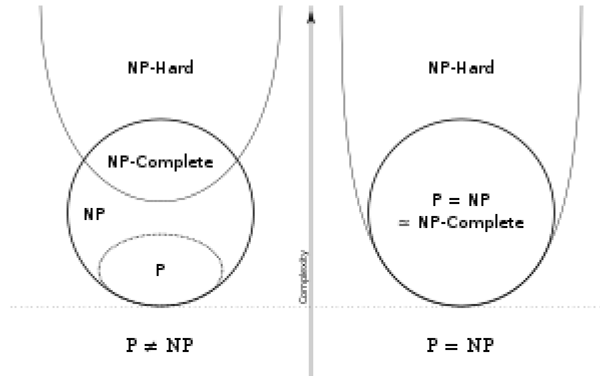
維基百科

NP (复杂度)

维基百科，自由的百科全书

非决定性多项式集合（英语：**non-deterministic polynomial**，缩写：**NP**）是计算理论中最重要的集合之一。它包含P和NP-complete。 P集合的问题即在多项式时间内可以找出解的决策性问题(decision problem)集合。注意NP包含P和NP-complete问题, 因此NP集合中有简单的问题和不容易快速得到解的难题。[NP等不等于P?]是一个计算机科学中知名的难题。



Euler diagram for P, NP, NP-complete, and NP-hard set of problems. Under the assumption that P≠NP, the existence of problems within NP but outside both **P** and NP-complete was established by Ladner.[1]

目录

定义与推论

NP, NP-hard, NP-complete的定义及推论

决策问题:一个决策问题(decision problem)是指其输出，只有“是”或“否”的问题。例如，搜索问题为询问 x 是否出现在一个集合 A 中?若有则输出“是”，否则输出“否”。
P集合: 当一个决策问题存在一个$O(n^k)$时间复杂度的算法时，则称此问题落在P 的集合中。

有一些决策问题，人类目前尚无法将他们归入集合 P 中。为了思考这些问题，于是在一般算法可采用的功能上，扩增以下虚构的新指令。这些新指令虽然不存在于现实中，但是对探讨这些难题的性质及彼此的关系，有很大的帮助。以下是这些虚构的新指令：

1. choice(S)：自集合 S 中，选出会导致正确解的一个元素。当集合 S 中无此元素时，则可任意选择一个元素。

2. failure()：代表失败结束。

3. success()：代表成功结束。
其中 choice(S)可以解释成，在求解的过程中，神奇地猜中集合 S 中其中一个元素，使其结果是成功的；并且这三个指令只需要 O(1)时间来运行。当然，choice(S) 是如何快速猜中的，在此是不需讨论的，因为

毕竟它只是虚构的。在添加这些虚构功能后，所设计出的算法，被称为非决定性算法(non-deterministic algorithm)；相较之下，原来一般的算法，就称为决定性算法(deterministic algorithm)。利用非决定性算法，我们定义出另一个集合 NP：

NP: 当一个决策问题存在一个$O(n^k)$时间复杂度的算法时，则称此问题落在NP 的集合中。

满足问题 (satisfiability problem，简称 SAT)，就是一个NP中的典型难题。

满足问题:令 x_1，x_2，…，x_n 代表布尔变量(boolean variables)(其值非真(true)即假(false)的变量)。令$-x_i$ 代表 x_i 的相反数(negation)。一个布尔公式是将一些布尔变量及其相反数利用而且(and)和或(or)所组成的表达式。满足问题是判断是否存在一种指定每个布尔变量真假值的方式，使得一个布尔公式为真。

输入:一个 n 个变量的布尔公式

例如: $(-x_1 \vee -x_2 \vee x_3) \wedge (x_1 \vee x_4) \wedge (x_2 \vee -x_1)$

输出:是否存在一种指定每个布尔变量真假值的方式，使得此公式为真？ 例如: 是(当 x_1=真，x_2=真，x_3=真，x_4=真时，此公式为真)

利用满足问题可以定义出NP-hard和NP-complete。但是我们需要一个问题转换的概念。 问题转换技巧，其所需要转换的时间皆需在多项式时间(即 $O(n^k)$)内完成。利用此多项式时间的转换，我们可以将 NP中的难题创建起一些有趣的关系。

问题转换:针对两个问题 A 和 B ，如果存在一个 $O(n^k)$时间的(决定性)算法，将每一个问题 A 的输入转换成问题 B 的输入，使得问题 A 有解时，若且惟若，问题 B 有解。此关系被称为，问题 A 转换成(reduce to)问题 B ，可表示成 $A \propto B$ 。

一个问题 L 被称为是 NP-hard，若且惟若，满足问题转换成 L(即满足问题$\propto$L)。 满足问题是 NP 中的难题，而 NP-hard 的问题则是满足问题派生(转换)出来的。

一个问题 L 被称为是 NP-complete，若且惟若，$L \in NP$ 而且 $L \in$NP-hard。

史蒂芬库克(Stephen Cook)证明了一个十分重要的性质：

性质(A)：“任一个 NP 内的问题都可以，在多项式时间内，被转换成满足问题。”

性质(B)：“任一个 NP 内的问题都可以，在多项式时间内，被转换成任一个 NP-complete 问题。”

性质(C)：“任一个 NP 内的问题都可以，在多项式时间内，被转换成任一个 NP-hard 问题。”

性质(D)：“满足问题在集合 P 中，当且仅当，P=NP。”

例子

比如说，一个决策性问题:输入一个整数x, 请回答x是否为偶数(even number)。我们利用一个程序判断x除以2是否整除即可得到最后结果 。此程序是决定性算法, 并且其时间复杂度为$O(1)=O(n^0)$, 因此此问题落入P集合中。

再举一个例子，下面是满足问题的一个非决定性算法。

Algorithm satisfiability ($E(x_1, \ldots, x_n)$)

{ Step 1: for i =1 to n do

x_i ←choice (true, false) /*利用 choice 直接猜中 x_i 的真假值*/

Step 2: if E (x_1, ... , x_n) is true then success () /*计算此布尔公式是否为真*/

else failure ();
}

上述的非决定性算法的时间复杂度为O(n^1)即代表满足问题落入NP集合中。

参考文献

引用

1. R. E. Ladner "On the structure of polynomial time reducibility," J.ACM, 22, pp. 151–171, 1975. Corollary 1.1. ACM site (http://portal.acm.org/citation.cfm?id=321877&dl=ACM&coll=&CFID=15151515&CFTOKEN=6184618) 页面存档备份 (https://web.archive.org/web/20200427145441/http://portal.acm.org/citation.cfm?id=321877&dl=ACM&coll=&CFID=15151515&CFTOKEN=6184618)，存于互联网档案馆.

来源

- Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. Introduction to Algorithms, Second Edition. MIT Press and McGraw-Hill, 2001. ISBN 0-262-03293-7. Section 34.2: Polynomial-time verification, pp. 979–983.
- Michael Sipser. Sections 7.3–7.5 (The Class NP, NP-completeness, Additional NP-complete Problems). Introduction to the Theory of Computation. PWS Publishing. 1997: pp. 241–271. ISBN 0-534-94728-X.
- David Harel, Yishai Feldman. Algorithmics: The Spirit of Computing, Addison-Wesley, Reading, MA, 3rd edition, 2004.
- 俞征武, 发现算法, 旗标出版股份有限公司, 2017.

外部链接

- Complexity Zoo: NP (https://complexityzoo.uwaterloo.ca/Complexity_Zoo:N#np)
- Graph of NP-complete Problems (http://page.mi.fu-berlin.de/aneumann/npc.html)
- American Scientist primer on traditional and recent complexity theory research: "Accidental Algorithms" (http://www.americanscientist.org/issues/pub/accidental-algorithms/)页面存档备份 (https://web.archive.org/web/20081012155440/http://www.americanscientist.org/issues/pub/accidental-algorithms/)，存于互联网档案馆