。

3.2 图灵机的时间复杂度和带复杂度问题

可以看出，虽然识别各种语言的自动机和图灵机的各种等价变体被提出，但它们解答的仍然是可识别、可判定、可计算的问题，对于实现的现实可能性，尤其是计算的复杂程度没有进行进一步分析。随着数字计算机的出现，降低计算的复杂性的问题

提上日程；利用图灵机模型从理论上定义计算的复杂性，探索更高效的计算方法成为研究的课题。

3.2.1 定义在多带图灵机上的时间复杂度

J. Hartmanis 和 R. E. Stearns 定义和研究了多带图灵机上的时间复杂度问题（1964）。对于序列α，如果存在一个多带图灵机在$T(n)$步中能够计算出α的第 n 项，则称α在复杂度类S_T中或α是T-可计算的[16]。他们证明了 13 条有关多带图灵机时间复杂度的定理和相应的推论，例如S_T递归可枚举，$S_T = S_{kT}$，$S_{T_1} \cap S_{T_2} = S_{MIN(T_1,T_2)}$，$\inf\limits_{n\to\infty} \frac{T(n)^2}{U(n)} = 0 \to S_T$真包含于$S_U$[16]。

3.2.2 定义在离线图灵机上的带复杂度

他们紧接着又定义与研究了图灵机的带复杂度问题（1965）。他们对分离输入带和工作带的离线图灵机（即输入带不能写入的双带图灵机）进行考察，定义对长度为 n 的输入，工作带上访问的最多单元数为图灵机的带复杂度 L(n)，全体可以在 L(n)复杂度被下离线图灵机识别的语言构成复杂度类C_L[17]。同样证明了若干定理和推论，比如

$$C_{L(n)}=C_{[L(n)/N]}$$ [17]

C_L包含非正则集合，$\sup\limits_{n\to\infty} \frac{L(n)}{\log\log n} = 0$ [17]

他们也对禁止输入带向右移动的特殊离线图灵机——在线图灵机，以及离线/在线的下推自动机进行研究，分别得到它们识别非正则语言集合的带复杂度。

上述定义在图灵机变形上的复杂度给今天计算理论领域的复杂度理论奠定了重要基础。两位作者因此在 1993 年获得了图灵奖。

3.2.3 复杂度问题在非确定图灵机上的研究

J. E.Hopcroft 和 J. D. Ullman 在非确定性图灵机中研究了空间复杂度，拓展了他们的结果（1969）[18]；研究了非确定图灵机时间空间复杂度之间的关系(1968)，指出如果一个语言 L 在时间复杂度 T(n)内被非确定单带图灵机识别，那么 L 就能被一个带复杂度为$T(n)^{1/2}$的非确定离线图灵机所识别；其中如果 $T(n) \geq n^2$，L 被带复杂度为$T(n)^{1/2}$的非确定单带图灵机识别。如果语言 L 在时间复杂度 T(n)内被非确定离线图灵机识别，则 L 被带复杂度为$(T(n)logn)^{1/2}$的非确定性离线图灵机识别[19]。

Walter J. Savitch 进一步研究了确定和非确定图灵机的带复杂度的关系（1970），证明了 Savitch 定理：如果 L 具有能在带复杂度 L(n)内被非确定图灵机识别，且 $T(n) \geq \log_2 n$，则能在带复杂度$L(n)^2$内被确定图灵机识别[20]。

4　图灵机模型的研究现状和未来方向

4.1 计算系统图灵完备性研究

计算机领域不断涌现出新的计算系统，例如编程语言、编译器乃至大模型和新型计算机，研究者通常期望它们拥有的计算能力达到图灵机水平。如果某个计算系统或编程语言具有和通用图灵机等效的计算能力，则称为图灵完备的。由于计算系统不断出现，因此需要与之匹配的研究证明或否定其图灵完备性，从而揭示计算系统的计算能力；为了便于验证新兴计算系统的图灵完备性，也可以从理论上构建与图灵机等价的图灵机变形，从而直观地构建新系统的示图。

例如，英国物理学家德意奇提出了等价于量子电路的量子图灵机[21]，启发了量子计算机的研发；Jorge P´erez 等人关注模型 Transformer，证明了该模型的图灵完备性[22]。

可以看出，基于图灵机的自动机研究与现实机器的发展相关。故可以猜想未来可能出现针对神经网络、生成式模型的图灵机进一步建模和研究，包括基于此对神经网络计算能力的揭示。

4.2 依赖图灵机变形的计算复杂性研究

当前，图灵机作为计算模型，其各种变体常用于定义和研究计算复杂性。例如，

在证明某些问题的计算复杂性下界时，研究者通常构造一个特定的图灵机模型，并证明任何试图解决该问题的算法都无法在该模型下以低于某个复杂度的方式运行。这种方法提供了评估问题难度的有力工具。

由于图灵机模型假设计算机的内存是无限的，而现实中的计算机内存是有限的。因此，在研究现实世界的计算问题时，研究者通常考察 RAM 机，纸带有限的离线图灵机模型等图灵机变形。

将来的研究者仍会沿着前人留下的开放问题，进一步强化已证明的数学结果。

结论

图灵机模型构建了一个抽象的计算设备，模拟和实现了任何可计算的过程，从而奠定了现代计算机科学发展的理论基础。图灵机的提出，不仅标志着计算理论的诞生，更为后续计算机设计、编程语言研究等提供了重要启示。

本文回顾了图灵机模型的提出背景，探讨了希尔伯特的可判定性问题，并对图灵机的物理结构、七元组结构、格局、可识别、可判定等基本概念和所解答的 Entschei dungsproblem 问题予以简要介绍。在图灵机的基础上，研究者们提出了多种变形模型，本文简要介绍了多带图灵机、非确定性图灵机和具有半无穷带的图灵机，并探讨论证其与图灵机等价性的思路。本文还介绍了 J. Hartmanis 和 R. E. Stearns 研究的多带图灵机时间复杂度、离线图灵机带复杂度定义和基本结论。

基于上述讨论，本文尝试分析了图灵机模型未来的潜在发展方向，认为图灵完备性的相关研究始终具有活力，依赖图灵机变形的计算复杂性问题中待解决的开放数学问题会将会被推进。

参考文献

[1] A. M. Turing. *On Computable numbers, with an application to the Entscheidungsproblem*[J]. Proceedings of the London Math. Society, 1936, 58(5): 345-363.

[2] John E. Hopcroft, Rajeev Motwani, Jeffrey Ullman. 自动机理论、语言和计算导论[M]. 孙家骕等，译. 北京：机械工业出版社：2008：215-254.

[3] Wikipedia. *Hilbert's programme*[DB/OL]https://en.wikipedia.org/wiki/G%C3%B6del%27s_completeness_theorem, 2022.10.9.

[4] Wikipedia. *Gödel's completeness theorems*[DB/OL]https://en.wikipedia.org/wiki/Hilbert%27s_program, 2014.9.22.

[5] Stanford Encyclopedia of Philosophy. *Gödel's incompleteness theorems*[DB/OL]https://plato.stanford.edu/entries/goedel-incompleteness/. 2020.4.2.

[6] Gödel, Kurt. *Die vollständigkeit der axiome des logischen funktionenkalküls.* Monatshefte für Mathematik und Physik, 1930, 37: 349-360.

[7] Gödel, Kurt. *Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I*[J]. Monatshefte für Mathematik und Physik, 1931, 38: 173-198.

[8] A. Church. *An undecidable problem in elementary number theory*[J]. American J. Math. 1936, 58: 345-363.

[9] E. Post. *Finite combinatory process-formulation*[J]. J. Symbolic Logic. 1936, 1: 103-105.

[10] S. C. Kleene. *General recursive functions of natural numbers*[J]. Mathematische Annalen. 1936, 112: 727-742.

[11] Hilbert, Ackermann. *Grundzige der Theoretischen Layik*[M]. Berlin, 1931: Chapter 3.

[12] Davis, Martin. *Computability & unsolvability*[M]. New York: McGraw-Hill book company, 1982: 70.

[13] Wikipedia. *Turing Machine*[DB/OL]https://en.wikipedia.org/wiki/Turing_machine, 2019.6.25.

[14] Tutorialspoint. *Multi-tape Turing Machine*[DB/OL]. https://www.tutorialspoint.com/automata_theory/multi_tape_turing_machine.htm.

[15] Tutorialspoint. *Semi infinite-tape Turing Machine*[DB/OL]. https://www.tutorialspoint.com/automata_theory/semi_infinite_tape_turing_machine.htm.

[16] Hartmanis, Juris, and Richard E. Stearns. *On the computational complexity of algorithms*[J]. Transactions of the American Mathematical Society. 1965, 117: 285-306.

[17] Stearns, Richard Edwin, Juris Hartmanis, and Philip M. Lewis. *Hierarchies of memory limited computations*[C]. 6th annual symposium on switching circuit theory and logical design (SWCT 1965), IEEE, 1965: 179-190.

[18] Hopcroft, John E., and Jeffrey D. Ullman. *Some results on tape-bounded Turing machines*[J]. Journal of the ACM (JACM). 1969. 16(1): 168-177.

[19] Hopcroft, John E., and Jeffrey D. Ullman. *Relations between time and tape complexities*[J]. Journal of the ACM (JACM). 1968, 15(3): 414-427.

[20] Savitch, Walter J. *Relationships between nondeterministic and deterministic tape complexities*[J]. Journal of computer and system sciences. 1970. 4(2) : 177-192.

[21] Deutsch, David. *Quantum theory, the Church–Turing principle and the universal quantum computer*[J]. Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences. 1985, 400(1818): 97-117.

[22] Pérez, Jorge, Pablo Barceló, and Javier Marinkovic. *Attention is turing-complete*[J]. Journal of Machine Learning Research. 2021, 22(75) : 1-35.