组合优化

浙江大学数学系　谈之奕

计算复杂性初步

NP 类

- 基于非确定性算法的NP 类定义
  - 对一判定问题，若存在一非确定性算法，使得对任何一个答案为“是”的实例，该算法能
    - 猜想出该实例的一个可行解
    - 该可行解规模不超过实例规模的多项式
    - 能在实例规模的多项式时间内验证猜想是否正确

  则称该问题属于NP 类

非确定性算法只是为研究而定义的一种理论算法模型，在现实生活中并不存在

$\mathcal{NP}$ 类

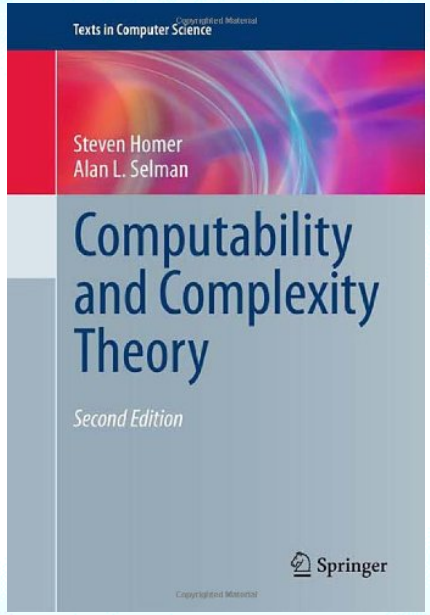
- **TSP判定形式** $\in \mathcal{NP}$
  - 猜想出实例的一个可行解 π
  - π 可按经过城市的标号顺序用 n 个数 $i_1, i_2, \cdots, i_n$ 表示，可行解规模为 $n\log_2 n$，不超过实例规模 $n+\log_2 L$ 的某个多项式函数
  - 用 n 次加法即可验证 $\sum_{i=1}^{n-1} c_{\pi(i)\pi(i+1)} + c_{\pi(n)\pi(1)} \le M$ 成立

给定城市间距离 c_{ij} 和整数阈值 M，问是否存在排列 π，使得环游长不超过 M

$\mathcal{NP}$ 类

- 不在$\mathcal{NP}$ 类中的问题
  - 不可判定问题（**Undecidable Problem**）
  - 非判定问题
  - 由时间分层定理（**time hierarchy theorem**）所给出的某些$\mathcal{NEXP}$ 类中的问题
- $\mathcal{NP}$ 类的结构
  - $\mathcal{P}$ 类中问题答案为“是”的实例对应的解可直接求出，故 $\mathcal{P} \subseteq \mathcal{NP}$

$\mathcal{NP}$ 类中最难的问题？

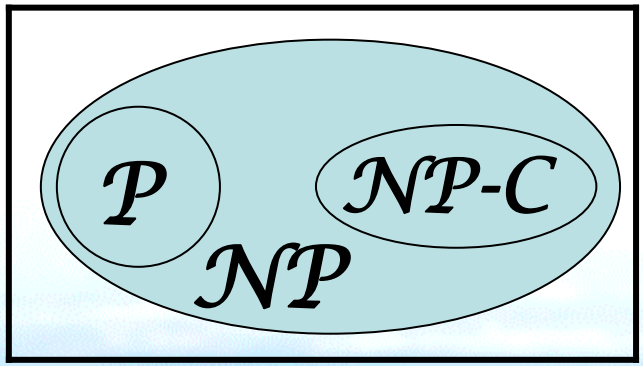


Homer S, Selman AL. *Computability and Complexity Theory*, Springer, 2011.

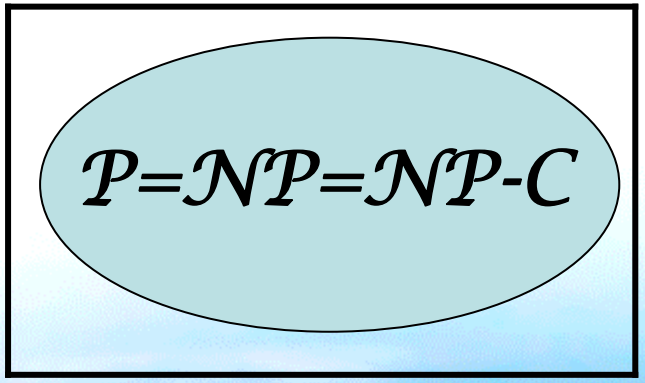
NP-完全问题

组合优化

- NP 类中最“难”的问题子集称为NP －完全类，记为NP-C。NP-C 类中的问题称为NP －完全问题（NP -complete problem）
  - 若NP-C 类中有一个问题有多项式时间算法，则NP 类中所有问题都有多项式时间算法



$P \neq NP\ (P \cup NP\text{-}C \neq NP)$



$P = NP$

P，NP，与NP—完全

浙江大学 ZheJiang University

组合优化

TEACHER'S BOOK 普通高级中学实验教科书 数学3（必修） 教师教学用书

4. 计算的复杂性

计算的复杂性测度函数有三类：一类是指数型的，常写为 c^n 的形式（c 是常数）；的，常写为 n^k 的形式（k 为非负整数）；另一类是对数型的，常写为 $\log n$ 的形式. 问题分别称为指数复杂性，多项式复杂性和对数复杂性.

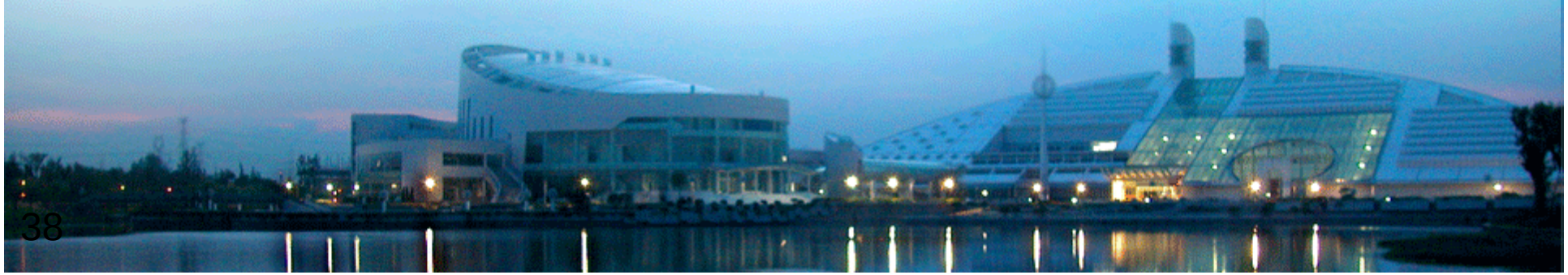
人们习惯于把理论上可计算的问题类称为能行可计算的，而把具有多项式复杂性效可计算的，通常称为 P 问题. NP 问题是指还未找到多项式复杂性算法的问题.

研究和实验表明，单纯靠提高计算机速度并不能解决 NP 问题. 例如，根据某些录，对于复杂性为 2^n 的问题，即使计算机速度提高 1 000 倍，也只能是多算约 10 道题为，解决 NP 问题的关键是要从数学上找出好的算法. 事实上，数学家们也找到了各种各样的好办法，大大简化了计算. 例如，用计算机对卫星照片进行处理，如果在一张 10 cm^2 的照片上以一微米为间隙打上格子，则处理一张照片需要进行 10^{16} 次运算，即使用每秒百亿次的计算机也要连续算上十多个昼夜.

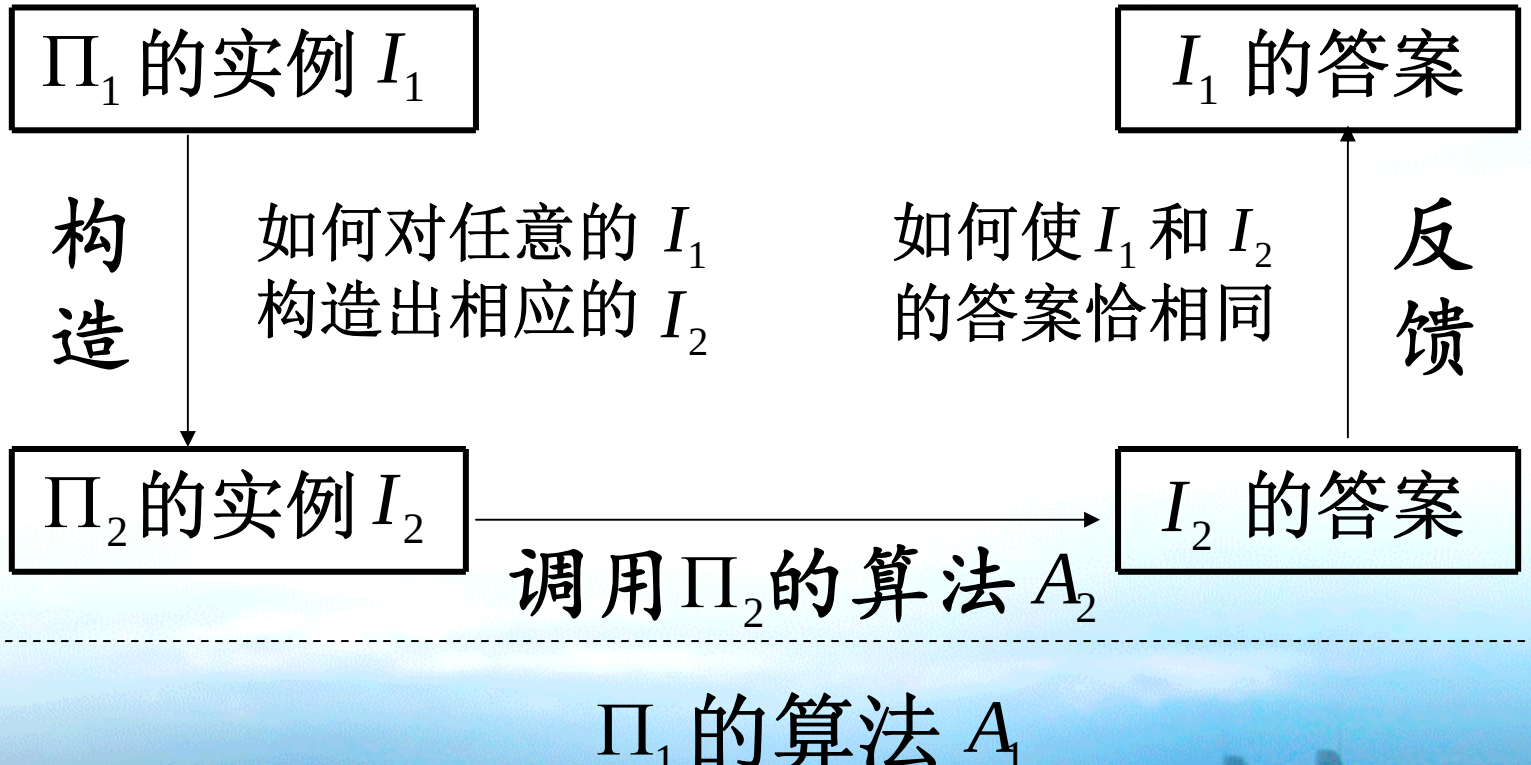
后来，有人发明了一种好的算法，大大降低了计算的复杂性，使得用同样的计算机计算只需要 $\frac{1}{3}$ 秒.

归约

- 设有判定问题 Π_1, Π_2，若对 Π_1 的任一实例 I_1，可在多项式时间内构造出 Π_2 的一个实例 I_2，使得 I_1 的答案为“是”当且仅当 I_2 的答案为“是”，则称 Π_1 可多项式时间归约到 Π_2，记为 $\Pi_1 \leq_m^p \Pi_2$
- 若 $\Pi_1 \leq_m^p \Pi_2$，则 Π_2 不会比 Π_1 更容易
  - 可以用求解 Π_2 的多项式时间算法设计出求解 Π_1 的多项式时间算法，但反之未必成立
- 归约作为两问题间的一种关系具有传递性



归约


Π_1 的实例 I_1
I_1 的答案
构造
如何对任意的 I_1
构造出相应的 I_2
如何使 I_1 和 I_2
的答案恰相同
反馈
Π_2 的实例 I_2
I_2 的答案
调用 Π_2 的算法 A_2
Π_1 的算法 A_1

Hamiltion圈

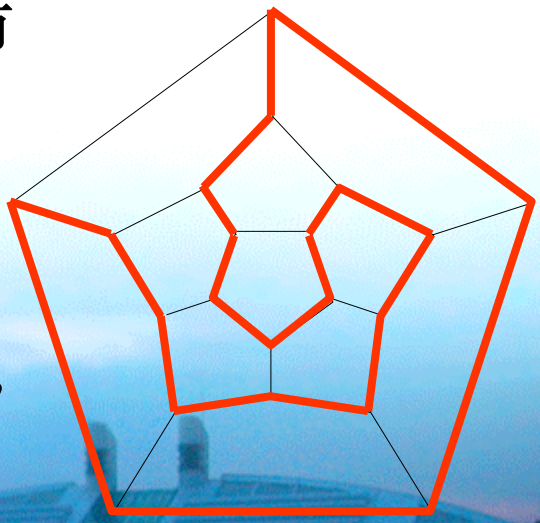
- 经过图的所有顶点恰好一次的圈称为 **Hamilton圈**（**Hamiltion cycle**）。存在**Hamilton**圈的图称为**Hamilton图**
- **Hamilton图问题（HC）**：判断图 G 是否为一**Hamilton**图

Hamilton图问题是图论中最重要的问题之一。图论中有很多判别**Hamilton**图的充分/必要条件和对不同类型特殊图是否为**Hamilton**图的讨论。在计算复杂性理论中重点关注**Hamilton**图判别算法的复杂性

William Rowan Hamilton
爱尔兰数学家
(1805-1865)

周游世界

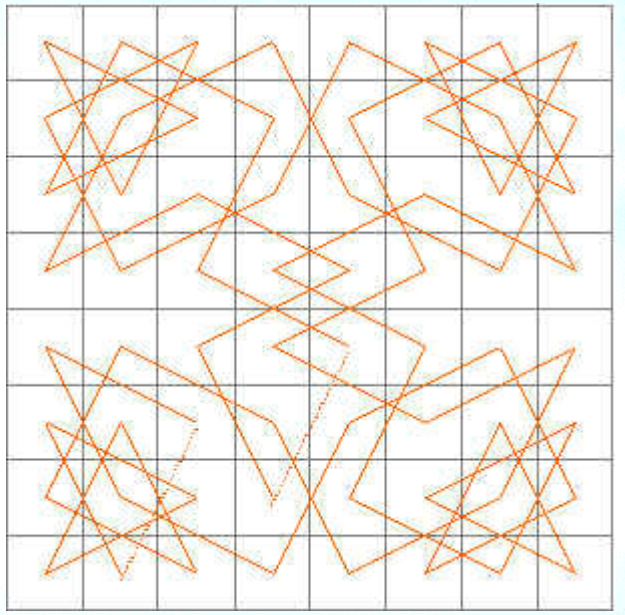
- **1859年Hamilton发明了周游世界的游戏icosian game**
  - 一个正十二面体的二十个顶点各代表一个城市，是否有一条从某个城市出发，沿正十二面体的棱行走，经过每个城市恰好一次，最后回到出发城市的路线

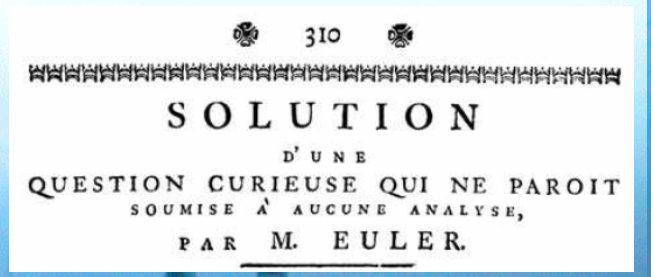


Amsterdam, Ann Arbor, Berlin, Budapest, Dublin, Edinburgh, Jerusalem, London, Melbourne, Moscow, Novosibirsk, New York, Paris, Peking, Prague, Rio di Janeiro, Rome, San Francisco, Tokyo, Warsaw

Knight's tour

- 在8×8国际象棋棋盘上，马能否按其走子规则，从一个格子出发，经过其它格子恰好一次，最后回到起点
  - 构造“跳马图”，每一格子为图的一个顶点，两个格子之间有边相连当且仅当马可按走子规则从一个格子跳到另一个格子
- $m \times n\ (m \le n)$ 方格棋盘对应的“跳马图”为**Hamiltonian**图，除非
  - m, n 均为奇数
  - $m = 1, 2, 4$
  - $m = 3, n = 4, 6, 8$





310

SOLUTION

D'UNE

QUESTION CURIEUSE QUI NE PAROIT

SOUMISE À AUCUNE ANALYSE,

PAR M. EULER.

Euler L, Solution of a curious question which does not seem to have been subjected to any analysis, *Mémoires de l'Academie Royale des Sciences et Belles Lettres*, 15, 310–337, 1759

Schwenk A J. Which rectangular chessboards have a knight's tour? *Mathematics Magazine*, 64, 325-332, 1991.

$HC \leq_m^p TSP$

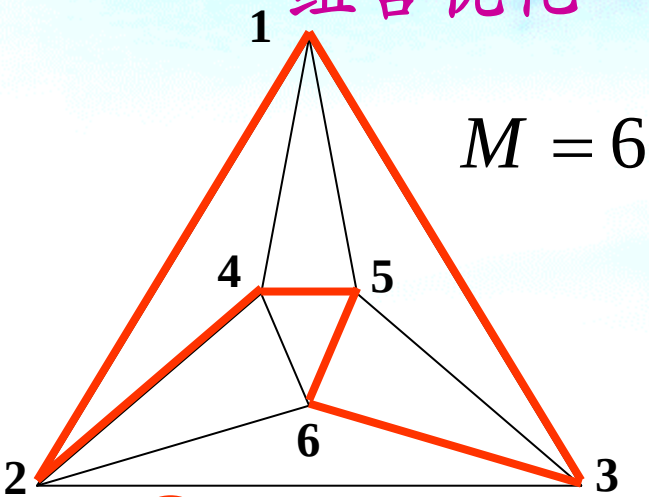
- 任给**HC**问题的实例 I_{HC}：图 $G=(V,E)$
- 构造**TSP**判定形式的实例 I_{TSP}
  - 城市数 $n=|V|$
  - 城市间距离 $c_{ij}=\begin{cases}1, \text{若 } (v_i,v_j)\in E,\\ 2, \text{若 } (v_i,v_j)\notin E,\end{cases} \quad i\neq j$
  - 整数阈值 $M=|V|$

图的顶点与城市一一对应。若两个顶点之间有边相连，对应城市间距离为**1**；若两个顶点之间无边相连，对应城市间距离为**2**

- I_{HC} 的答案为“是”当且仅当 I_{TSP} 的答案也为“是”
  - G 中存在一**Hamilton**圈当且仅当存在总长度不超过 $|V|$ 的环游

$HC \leq_m^p TSP$

- 若 G 中存在一**Hamilton** 圈，则按该圈经过各顶点顺序依次到达每个城市， 圈中每条边的两个端点所对应的两个城市间距离均为**1**，环游总长度恰为 $|V|$
- 若环游总长度不超过 $|V|$，由于环游需经过 $|V|$ 个城市，环游中相邻两个城市间距离均为**1**，它们所对应的 G 中两个顶点之间均有边相连，所有这些边恰组成 G 的一个**Hamilton**圈



$M = 6$

$$\begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 2 \\ 1 & 0 & 1 & 1 & 2 & 1 \\ 1 & 1 & 0 & 2 & 1 & 1 \\ 1 & 1 & 2 & 0 & 1 & 1 \\ 1 & 2 & 1 & 1 & 0 & 1 \\ 2 & 1 & 1 & 1 & 1 & 0 \end{pmatrix}$$

$\mathcal{NP}$-完全问题

- 若$\Pi \in \mathcal{NP}$，且对任意的 $\Pi' \in \mathcal{NP}, \Pi' \leq_m^p \Pi$，则称 Π 是$\mathcal{NP}$ －完全问题（$\mathcal{NP}$- **complete problem**）
- 所有$\mathcal{NP}$ －完全问题的集合称为$\mathcal{NP}$ －完全类，记为$\mathcal{NP}$-C
  - $\mathcal{NP}$-C 类是$\mathcal{NP}$类的一个子类，包含了$\mathcal{NP}$ 类中最“难”的问题
  - 若$\mathcal{NP}$-C 类中有一个问题有多项式时间算法，则$\mathcal{NP}$ 类中所有问题都有多项式时间算法

从定义出发证明一问题的$\mathcal{NP}$ －完全性是困难的

$\mathcal{NP}$ —完全性判定定理

浙江大学 ZheJiang University

组合优化

- $\mathcal{NP}$ —完全性判定定理
  - 若 $\Pi \in \mathcal{NP}$，且存在某个 $\Pi^C \in \mathcal{NP}\text{-}C$，$\Pi^C \leq_m^p \Pi$，则 $\Pi \in \mathcal{NP}\text{-}C$
    - 任取 $\Pi' \in \mathcal{NP}$，由于 $\Pi^C \in \mathcal{NP}\text{-}C$，故由$\mathcal{NP}$ —完全问题的定义，$\Pi' \leq_m^p \Pi^C$
    - 由定理条件和归约的传递性 $\Pi' \leq_m^p \Pi$
    - 由Π'的任意性和$\mathcal{NP}$ —完全问题的定义，$\Pi \in \mathcal{NP}\text{-}C$

第一个$\mathcal{NP}$—完全问题

数理逻辑

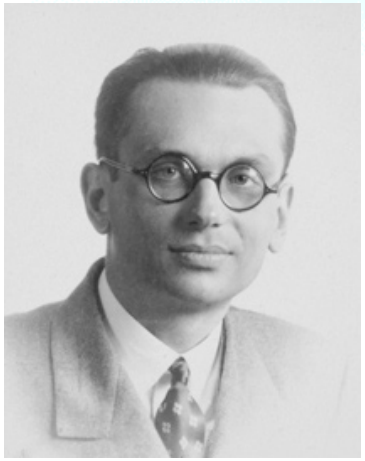
- 数理逻辑（**mathematical logic**）：用数学的方法研究逻辑推理和数学计算，将推理论证、数学计算的过程符号化、形式化、公理化的学科

1956年Gödel致von Neumann信，信中对若干数理逻辑问题算法和复杂性的讨论被认为是计算复杂性研究的开端

Princeton 20./III.1956.

Lieber Herr v. Neumann!

Ich habe mit grösstem Bedauern von Ihrer Erkrankung gehört. Die Nachricht kam mir ganz unerwartet. Morgenstern hatte mir zwar schon im Sommer von einem Schwächeanfall erzählt, den Sie einmal hatten, aber er meinte damals, dass dem keine grössere Bedeutung beizumessen sei. Wie ich höre, haben Sie sich in den letzten Monaten einer radikalen Behandlung unterzogen u. ich freue mich, dass diese den gewünschten Erfolg hatte u. es Ihnen jetzt besser geht. Ich hoffe u. wünsche Ihnen, dass

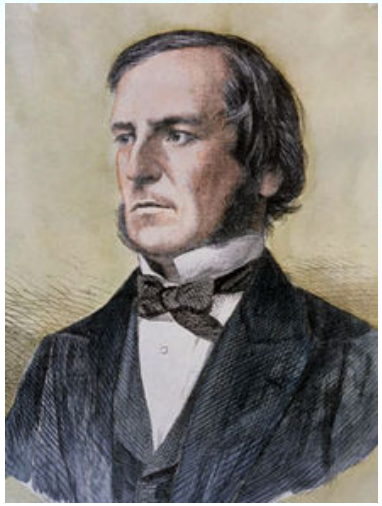


Kurt Friedrich Gödel
（1906－1978）
奥地利哲学家、数学家

John von Neumann
（1903－1957）
匈牙利裔美国科学家

Boolean变量

- 仅可取“真”和“假”（True(T)和False(F)，1和0）两种值的变量称为Boolean变量
- Boolean变量的运算
  - 非（negation）¬：$\neg x = 1 \Leftrightarrow x = 0$
  - 析取（disjunction）∨
    $x_1 \vee x_2 = 1 \Leftrightarrow x_1 = 1 或 x_2 = 1$
  - 合取（conjunction）∧
    $x_1 \wedge x_2 = 1 \Leftrightarrow x_1 = 1 且 x_2 = 1$
- Boolean变量的运算遵从双重否定律、De Morgan律、析取对合取的分配率、合取对析取的分配率等运算定律



George Boole
（1815-1864）
英国数学家、哲学家、逻辑学家

Boole家族

George Boole（1815-1864） × **Mary Everest Boole**（1832-1916） 数学教育家、科普作家

- **Mary Ellen Hinton (1856-1908)**
  - **Geoffrey Ingram Taylor (1886-1975)** 数学家、物理学家 英国皇家学会会士
  - **Julian Taylor (1889-1961)** 外科学家
- **Margaret Taylor (1858-1935)**
- **Alicia Boole Stott (1860-1940)** 数学家 四维几何的早期研究者
- **Lucy Everest Boole (1862-1904)** 化学家 伦敦Royal Free Hospital首位女性化学教授
- **Ethel Lilian Voynich (1864-1960)** 作家 代表作《牛虻》（The Gadfly）

Boole家族

浙江大学 ZheJiang University

组合优化

Mary Ellen Hinton (1856-1908) × **Charles Howard Hinton（1853-1907）** 普林斯顿大学数学教师 科幻小说家

George Boole Hinton (1882-1943)

Howard Everest Hinton (1912-1977)

昆虫学家、英国皇家学会会士

Geoffrey Everest Hinton (1947-)

计算机学家、英国皇家学会会士

2018年Turing奖得主

Eric Biorn Hinton (1883-c.1930)

William Howard Hinton (1886-1909)

Sebastian Hinton (1887-1923) × **Carmelita Hinton (1890-1983)**

教育家，1935年创办 The Putney School (帕特尼中学)

Jean Hinton Rosner (1917-2002)

记者、农学家。1945年到中国，重庆谈判期间与毛泽东、周恩来有多次会面，亲历山西土改，创作纪实文学《翻身》。1953年返美后受迫害，1971年重返中国

William H. Hinton(韩丁) (1919-2004)

Joan Chase Hinton(寒春) (1921-2010)

寒春，核物理学家，曼哈顿计划参与者，1948年到上海，后到延安。

Erwin Engst(阳早) (1919-2003)

阳早，农学家，1946年到延安。

建国后夫妇均在中国农业机械研究院工作，从事奶牛养殖技术研究

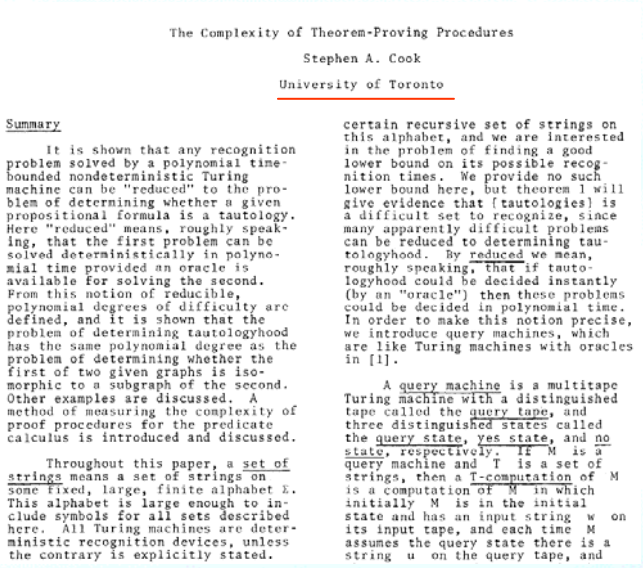
Boolean表达式

- 若干**Boolean**变量用运算符和括号按一定的逻辑关系联结起来的表达式称为**Boolean表达式**（**Boolean expression**）
- 对出现在**Boolean**表达式中的所有**Boolean**变量各指定一个值，可按**Boolean**变量的运算法则确定表达式的值**1**或**0**
- 任一**Boolean**表达式都存在与之等价的合取范式（**conjunctive normal form, CNF**）
  - 文字（**literal**）：变量或变量的非
  - 子句（**clause**）：若干个文字的析取
  - **CNF**：若干个子句的合取

$$\neg(\neg x_1 \vee x_2) \vee x_3 \Leftrightarrow (\neg\neg x_1 \wedge \neg x_2) \vee x_3 \Leftrightarrow (x_1 \wedge \neg x_2) \vee x_3$$
$$\Leftrightarrow (x_1 \vee x_3) \wedge (\neg x_2 \vee x_3)$$

SAT

- 可满足性问题（**Satisfiability，SAT**）
  - 给定一合取范式，问是否存在其变量的一种赋值，使得该表达式值为真
    - $(x_1 \vee \neg x_2) \wedge \neg x_2$：$x_1$取**1**，$x_2$取**0** 可使表达式值为**1**
    - $(x_1 \vee x_2) \wedge (x_1 \vee \neg x_2) \wedge \neg x_1$：不论 x_1, x_2 取值为何值，表达式值均为 **0**
  - 猜想所有变量的一种赋值，在多项式时间内可验证表达式值确为**1**，故 $\mathrm{SAT} \in \mathcal{NP}$
- **1971**年，**Cook**运用图灵机语言，通过$\mathcal{NP}$问题的一种等价定义，用$\mathcal{NP}$－完全问题的定义证明了**SAT**问题的$\mathcal{NP}$－完全性。**SAT**问题被认为是第一个$\mathcal{NP}$－完全问题



The Complexity of Theorem-Proving Procedures

Stephen A. Cook

University of Toronto

Summary

It is shown that any recognition problem solved by a polynomial time-bounded nondeterministic Turing machine can be "reduced" to the problem of determining whether a given propositional formula is a tautology. Here "reduced" means, roughly speaking, that the first problem can be solved deterministically in polynomial time provided an oracle is available for solving the second. From this notion of reducible, polynomial degrees of difficulty are defined, and it is shown that the problem of determining tautologyhood has the same polynomial degree as the problem of determining whether the first of two given graphs is isomorphic to a subgraph of the second. Other examples are discussed. A method of measuring the complexity of proof procedures for the predicate calculus is introduced and discussed.

Throughout this paper, a set of strings means a set of strings on some fixed, large, finite alphabet Σ. This alphabet is large enough to include symbols for all sets described here. All Turing machines are deterministic recognition devices, unless the contrary is explicitly stated.

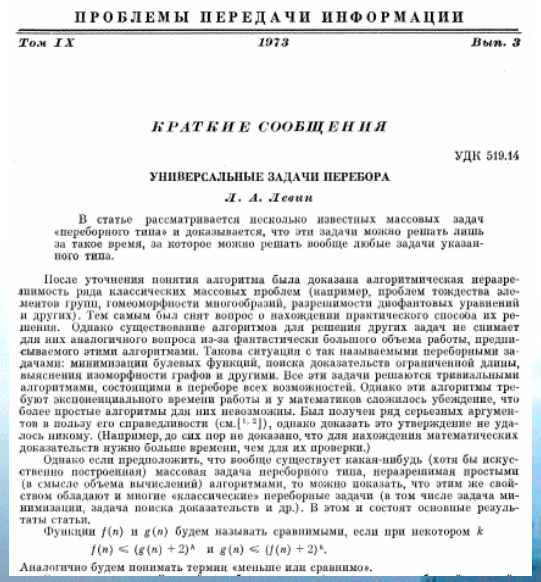
certain recursive set of strings on this alphabet, and we are interested in the problem of finding a good lower bound on its possible recognition times. We provide no such lower bound here, but theorem 1 will give evidence that {tautologies} is a difficult set to recognize, since many apparently difficult problems can be reduced to determining tautologyhood. By reduced we mean, roughly speaking, that if tautologyhood could be decided instantly (by an "oracle") then these problems could be decided in polynomial time. In order to make this notion precise, we introduce query machines, which are like Turing machines with oracles in [1].

A query machine is a multitape Turing machine with a distinguished tape called the query tape, and three distinguished states called the query state, yes state, and no state, respectively. If M is a query machine and T is a set of strings, then a T-computation of M is a computation of M in which initially M is in the initial state and has an input string w on its input tape, and each time M assumes the query state there is a string u on the query tape, and

Cook SA. The complexity of theorem-proving procedures, *Proceedings of the 3rd annual ACM Symposium on Theory of Computing*, 151-158, 1971.

Cook-Levin定理

- 与**Cook**同时，**Levin**独立地给出了若干$\mathcal{NP}$－完全问题，$\mathrm{SAT} \in \mathcal{NP}\text{-}C$
  因此被称作**Cook-Levin**定理



ПРОБЛЕМЫ ПЕРЕДАЧИ ИНФОРМАЦИИ

Том IX *1973* *Вып. 3*

КРАТКИЕ СООБЩЕНИЯ

УДК 519.14

УНИВЕРСАЛЬНЫЕ ЗАДАЧИ ПЕРЕБОРА

Л. А. Левин

В статье рассматривается несколько известных массовых задач «переборного типа» и доказывается, что эти задачи можно решать лишь за такое время, за которое можно решать вообще любые задачи указанного типа.

После уточнения понятия алгоритма была доказана алгоритмическая неразрешимость ряда классических массовых проблем (например, проблем тождества элементов групп, гомеоморфности многообразий, разрешимости диофантовых уравнений и других). Тем самым был снят вопрос о нахождении практического способа их решения. Однако существование алгоритмов для решения других задач не снимает для них аналогичного вопроса из-за фантастически большого объема работы, предписываемого этими алгоритмами. Такова ситуация с так называемыми переборными задачами: минимизации булевых функций, поиска доказательств ограниченной длины, выяснения изоморфности графов и другими. Все эти задачи решаются тривиальными алгоритмами, состоящими в переборе всех возможностей. Однако эти алгоритмы требуют экспоненциального времени работы и у математиков сложилось убеждение, что более простые алгоритмы для них невозможны. Был получен ряд серьезных аргументов в пользу его справедливости (см. [1-2]), однако доказать это утверждение не удалось никому. (Например, до сих пор не доказано, что для нахождения математических доказательств нужно больше времени, чем для их проверки.)

Однако если предположить, что вообще существует какая-нибудь (хотя бы искусственно построенная) массовая задача переборного типа, неразрешимая простыми (в смысле объема вычислений) алгоритмами, то можно доказать, что этим же свойством обладают и многие «классические» переборные задачи (в том числе задача минимизации, задача поиска доказательств и др.). В этом и состоят основные результаты статьи.

Функции $f(n)$ и $g(n)$ будем называть сравнимыми, если при некотором k

$$f(n) \leqslant (g(n)+2)^k \quad \text{и} \quad g(n) \leqslant (f(n)+2)^k.$$

Аналогично будем понимать термин «меньше или сравнимо».

Levin LA, Universal Sequential Search Problems, *Problems of Information Transmission*, 9, 115–116, 1973

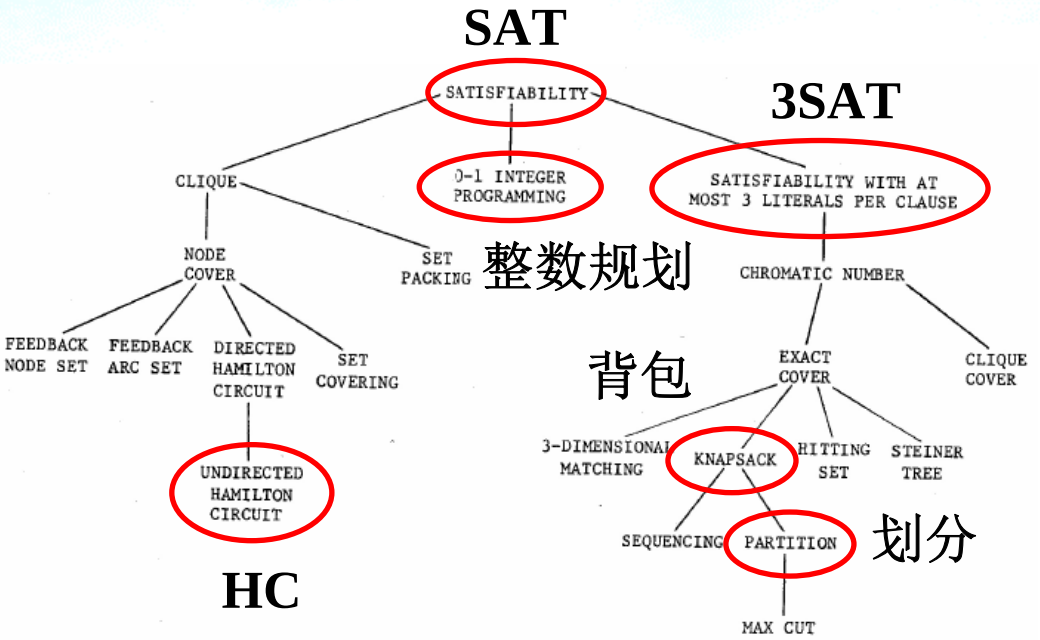
Trakhtenbrot BA, A survey of Russian approaches to perebor (brute-force searches) algorithms. *Annals of the History of Computing*, 6, 384-400, 1984

Leonid Anatolievich Levin
（1948－ ）
苏联计算机学家

Karp归约

浙江大学 ZheJiang University

组合优化



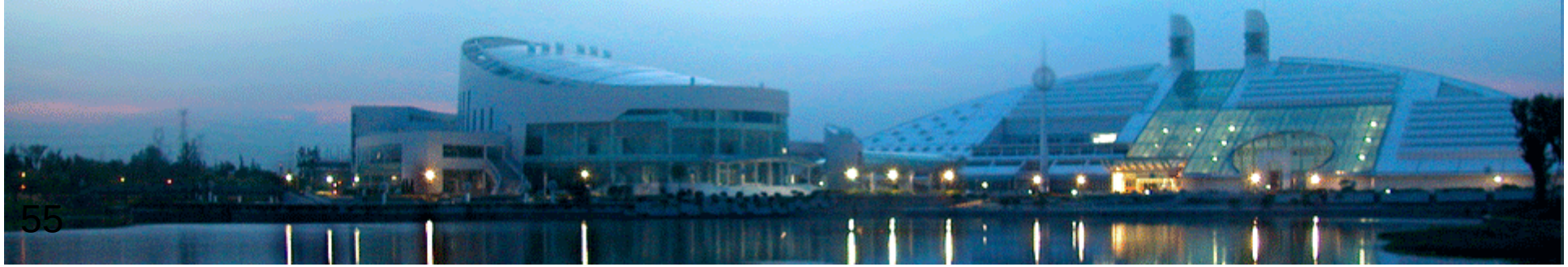
REDUCIBILITY AMONG COMBINATORIAL PROBLEMS†

Richard M. Karp

University of California at Berkeley

Abstract: A large class of computational problems involve the determination of properties of graphs, digraphs, integers, arrays of integers, finite families of finite sets, boolean formulas and elements of other countable domains. Through simple encodings from such domains into the set of words over a finite alphabet these problems can be converted into language recognition problems, and we can inquire into their computational complexity. It is reasonable to consider such a problem satisfactorily solved when an algorithm for its solution is found which terminates within a number of steps bounded by a polynomial in the length of the input. We show that a large number of classic unsolved problems of covering, matching, packing, routing, assignment and sequencing are equivalent, in the sense that either each of them possesses a polynomial-bounded algorithm or none of them does.

Karp RM. Reducibility Among Combinatorial Problems, *Proceedings of a Symposium on the Complexity of Computer Computations*, 85-103, 1972



$\mathcal{NP}$ －完全性证明

浙江大学
ZheJiang University

组合优化

- 问题 Π 的$\mathcal{NP}$ －完全性证明
  - 证明 $\Pi \in \mathcal{NP}$
  - 寻找与 Π 联系紧密，且实例结构较为规范、简明的已知$\mathcal{NP}$ －完全问题 Π^C
  - 证明 $\Pi^C \leq_m^p \Pi$
    - 任取 Π^C的实例 I_1，构造 Π 的实例 I_2
    - 若 I_1 的答案为“是”，I_2 的答案也为“是”
    - 若 I_2 的答案为“是”，I_1 的答案也为“是”
      （或：若 I_1 的答案为“否”，I_2 的答案也为“否”）


浙江大学
ZheJiang University
谢　谢